# Oracle® Warehouse Builder

Release Notes

10*g* Release 2 (10.2)

**B28226-05**

September 2008

This document contains important information not included in the Oracle Warehouse Builder documentation.

This document provides the following information on Oracle Warehouse Builder 10*g* Release 2 (10.2):

- Documentation Accessibility
- Related Publications
- Requirements
- Installing the Warehouse Builder 10.2.0.4 Patch
- Known Issues and Limitations
- Desupported Utilities
- Documentation Errata
- Resolved Issues

## Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. 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.

**ORACLE**®

**TTY Access to Oracle Support Services**

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

## Related Publications

The Oracle Warehouse Builder 10*g* Release 2 documentation set includes these publications:

- Oracle Warehouse Builder Installation and Administration Guide

- Oracle Warehouse Builder User's Guide

- Oracle Warehouse Builder API and Scripting Reference

To access the documentation set, including the latest version of the release notes, refer to the following Web site:

http://www.oracle.com/technology/documentation/warehouse.htm l

## Requirements

For the latest information on platform-specific requirements, certifications and patches, refer to the following Web site:

http://metalink.oracle.com

## Installing the Warehouse Builder 10.2.0.4 Patch

The following are reasons to upgrade to Warehouse Builder 10*g* Release 2 (10.2):

- After installing the patch, you can upgrade directly from Warehouse Builder 9.2.x releases.

- The patch resolves the bugs listed in Resolved Issues and introduces new enhancements as described in Documentation Errata.

**Pre-Installation**

Before you install this patch, check the following:

- On Unix-based platforms, you must be logged in as the UNIX user that owns the Oracle Warehouse Builder installation before performing the patch install.

- Stop and exit all Warehouse Builder components, including the Control Center Service, known as the Runtime Platform Service in Warehouse Builder 10.1 and earlier releases.

    To stop the Control Center, run the stop_service.sql script. In SQL*Plus, execute the following command:

    ```
      sqlplus owb_owner/owb_owner @owb home/owb/rtp/sql/stop_
    service.sql
    ```

    A return value of Unavailable indicates that you successfully stopped the service.

- If you applied DST patch 5921358 to your Warehouse Builder 10*g* Release 2 (10.2) installation, you must add write permissions to the *owb home*/jdk directory.

  In Unix, use the command prompt to navigate to the *owb home*/jdk directory and execute the following command:

  ```
  chmod -R +w *
  ```

  In Windows, use the Windows Explorer to perform the following steps:

  1. Right-click the *owb home*\jdk directory and select **Properties**.

  2. In the Attributes section, clear **Read-only**.

  3. Click **Apply**.

### Debugging the Warehouse Builder 10.2.0.4 Patch Installation

While installing Warehouse Builder 10.2.0.4, you may encounter the following errors:

- FileNotFoundException (Permission denied)

- FileNotFoundException (Access is denied) on *owb home*\jdk files

This means that you:

- patched a Warehouse Builder 10.2 home with DST patch 5921358

- did not follow the step on setting JDK write permissions described in the Pre-Installation section.

Follow these steps to recover from this error:

1. Leave the Oracle Universal Installer running.

2. Apply JDK write permissions as described in the Pre-Installation section.

3. After the fix, click **Continue** in the Oracle Universal Installer.

### Upgrading from Warehouse Builder Versions Earlier Than 10.2.0.1

To upgrade from a 9.2.x or 10.1.0.x version, first download and install Warehouse Builder 10.2.0.1 into a new ORACLE_HOME. Apply the 10.2.0.4 patch over this new home directory as described in the next section.

After applying the 10.2.0.4 patch, migrate the design data, using MDL export, and migrate the runtime data. For instructions on migrating data, refer to the installation guide for Oracle Warehouse Builder 10*g* Release 2.

### Upgrading from Warehouse Builder 10.2.0.1 or 10.2.0.2

Without deinstalling the existing Warehouse Builder software, install the Warehouse Builder 10.2.0.4 patch into the existing Warehouse Builder 10.2 home directory.

**To install the patch take the following steps:**

1. Using WinZip, extract the archive into a temporary directory on your local system. For example, such as C:\OWB10203.

2. Use WinZip to open the archive file. Be sure to enable the option **Use Folder Names.**

3.  Run the Oracle Universal Installer from the Warehouse Builder home directory.

    For Windows, launch *owb home*\oui\bin\setup.exe.

    For Linux, launch *owb home*\oui\bin\runInstaller.sh.

4.  Follow the prompts in the Oracle Universal Installer.

    For the source location, specify Disk1\stage\products.xml located under the temporary directory you created in step 1.

    For the home details, specify the home directory of the existing Warehouse Builder 10.2 installation.

5.  Navigate to the appropriate execution directory under the Warehouse Builder home directory and run the upgrade utility. The specific file you should execute depends on the release of the Oracle Database you are running.

    For Windows, the .bat files are located under the win32 directory. If your Oracle Database is either 9*i*R2 or10*g*R1 run the following:

    *owb home*/owb/bin/win32/DTupg102to10203_10gR1.bat

    Or, if your Oracle Database is 10*g*R2 run the following:

    *owb home*/owb/bin/win32/DTupg102to10203_10gR2.bat

    For UNIX, the .sh files are located under the unix directory. If your Oracle Database is either 9*i*R2 or 10gR1, run the following:

    *owb home*/owb/bin/unix/DTupg102to10203_10gR1.sh

    Or, if your Oracle Database is 10*g*R2 run the following:

    *owb home*/owb/bin/unix/DTupg102to10203_10gR2.sh

6.  Enter the database connection information, the Warehouse Builder repository owner credentials, and the SYS user credentials.

    The upgrade utility proceeds to upgrade the Warehouse Builder objects associated with the selected repository owner.

    The upgrade utility logs any errors in the file *owb home*/owb/reposasst/upg/log.txt. Note that you can ignore any creation errors reported due to objects already existing.

7.  Refer to the "Known Issues and Limitations" section for a list of limitations concerning the "Upgrade" process.

    For certain upgrade scenarios, you may need to take additional steps.

8.  Repeat steps 5, 6 and 7 for each repository owner.

9.  Start the Control Center Service using the start_service.sql script.

     sqlplus owb_owner/owb_owner @*owb home*/owb/rtp/sql/start_service.sql

    A return value of Available indicates that you successfully started the service. If the service fails to start, run the start_service.sql script again.

    If the service fails to start after two attempts, run the service_doctor script to determine the cause.

```
 sqlplus owb_owner/owb_owner @owb home/owb/rtp/sql/service_
doctor.sql
```

The return value should indicate that the service script is accessible to the database server.

If you encounter an obfuscation error when attempting to start the Control Center Service, run the script *owb home*/owb/rtp/sql/reset_repository.sql.

## Known Issues and Limitations

This section lists the known issues and limitations in the Oracle Warehouse Builder 10*g* Release 2 (10.2) release. They are grouped into the following categories:

- Business Intelligence Objects
- Database Dependencies
- Database Links
- Data Profiling
- Deployment
- Dimensional Objects
- Experts
- Importing Data from an Oracle Database
- Languages and Translations
- Mapping Debugger
- Mapping Designs and Configurations
- MDL Import
- MOLAP
- OLAP Catalog
- OMB Plus Scripts
- Oracle Enterprise Manager
- Repository Browser
- Upgrade
- User Interface
- Warehouse Builder Connector to SAP R/3

### Business Intelligence Objects

5213153, 5191402: NLS: NON-ASCII CHARACTERS ARE CORRUPTED IN GENERATED EEX FILE

When you deploy business definitions that include non-ASCII characters, the deployment fails with errors.

**Workaround**

Manually import the .eex file generated by Warehouse Builder for the deployed objects into your Discoverer EUL. The .eex file is located in the *owb home*/owb/deployed_files directory.

## Database Dependencies

### Oracle8i (8.1.7.4)

Oracle Scalar Data Types do not support BINARY_FLOAT, BINARY_DOUBLE, TIMESTAMP, TIMESTAMP WITH TIME ZONE, TIMESTAMP WITH LOCAL TIME ZONE, INTERVAL YEAR TO MONTH, INTERVAL DAY TO SECOND, XMLTYPE when Oracle Database 8.1.7.4 is used as source.

### Oracle9i

The Name and Address Cleansing and the Match-Merge features require Oracle9*i* Database at minimum.

Oracle Scalar Data Types do not support BINARY_FLOAT and BINARY_DOUBLE when Oracle9*i* is used as source and target.

2789391:DATABASE LINKS THAT CONNECT TO GATEWAYS REQUIRE GLOBAL-NAMES SET TO FALSE

If you have database links that connect to a Gateway version lower than 9*i*, set GLOBAL_NAMES to false on the Oracle instance that connects to the gateway.

## Database Links

4435282: PL/SQL CANNOT USE DATABASE LINK WITH NAME LONGER THAN 30 CHARACTERS

Referring to a database link whose name contains more than 30 characters results in an error. This is incorrect because database link names can contain up to 128 characters.

### Workaround

Reduce the length of the database link name. Or set GLOBAL_NAMES to FALSE.

## Data Profiling

6467940: DATA PROFILING RESTRICTION

The data profiling workspace location should be an Oracle database with version 10*g* or higher.

6505764: DPF0029 ERROR WHILE CREATING DATA PROFILE ON A TABLE CONTAINING MORE THAN 165 COLUMNS

Performing data profiling on a table containing more than 165 columns results in an error. The maximum number of columns that a table may contain for data profiling is 165.

### Workaround

For information about how to profile a table that contains more than 165 columns, see "Using Attribute Sets to Profile a Subset of Columns from a Data Object" on page 16.

### Deployment

6000145: ORA-02011 WHEN DEPLOYING MAPPING WITH DEPLOY ACTION SET TO REPLACE

When you use the SQL*Net connection type to create a location that corresponds to a remote database, deployment to this location fails.

**Workaround**

When you create Oracle locations of type SQL*Net, you must set up a TNS name entry for these locations. The TNS name must be accessible from the Oracle Database home. To do this, run the Net Configuration Assistant from the Oracle Database home.

While setting up a TNS name for use during deployment and execution of maps and process flows, the TNS name must be accessible from the Warehouse Builder home used to run the control center service. To make the TNS name accessible, run the Net Configuration Assistant from the Warehouse Builder home. Next, restart the control center service so that it can pick up the changes.

## Dimensional Objects

5189802: EXECUTING A CUBE MAPPING MAY GENERATE ERROR ORA-20101

When you attempt to execute a mapping that includes a cube partitioned on a time dimension, you may encounter ERROR ORA-20101: <timestamp> Error Occurred in <error location>: Attempt to write to non-existent partition...

6048511: SLOWLY CHANGING DIMENISON TYPE TWO WITH SKIP LEVELS NOT SUPPORTED

Avoid designing Type 2 SCDs with skip levels.

## Experts

5176858: USER INPUT IN A CUSTOM DIALOG MAY TRIGGER THE INCORRECT OUTCOMING TRANSITION

When you run an expert that includes multiple custom dialogs, the expert traverses the correct transition for the first custom dialog but may proceed with the wrong transition in subsequent custom dialogs. For example, if you select CANCEL on the first custom dialog and OK on a subsequent custom dialog, the expert may fail to reset a flag and therefore erroneously traverse the CANCEL transition after both custom dialogs.

**Workaround**

Reset the flag by entering `set EXIT_CODE 0` in the pre-processing of the task.

## Importing Data from an Oracle Database

5467424: IMPORT WIZARD THROWS ORA-00997 IMPORTING A TABLE

When you import table definitions from a database in which the `CURSOR_SHARING` parameter is set to FORCE or SIMILAR, the import fails.

**Workaround**

Set the database parameter `CURSOR_SHARING` to EXACT.

## Languages and Translations

5372377: CANNOT SET DATE WHEN MIDDLE-TIER LOCALE IS DIFFERENT

When the Locale setting of the Repository Browser listener is different from the Locale setting of the Repository Browser, errors occur when you try to display reports using the Repository Browser.

5112489: REPOSITORY ASSISTANT DOES NOT FOLLOW LOCALE PREFERENCE

In the Design Center, if you set the locale to a different language, the change does not affect the Repository Assistant. That is, if you select a new locale from Tools->Preferences->Appearance, commit the changes, and then launch the Repository Assistant, the system default language displays in the assistant despite your selection in the Design Center.

5122612: UNABLE TO CREATE MULTIBYTE TARGET USER IN OMB PLUS

When you create a target user using OMB Plus, if you use multibyte characters in the user name, the target user creation fails.

5031270: NLS:OWB1012:HARDCODE IN DESIGN CENTER.

When you change the Locale preference in the Design Center, certain object names or folder names still display in English. These include the names of various categories of pre-defined Warehouse Builder transformations (displayed under the Public Transformations node of the Global Explorer), MY_PROJECT, the default control center DEFAULT_CONTROL_CENTER, the folder called DERIVED_DATA_RULES that contains derived data rules, and the default configuration called DEFAULT_CONFIGURATION.

4999039: ASIAN(CJK) LANGUAGES ARE NOT SUPPORTED ON SUSE WITH JDK

Since JDK 1.4.2 does not support certain Asian languages on SLES-9, the characters in Japanese, Korean, Simplified Chinese, and Traditional Chinese may not be displayed correctly.

For more details about this issue, please refer to the following Web sites:

http://java.sun.com/j2se/1.4.2/system-configurations.html

http://java.sun.com/j2se/1.4.2/docs/guide/intl/locale.doc.html#jfc

5017820, 4915836: NON-EUROPEAN FONT GLYPHS ARE CORRUPTED IN SVG

When you export the diagram displayed on the Data Object Editor canvas to an .SVG file, certain Arabic and Asian characters are corrupted.

**Workaround**

Export the diagram using the JPEG format.

3066632: WAREHOUSE BUILDER FLAT FILE SAMPLE WIZARD COUNTS A MULTI-BYTE CHARACTER LENGTH AS 1

The Warehouse Builder Flat File Sample Wizard does not handle multi-byte characters in delimited files correctly, when translated into Japanese. The Sample Wizard handles a multi-byte character in delimited files as 1 length.

**Workaround**

You need to manually edit the length using the Flat File Sample Wizard.

## Mapping Debugger

2887449: MAP EXECUTION DONE MESSAGE COMES UP BEFORE RESULTS ARE DISPLAYED

In the current release, the Debugger displays a message that the mapping execution is complete, the progress bar continues to show activity, and then displays the debugging results. This is incorrect. The message that the mapping execution is complete should not appear until after the debugging results display.

### General Restrictions in the Mapping Debugger

These issues include functions currently included in the Mapping Debugger that have not yet been enabled as well as functions that intend to be added in future releases.

1.  Mappings run using the debug mode in the Mapping Editor are intended to be used for debug purposes only. Mappings run from the Mapping Editor do not perform as well as mappings that are run using the Control Center. This is attributed to the set up of temporary objects necessary to support the debugging capabilities. Use the Control Center to run mappings.

2.  You cannot pause an active debug run using the Pause button on the toolbar or the associated item in the debug menu.

3.  Mapping statistics will be provided in a future release. The statistics will appear as an additional tab on the left bottom panel.

4.  You cannot use the Repository Browser to view the results of a mapping run in debug mode.

5.  Breakpoint and watch settings are not preserved between debug sessions.

6.  Only mappings that can be implemented as a PL/SQL package can currently be run in debug mode. ABAP mappings are not supported in the debugger.

7.  The following mapping operators are not supported when running mappings in debug mode:

    ■  Advanced Queue

    ■  Flat File

## Mapping Designs and Configurations

2887323: DO NOT SET BREAKPOINTS FOR OPERATORS OR MAPPINGS THAT CANNOT BE STEPPED THROUGH

You will not be able to set breakpoints for operators that cannot be stepped through. In the Warehouse Builder client, the breakpoint button is not disabled. However, when you click this button, no action is performed.

2983137: DEBUGGER CREATES LOG FILE

The Debugger creates a log file called debugger.log in the *owb home*\owb\bin\admin directory. The log file includes each step processed during mapping debug sessions and the generated code.

**Workaround**

Delete this file, if it becomes very large.

2763192: FULL OUTER JOIN QUERIES TO TABLE FUNCTION GIVES WRONG NUMBER OF ROWS ON BULK FETCH

When the query containing a full outer join is passed as a ref cursor parameter to the table function, during execution, the wrong number of rows is fetched.

**Workaround**

Avoid mappings that result in a full outer join being used as a ref cursor parameter to a table function.

2577706, 2797671: POST MAPPING PROCESS RUNS REGARDLESS OF SUCCESS/FAILURE OF MAPPING

The mapping return status has one of three values:

- SUCCESS - Mapping completes successfully with no errors

- WARNING - Mapping completes with errors but not over the max error limit

- ERROR - Mapping does not complete OR mapping has errors over the max error limit

The Maximum Number of Errors parameter applies to the count of errors for the entire mapping run, whether run in set-based, row-based, or failover modes. Consider the following cases:

- Max number of errors set to 50, mapping run in set-based mode. Data did not load successfully. One error resulted from failure of the set-based load DML statement. Mapping return status is "WARNING".

- Max number of errors set to 50, mapping run in set-based mode, "Enable constraint" property set to false. Data loaded successfully but 60 constraint violation errors occurred during re-enabling of constraint. Mapping return status is "ERROR".

- Max number of errors set to 50, mapping run in row-based mode. Some data loaded successfully but with many errors. Mapping will terminate after hitting the 50th error. Mapping return status is "ERROR".

- Max number of errors set to 50, mapping run in set-based failover to row-based mode. Data did not load successfully in set-based process. One error resulted from failure of the set-based load DML statement. Some data loaded successfully in row-based process but with many errors. Mapping will terminate after hitting the 49th error in row-based because there was one error counted in set-based. Mapping return status is "ERROR".

3014556: CANNOT COPY A SAP SOURCE TABLE IN DEFINE TEST DATA

In this release, you may encounter the warning message

```
ORA-01861: literal does not match format string
```

when you use the Create Table option for an SAP source table in the define test data.

4483510: PLUGGABLE MAPPINGS DO NOT WORK IN SAP MAPPINGS.

5212360: MAPPINGS CREATED VIA OMB PLUS MAY DISPLAY INCORRECTLY

In the Mapping Editor, a mapping that was created using OMB Plus may display as containing a single operator despite having multiple operators.

**Workaround**

When viewing mappings created using scripting, use the auto-layout command in the Mapping Editor to view the entire mapping.

6074038: NEED TO DOCUMENT FIXES FOR BUG 5988620 (ROWID SUPPORT)

Starting with Oracle Warehouse Builder 10.2.0.3, you can use the pseudocolumns `ROWID` and `ROWNUM` in mappings. The `ROWNUM` pseudocolumn returns a number indicating the order in which a row was selected from a table. The `ROWID` pseudocolumn returns the rowid (binary address) of a row in a database table.

You can use the `ROWID` and `ROWNUM` pseudocolumns in Table, View, and Materialized View operators in a mapping. These operators contain an additional column called COLUMN USAGE that is used to identify attributes used as `ROWID` or `ROWNUM`. For normal attributes, this column defaults to TABLE USAGE. To use an attribute for `ROWID` or `ROWNUM` values, set the COLUMN USAGE to ROWID or ROWNUM respectively.

You can map a ROWID column to any attribute of data type `ROWID`, `UROWID`, or `VARCHAR2`. You can map ROWNUM column to an attribute of data type `NUMBER` or to any other data type that allows implicit conversion from `NUMBER`.

Note that `ROWID` and `ROWNUM` pseudocolumns are not displayed in the Data Object Editor since they are not real columns.

## MDL Import

4558688: SCRIPT GENERATION FOR IMPORT OF METADATA FOR EXTERNAL TABLES FAILS FOR LONGER DEFINITIONS

When you import metadata for an external table from an Oracle database, if the length of the external table access parameters is more than 4000 characters, the access parameters are truncated. Thus the imported external table cannot be generated successfully.

5075463: AFTER MDL IMPORT, LOCATIONS MUST BE TESTED BEFORE SOURCE IMPORT/DATA VIEWER

When you import an MDL file that contains multiple modules and their associated locations, the locations are imported without their passwords. If you try to use any of these modules, for example, import metadata into the module or view data stored in the objects, Warehouse Builder displays a dialog that enables you to complete the location information. However, after you specify the location details, test the location, and close the dialog, an error is displayed.

**Workaround**

After you import the MDL, first test all the locations before you use the modules associated with these locations.

## MOLAP

XML LOAD FROM VIEW WITH DBLINKS DOES NOT LOAD ROWS, LOCAL VIEW DOES LOAD ROWS

When you create a view in the database that links to a remote table via a dblink, and try to load from this view into the analytic workspace (for example, AW Manager) no rows are loaded. An exact replica of the view on a local table does produce rows.

LOADING INTO AW WITH AW XML FAILS WHEN VIEW CONTAINS TABLE FUNCTION

When using a view that uses a table function, load errors occur.

## OLAP Catalog

5124592: ROLAP CATALOG LIMITATIONS

For dimensions with a ROLAP implementation, there are implications and limitations related to the various dimension structures when either reporting on the underlying tables or deploying to the OLAP catalog. Although the dimension may be successfully deployed, errors could occur when other applications, such as BI Beans or Discoverer access the OLAP catalog. The following are items that are affected by this limitation:

- No reporting tool has metadata about all aspects of dimensional metadata we capture, so this must be incorporated into the query/reports. Otherwise you will see odd information because of the way the data is populated in the implementation tables.

  The dimension and cube implementation tables store solved rows which contain negative key values. You can filter out these rows in your queries or reports. When you create a query or report, use the view that is associated with a dimension instead of the dimension itself. Each dimension has a view that is associated with it. The view name is specified in the configuration property View Name of the dimension or cube.

- Skip-level hierarchies and ragged hierarchy metadata not deployed to OLAP catalog.

  If you create a dimension that contains skip-level or ragged hierarchies, the metadata for these is stored in the Warehouse Builder repository but is not deployed to the OLAP catalog.

- Dimensions with multiple hierarchies must have all dimension attributes mapped along all the hierarchies. This is because of OLAP bug 4945008.

## OMB Plus Scripts

OMB Plus scripts that worked in Warehouse Builder 10.1 release may not work in the current 10*g* Release 2 (10.2) release. The new release introduces changes in syntax and some of the property values have been changed for NLS compliance. For further information, see the *Oracle Warehouse Builder Scripting Reference*.

4581172, 4658511: DEFERRED OBJECTS LISTED IN METAMODEL QUERY

OMB Plus commands that describe the model display certain objects that were descoped from Oracle Warehouse Builder10*g* Release 2 (10.2). These objects will be supported in a future release of Warehouse Builder. The list of objects that are listed in OMB Plus model queries, but are not available for use in the current release are as follows:

- ADVANCED_QUEUES (limited support)

- ASSOCIATION_RULES_MODEL

- ATTRIBUTE_IMPORTANCE_MODEL

- CLASSIFICATION_MODEL

- CLUSTERING_MODEL

- FEATURE_SELECTION_MODEL

- LCRCAST_OPERATOR

- LCRSPLITTER_OPERATOR

- MINING_MODEL

- MULTIPLEXER_OPERATOR

- QUEUE_PROPAGATION

- QUEUE_TABLE

- REGRESSION_MODEL

- REAL_TIME_MAPPING

- STREAMS_CAPTURE_PROCESS

- STREAMS_QUEUE_SUBPROCESS

- STREAMS_QUEUE

- TABLE_FUNCTION (limited support)

## Oracle Enterprise Manager

2614173, 2624121: OEM JOB EXECUTION PERFORMANCE ISSUE

When executing jobs through OEM there may be a delay between the job completing as seen in the OEM console and the Control Center Service being notified by OEM that the job has completed. Although the job completes successfully, there may be a 50 second delay.

**Workaround**

Wait a minute or two for the notification.

## Repository Browser

5129228: MULTIPLE REPOSITORY BROWSER SESSIONS FROM A SINGLE CLIENT MACHINE NOT SUPPORTED

If you launch multiple Repository Browser sessions from a single internet browser on a single client machine, you will encounter unexpected results. This scenario is not supported. As a consequence, you cannot browse multiple repositories with a single internet browser on a single machine.

**Workaround**

Each Repository Browser session requires its own internet cookies storage. If feasible, you can launch multiple, different internet browsers to maintain an equivalent number of Repository Browser sessions on the same machine.

## Upgrade

1477144: WAREHOUSE UPGRADE CANNOT DELETE AN INDEX

The Warehouse Upgrade does not drop an index from the data warehouse when you delete that index from the model in the Warehouse Builder repository. The upgrade script is created and deployed successfully but the index remains in the database.

**Workaround**

Use another database tool external to Warehouse Builder (such as, SQL*Plus, or Enterprise Manager) to drop the index.

1811047: PARTITION VALUES AND WAREHOUSE UPGRADE SCRIPTS

Warehouse Upgrade scripts are incorrectly generated for objects that have had partition values changed or new partition keys added to a previously deployed partition.

**Workaround**

Using an external Oracle database tool (SQL*Plus or Enterprise Manager), drop the partition and then recreate it.

## User Interface

5127178: LAYOUT OF OWB WINDOW SEEMS CORRUPTED OR UNUSABLE

Sometimes, although very rarely, the layout of an editor window may become corrupted or unusable.

**Workaround**

Delete the appropriate layout editor file, close the Warehouse Builder client, and then log in again.

The layout editor files use self-explanatory names and are located in the *owb home*/owb/bin/admin directory. For example, the Data Object Editor layout file is called SchemaEditorLayout.xml and the Mapping Editor layout file is called MappingEditorLayout.xml.

## Warehouse Builder Connector to SAP R/3

Using the Warehouse Builder Connector to SAP R/3, you can access versions of SAP that rely on the SAP R/3 technology, including SAP 3.x, 4.x, 4.7, 5.0 and mySAP ERP 2004. For a list of supported platforms and versions, refer to the certification information available a the following Web site:

http://metalink.oracle.com

5215322: WAREHOUSE BUILDER CONNECTOR TO SAP R/3 REQUIRES SAP LIBRARY FILE

To access SAP R/3 in the Design Center, first obtain the SAP RFC library from SAP and copy it to `<owb_client_home>/owb/bin/admin`.

To access SAP R/3 in the Control Center, first obtain the SAP RFC library from SAP and copy it to `<owb_server_home>/owb/bin/admin` where owb_server_home is the directory where Warehouse Builder control center service is installed.

## Desupported Utilities

The following utilities are not supported in Oracle Warehouse Builder 10*g* Release 2 (10.2):

- The command-line utility for the MDL Import and MDL Export

- The MDL Upgrade Utility

- CWM Export

  You cannot use the bridge to export metadata into an OMG CWM format.

- You cannot use the Parallel Row Code configuration option in mappings.

### Objects With Restricted Functionality

You cannot create advanced queues in Oracle Warehouse Builder 10*g* Release 2 (10.2). You can however import an MDL file that contains advanced queues created using a previous release and deploy these advanced queues.

Note that, in the current release, you cannot create a mapping with an AQ operator. You can have a mapping containing an AQ operator only if you import an MDL file that was created using an earlier Warehouse Builder version.

## Documentation Errata

This section includes the following topics:

- Revisions to the Warehouse Builder User's Guide 10g Release 2

- New in Warehouse Builder 10.2.0.3 and 10.2.0.4

- New in Warehouse Builder 10.2.0.2

### Revisions to the Warehouse Builder User's Guide 10*g* Release 2

#### Revision to List of Supported Sources and Targets

There is an omission in the table entitled "Supported Sources and Targets". Warehouse Builder can target any database accessible through Oracle Heterogeneous Services, including but not limited to DB2, DRDA, Informix, SQL Server, Sybase, and Teradata. Warehouse Builder can also target any data store accessible through the ODBC Data Source Administrator, including but not limited to Excel and MS Access. To load data into spreadsheets or third-party databases, first deploy to a comma-delimited or XML format flat file.

#### Revision to Discussion on Using Subprocess Activities

In Chapter 22, Using Activities in Process Flows, the text incorrectly states the following:

"Use a subprocess activity to launch a previously created process flow. From one process flow, you can launch any other process flow that is contained within the same process flow package."

The correct statements are as follows:

"Use a subprocess activity to launch a previously created process flow. From one process flow, you can launch any other process flow that is contained within the same *or any other* process flow package."

**Revision to Sparse Dimensions Discussion**

The following text replaces the content in the section entitled "Advanced Dialog" in Chapter 4, Defining Dimensional Objects, that begins with "All the sparse dimensions in a cube" and ends with the following sentence: When a cube uses a normal storage (not compressed), list the dimensions such that the least sparse dimension appears at the top of the list.

All the sparse dimensions in a cube must be grouped together starting from the least sparse to the most sparse. For example, the Sales cube references the dimensions Times, Products, Promotions, and Channels. This is the order in which the dimensions are listed in the Advanced dialog box. The cube data is sparse along the dimensions Promotions and Channels, with Promotions being the most sparse. Then all these dimensions should appear as a group in the following order: Times, Products, Channels, and Promotions. You cannot have any other dimension listed in between these dimensions.

Use the following guidelines to order dimensions:

- List the time dimension first to expedite data loading and time-based analysis. Time is often a dense dimension, although it may be sparse if the base level is Day or the cube has many dimensions.

- List the sparse dimensions in order from the one with the most members to the one with the least. For a compressed cube, list the sparse dimensions in order from the one with the least members to the one with the most.

**Revision to Data Flow Operators, Splitter Operator Discussion**

In Chapter 26, Data Flow Operators, in Oracle Warehouse Builder User's Guide, the section titled Splitter Operator should be clarified as follows.

Replace this sentence:

"The output group `REMAINING_ROWS` contains all input rows that are not included in any output group." with this sentence:

In most cases, the output group `REMAINING_ROWS` contains all input rows that are not included in any output group. However, when the split condition contains an attribute whose value is null, the corresponding rows are not moved to the `REMAINING_ROWS` output group.

**Revision to the Discussion on Performing Data Profiling**

The following information should be included in the section entitled "Using Data Profiling" in Chapter 20, Ensuring Data Quality.

Data profiling can only profile 165 columns per table, view, or materialized view. This limitation is described in bug 6505764.

If your table, view or materialized view has more than 165 columns, you must use an attribute set to select 165 or fewer columns for profiling. The following section describes the procedure for profiling data using attribute sets.

## Using Attribute Sets to Profile a Subset of Columns from a Data Object

You can use an attribute set to restrict a data profiling operation to a subset of columns from a table, view or materialized view. Reasons to use an attribute set include:

- You can decrease profiling time by excluding columns for which you do not need profiling results.

- Data profiling can only profile up to 165 columns from a table, view or materialized view at a time. You can use an attribute set to select a set of 165 or fewer columns to profile from the object.

Data profiling, using attribute sets, consists of the following high-level steps:

1. Defining Attribute Sets
2. Profiling the Columns in an Attribute Set

**Defining Attribute Sets**   Use the following steps to define an attribute set in a table, view, or materialized view.

1. In the Project Explorer, double-click the table, view, or materialized view.

   The Data Object Editor for the selected object is opened.

2. In the Details panel, select the Attribute Sets tab.

3. In the Attribute Sets section, click a blank area in the Name column and enter the name of the attribute set to create.

4. Close the Data Object Editor.

5. Double-click the data object in which you created an attribute set in step 3.

   The Data Object Editor for the selected object is displayed.

6. On the Attributes Sets tab, select the name of the attribute set created in step 3.

   The Attributes of the selected attribute set section displays the attributes in the data object.

7. Select **Include** for all the attributes that you want included in the attribute set.

8. Save your changes and close the Data Object Editor.

**Profiling the Columns in an Attribute Set**   Use the following steps to profile columns contained in the attribute set.

1. In the Project Explorer, right-click the Data Profiles node and select **New**.

   The Welcome page of the Create Data Profile Wizard is displayed.

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

3. On the Name and Description page, enter a name and an optional description for the data profile. Click **Next**.

4. On the Select Objects page, select the data object that you want to profile and use the shuttle arrows to move the data profile to the Selected list.

5. When the selected data object contains attribute sets, the Choose Attribute Set dialog box is displayed.

6. Select the attribute set that you want to profile and click **OK**.

   The Select Objects page is displayed.

7. On the Select Objects page, click **Next**.

8. On the Summary page, review the options you choose on the previous wizard pages and click **Finish**.

   The data profile is created and added to the Navigator tree.

For details about how to profile objects using the data profile, refer to the section titled "Using Data Profiling" in the *Oracle Warehouse Builder User's Guide*.

## New in Warehouse Builder 10.2.0.3 and 10.2.0.4

The following sections describe functionalities that are new in this patch and not yet documented in the Oracle Warehouse Builder User's Guide or online help:

- Enhancements to Support Oracle Applications Products
- Enhancements for Designing Mappings and Using Operators
- Using DML Error Logging For Handling Error Tables
- Enhancements to Type 2 Slowly Changing Dimensions

### Enhancements to Support Oracle Applications Products

Warehouse Builder 10.2.0.4 includes enhancements and bug fixes to internally support Oracle Applications products. Currently, there are no plans to merge these fixes into Warehouse Builder 11*g* Release 1.

5385677 EPM 1

5339843 EPM 2

5385670 EPM 3

4525937 EPM 4

5878289 EPM 5

6032502 EPM 6

5006525 LCM 1

5233636 LCM 2

6022685 MIG 1

5879075 CDH 1

### Enhancements for Designing Mappings and Using Operators

#### 6073830: SET BASED DELETE IN MAPPING CODE GENERATION

Beginning in Warehouse Builder 10.2.0.3, you can design a mapping with loading type set to `Delete` and code generation set to `Set Based`.

#### 6073813: AVAILABILITY OF MERGE OPTIMIZATION PROPERTY FOR TABLE OPERATORS

Beginning in Warehouse Builder 10.2.0.3, you can enable the `Merge Optimization` property for table operators. When set to `True`, this property optimizes the invocation or execution of expressions and transformations in the MERGE statement.

Consider, for example, a mapping in which the target table contains a column that is part of the update only and is mapped to a transformation. In previous releases, Warehouse Builder would execute the transformation for all rows, unnecessarily including rows that did not require transformation. Beginning in this release, if `Merge Optimization` is enabled, then Warehouse Builder calls the transformation only in the update part of the MERGE statement.

### Using DML Error Logging For Handling Error Tables

#### 6020158: INCLUDE DML ERROR LOGGING FOR DML ON TABLES

Warehouse Builder 10.2.0.3 introduces enhancements for handling error tables. In previous releases, shadow table name and shadow tablespace captured only logical errors related to applying data rules to tables, views, and materialized views.

Beginning in this patch, the error table is generated if a shadow table name is specified. The Oracle DML error columns are always generated if shadow table name is specified. The data rule columns are only generated if a data rule is applied.

The following changes are made to the definition of the Warehouse Builder generated error table to support DML error logging. These should have no impact on existing customers with data rule maps.

Set based SQL statements such as INSERT/MERGE and multi table insert support DML error logging. The DML error logging is enabled when the error table is defined on the table, view, or materialized view operator in the mapping. If there is no error table name on the operator, no error logging will be generated.

For the DML, the error table is taken from the error table name property on an operator. The reject limit is the max number of error configuration property on the map. This cannot be altered at runtime. The DML errors are tied to the Warehouse Builder runtime audit data using the step or detail audit id.

The code generated for set-based PL/SQL maps uses DML error logging if there is an error table defined on the table operator.

For row-based code, the table, view, or materialized view operator also has a property for truncating the error table before running. This property when enabled truncates the error table before executing the map definition.

### Enhancements to Type 2 Slowly Changing Dimensions

#### 5934899: NEED TO DOCUMENT IN DETAIL SCD2 UPDATE BEHAVIOR

All the levels in a dimension need not store historical data. Typically, only the lowest level, also called the leaf level, stores historical data. However, you can also store historical data for other dimension levels.

For example, consider the Customers Type 2 SCD that contains two levels, Household and Customer. Table 1 lists level attributes of the Customers Type 2 SCD.

*Table 1    Customers Type 2 SCD Attributes*

| Attribute Name | Type of Identifier |
| --- | --- |
| ID | Surrogate identifier |
| BUSN_ID | Business identifier |
| ADDRESS | |
| ZIP | |
| MARITAL_STATUS | |
| HOME_PHONE | |

*Table 1   (Cont.)  Customers Type 2 SCD Attributes*

| Attribute Name | Type of Identifier |
| --- | --- |
| EFFECTIVE_DATE | Effective Date |
| EXPIRATION_DATE | Expiration Date |

Customer is the leaf level and Household is the non-leaf level.

The Household level implements the following attributes: ID, BUSN_ID, ADDRESS, ZIP, EFFECTIVE_DATE, and EXPIRATION_DATE. The Customer level implements the following attributes: ID, BUSN_ID, MARITAL_STATUS, HOME_PHONE, EFFECTIVE_DATE, and EXPIRATION_DATE.

The table that implements this Type 2 SCD (for a relational or ROLAP implementation) contains the following columns: DIMENSION_KEY, H_ID, H_BUSN_ID, H_ADDRESS, H_ZIP, H_EFFECTIVE_DATE, H_EXPIRATION_DATE, C_ID, C_BUSN_ID, C_MARITAL_STATUS, C_HOME_PHONE, C_EFFECTIVE_DATE, and C_EXPIRATION_DATE.

**What is Hierarchy Versioning**?

When the non-leaf level of a dimension contains versioned attributes, the versioning of this non-leaf level results in the versioning of its corresponding child records, if they have effective date and expiration date attributes. For example, when the value of the H_ZIP is updated in a particular Household level record, the child records corresponding to this Household level are automatically versioned.

**Enabling Hierarchy Versioning**

Hierarchy versioning is not enabled by default for Type 2 SCDs. When you create a Type 2 SCD using the Create Dimension Wizard, hierarchy versioning is disabled. You must use the Data Object Editor to enable hierarchy versioning.

**To enable hierarchy versioning:**

1.  Right-click the Type 2 SCD in the Project Explorer and select **Open Editor**.

    The Data Object Editor is displayed.

2.  Navigate to the SCD tab.

3.  Click **Settings** to the right of the Type 2: Store the Complete change history option.

    The Type 2 slowly changing dimension dialog box is displayed. The attributes of a level are displayed under the respective level node.

4.  In the child level that should be versioned when its parent attribute changes, for the attribute that represents the parent attribute of this child level, select **Trigger History** in the Record History column.

    For example, you create the Customers Type 2 SCD using the Create Dimension Wizard. Open the Data Object Editor for this Type 2 SCD and navigate to the Type 2 slowly changing dimension dialog box as described in steps 1 to 3. The Customer level has an attribute called HOUSEHOLD_ID. This attribute represents the parent attribute of each Customer record. For the HOUSEHOLD_ID attribute, select **Trigger History** in the Record History column.

**Updating a Type 2 SCD**

You can update the following in a Type 2 SCD:

- Leaf level attribute

- Leaf level versioned attribute

- Non-leaf level attribute

- Non-leaf level versioned attribute

- Leaf level parent attribute

The following sections describe the functionality for these update operations.

**Updating a Leaf Level Attribute**

When you update a leaf level attribute, the value of this attribute is updated in the corresponding record. For example, if you update the value of C_HOME_ PHONE in a Customer level record, the record is updated with the changed phone number.

**Updating a Leaf Level Versioned Attribute**

When you update a leaf level versioned attribute, the current record is marked as closed. A new record is created with the updated value of the versioned attribute. For example, if you update the marital status of a customer, the current record is marked as closed. A new record with the updated marital status is created for that customer.

**Updating a non-leaf Level Attribute**

When you update an attribute in a non-leaf level, the open records of the non-leaf level and the child records corresponding to this non-leaf level are updated with the new value. For example, when you update the H_ADDRESS attribute in a Household level record, the current open record for that household is updated. All open child records corresponding to that particular household are also updated.

**Updating a non-leaf Level Versioned Attribute**

The update functionality depends on whether hierarchy versioning is enabled or disabled.

**Hierarchy Versioning Disabled**

The non-leaf level record corresponding to the versioned attribute is closed and a new record is created with the updated value. The child records of this non-leaf level record are updated with the changed value of the non-leaf level versioned attribute.

For example, when the value of H_ZIP in a Household level record is updated, the current open record for that household is closed. A new record with the updated value of H_ZIP is created. The value of H_ZIP is updated in all the child records corresponding to the updated household record.

**Hierarchy Versioning Enabled**

The non-leaf level record corresponding to the versioned attribute is closed and a new record is created with the updated value. Child records corresponding to

this non-leaf level record are also closed and new child records are created with the updated value.

For example, when the value of H_ZIP in a Household level record is updated, the current open record for that household and its corresponding child records are closed. New records are created, with the updated value, for the household and for the child records corresponding to this household.

### Updating the Leaf Level Parent Attribute

In addition to updating the level attributes in a Type 2 SCD, you can also update the parent attribute of a child record. In the Customers Type 2 SCD, the attribute H_BUSN_ID in a Customer record stores the parent attribute of that customer. The update functionality for the leaf level parent attribute depends on whether hierarchy versioning is enabled or disabled.

### Hierarchy Versioning Disabled

The child record is updated with the new parent attribute value.

For example, when you update the value of the H_BUSN_ID attribute representing the parent record of a Customer record, the Customer record is updated with the new values.

### Hierarchy Versioning Enabled

The child record is closed and a new record with the changed parent attribute value is created.

For example, when you update the H_BUSN_ID attribute of a customer record, the current customer record is closed. A new customer record with the updated H_BUSN_ID is created.

### 5980875: NO DOC FOR CUBE ACTIVE_DATE IN MAPPING

The Cube operator contains an attribute called ACTIVE_DATE. This attribute represents the point in time that is used to determine which record in a Type 2 SCD is the active record. This property is applicable only when the cube you are loading has one or more Type 2 SCDs.

If you do not map an attribute from the source to the ACTIVE_DATE, SYSDATE is used as the default.

If you map a source attribute to ACTIVE_DATE, the value of the source attribute is used to determine which version of the Type 2 SCD record is referenced by the cube record.

For any cube that references a dimension in which the level is of a Type 2 SCD, the WHERE clause generated to determine the dimension member is as follows:

```
...
WHERE
(...
   (<dim_name>.DIMKEY = <lookup_for_dimension_dimkey> AND
        (<level>_EFFECTIVE_DATE <= ACTIVE_DATE AND
               <level>_EXPIRATION_DATE >= ACTIVE_DATE) OR
        (<level>_EFFECTIVE_DATE <= ACTIVE_DATE AND
               <level>_EXPIRATION_DATE IS NULL))
...)
```

### 5978298: NEED TO DOCUMENT FIXES FOR BUG 5957778

In a mapping that loads a Type 2 SCD, if you map attributes from the source to the effective date attribute of a level, Warehouse Builder performs the following:

- While loading the initial record, if the value of the source attribute is earlier than the value specified by the Default Effective Time of Initial Record property of the dimension operator (bound to the Type 2 SCD), the value from the source is used as the effective date of the record; else the value specified in the Default Effective Time of Initial Record property is used as the effective date of the record.

- During subsequent loads for the record, if the record is being versioned, the effective time of the new record is taken from the source. If no value is given for the effective time, SYSDATE is used. The expiration time of the closed record is set to the effective time of the new record minus the gap.

If you do not map attributes from the source to the effective date attribute of a level, Warehouse Builder performs the following:

- While loading the initial record, the value specified in the Default Effective Time of Initial Record property is used as the effective date of the record.

- During subsequent loads for the record, if a new version is being created, the effective time of the new record is taken from the source. If no value is given for the effective time, SYSDATE is used. The expiration time of the previous version is taken as the effective time of the new version minus the gap.

  For more information about the gap, see 5936837: DIMENSION OPERATOR PROPERTIES EFFECTIVE TIME/EXPIRATION TIME, GAP, GAP INTERVAL - DOC ISSUES.

---

**Note:** Mapping to the Expiration Date attribute of a level is not allowed. While loading a record, the Default Expiration Time of Open Record property is used as the expiration date. The default value of this property is NULL.

---

For example, you create a mapping that loads the Products Type 2 SCD. The leaf level of this Type 2 SCD, Product, is loaded from a source table. The Dimension operator has the following properties:

Default Effective Time of Initial Record: 01-jan-2000

Default Effective Time of Open Record: SYSDATE

Default Expiration Time of Open Record: 01-jan-2099

The effective date attribute of the Product level is mapped from the source attribute EFF_DATE. Consider a source Product level record with the value of EFF_DATE as 21-mar-2007.

When the initial Product level record is loaded, the effective date stored in this record is 01-jan-2000 and the expiration date is 01-jan-2099. When this Product level record is versioned during a subsequent load on 21-mar-2007, the value of the source attribute overrides the Default Effective Time of Open Record property. The effective date stored in the new Product level record is 21-mar-2007 and expiration date is set to 01-jan-2099. The initial Product level record is closed with the value of expiration date set to 21-mar-2007 minus the Type 2 Gap value.

**6011680: NEED TO DOCUMENT FIXES FOR BUG 5979392**

If a mapping that loads a cube references at least one Type 2 SCD that has the Default Expiration Time of Open Record set to a non-NULL value, then the ACTIVE_DATE attribute of the Cube operator must be mapped from the source that contains the date value that defines the range for the dimension record.

If the ACTIVE_DATE attribute is not mapped from the source, then the SYSDATE value will define the date range for the dimension record.

When the ACTIVE_DATE attribute is mapped from the source, the source attribute value is used to perform a range comparison to determine which dimension record should be loaded.

For a description of the logic used to perform the lookup for the dimension member, see 5980875: NO DOC FOR CUBE ACTIVE_DATE IN MAPPING.

**5936837: DIMENSION OPERATOR PROPERTIES EFFECTIVE TIME/EXPIRATION TIME, GAP, GAP INTERVAL - DOC ISSUES**

The following configuration properties are applicable to Type 2 SCDs only.

- **Default Effective Time of Initial Record:** It represents the default value assigned as the effective time for the initial load of a particular dimension record. The default value set for this property is SYSDATE.

- **Default Effective Time of Open Record:** It represents the default value assigned as the effective time of open records, after the initial record. The default value set for this property is SYSDATE. This value should not be modified.

- **Type2 Gap:** When a record is versioned, the Type 2 Gap property represents the time interval between the expiration time of an old record and the effective time of the current record.

  When the value of a triggering attribute is updated, the current record is closed and a new record is created with the updated values. Since the closing of the old and opening of the current record occur simultaneously, it is useful to have a time interval between the expiration time of the old record and the effective time of the open record, instead of using the same value for both.

- **Type2 Gap Units:** It represents the unit of time used to measure the gap interval represented by the Type2 Gap property. The options are: Seconds, Minutes, Hours, Days, and Weeks.

For example, the mapping that loads the Products Type 2 SCD has the following configuration properties:

Default Effective Time of Initial Record: 01-jan-2000

Default Effective time of Open Record: SYSDATE

Default Expiration Time of Open Record: 01-jan-2099

Type2 Gap: 1

Type2 Gap Units: MINUTE

Product is the leaf level in this Type 2 SCD. The data type of the effective date and expiration date attributes is TIMESTAMP. The first product record is loaded on 20-Apr-2007. The product record uses the following values:

Effective Time: 01-jan-2000

Expiration Time: 01-jan-2099

When the triggering attribute of this product record is updated on 22-apr-2007 at 10:45 AM, the current product record is closed and a new record is opened with the updated values.

The closed product record has the following values:

Effective Time: 01-jan-2000

Expiration Time: 22-apr-2007 10:44:00

The currently open `Product` record has the following values:

Effective Time: 20-apr-2007 10:45:00

Expiration Time: 01-jan-2099

## New in Warehouse Builder 10.2.0.2

5412790: TREE WALKING LOGIC IN WAREHOUSE BUILDER MAPPINGS

Beginning in Warehouse Builder 10.2.0.2, the Mapping Editor enables you to recursively join a table onto itself.

Also known as *tree walking*, recursively joining a table back onto itself enables you to retrieve records in a hierarchy. For example, consider a table that contains employee data including the manager for each employee. Using tree walking, you could determine the hierarchy of employees reporting up to a given manager.

To perform tree walking in the Warehouse Builder, create a mapping and add the table that contains the hierarchal definition. Connect that table to a filter operator. In the filter operator, define the filter condition with CONNECT BY as the first two words. Include only the connect by logic in the filter operator. That is, do not include any AND or OR logic in the filter.

5638033: SET OPERATOR FACILITATES MATCHING ATTRIBUTES

Beginning in Warehouse Builder 10.2.0.2, the set operator in the Mapping Editor assists you in matching attributes between two data streams. To match attributes from two data streams in a mapping, define the data streams as input groups into the set operator. On the Input Attributes tab, select the **Synchronize <Input Group Name>.** The synchronize operation rearranges and add attributes to the target group such that the target group most closely matches the source group. The synchronize operation uses the following rules to find or create a match in the target:

1) Looks for an existing attribute in the target that matches name and datatype

2) Looks for an existing attribute in the target whose description matches the source name, and datatype matches source datatype

3) If (1) and (2) fail, creates a new attribute with the source name and datatype, and puts it in the correct matching position

Any unmatched target group attributes are indicated by "***UNMATCHED***" into the attribute description.

To force a target attribute to match a specified source attribute, type the source group attribute as the target attribute description.

## Resolved Issues

The following is a list of issues resolved in Oracle Warehouse Builder 10*g* Release 2 (10.2).

6052000 ORA-00918 WHEN DEPLOYING DIM MAP WITH COLUMN REUSED FOR ALL LEVELS

6020158 INCLUDE DML ERROR LOGGING FOR DML ON TABLES

6013888, 5960870: UNABLE TO UPGRADE OBJECTS CONFIGURED WITH COMPRESSION VALUES

6013853 EDIT MAP INSTANTIATES ALL PLUGGABLE MAPS IN PROJECT

6004133 REG: CHILD LOOSES PARENT REFERENCE WHEN SCD2 ATTRIBUTE CHANGE IN PARENT

6000783 ONE-OFF PATCH FOR BUG 5701125 ON TOP OF VERSION 10.2.0.2

5997728 ADD EXECUTION PARAMETERS TO DBMS SCHEDULER SCHEDULE JOBS

5988620 ROWID/ROWNUM SUPPORT NEEDED IN MAPPING

5983807 EXTRACTION FROM CUBE DOES NOT WORK

5983413 REMOVE FROM CUBE SHOULD REMOVE ALL RECORDS THAT REFERENCE CLOSED AND OPENED DIM

5979392 CUBE LOADING: WRONG DIMENSION LOOKUP IF DEFAULT EXP.DATE IS NOT NULL

5978644 10.2.0.3: FORWARD PORT FOR BUG 5955424

5978530 PORT TOKYO BUG 5976723 FIX TO PARIS CODE LINE.

5963623 JNP WHEN SETTING DEFAULT_EFFECTIVE_TIME_OF_INITIAL_ RECORD

5960870 METCASH - UNABLE TO UPGRADE TABLES ISSUES AFTER APPLYING MEGAPATCH#5941546

5960663 10.2.0.3: FORWARD PORT FOR BUG 5394180

5960099 BACKWARD PORT OF BUG 5871297 IN 10.2.0.3 PARIS PATCH

5955424 OWB GENERATED TABLE OP LOADING TYPE SET TO UPDATE/INSERT NOT INSERT FOR CUBE

5938625 FULL OUTER JOIN GENERATED FOR KEY LOOKUP

5936574 REG: MAP WITH SCD2 SNOWFLAKE (SCD2 ON LOWEST LEVEL) - INVALID

5929994 ERROR THE EXPRESSION IS NOT PROPERLY FORMED - IN DESIGN CENTER POST UPGRADE.

5927402 REG: MAPS WITH PEL AND DATE INPUT PARAMETER - EXECUTION ERROR

5924390 REG: MAPS WITH MORE THAN 2 LOOKUP IN A ROW - WRONG CODE GENERATED

5919528 PLUGGABLE MAPPING FOLDERS TREE PERFORMANCE UNBEARABLE

5897921 MAPPING TRANSFORMATION TYPE AGGREGATOR VALIDATE ISSUE CHANGES ORDER

5894819 BACKPORT OF BUG 5486490 ON TOP OF VERSION 10.2.0.2

5880625 MATCH MERGE MATCH NEW RECORDS ONLY NOT WORKING

5871698 DATA PROFILING ALWAYS FAILS USING OWB 10.2.0.2.8

5865654 INCORRECT ANSI OUTER JOIN CONDITION CREATED FOR MAP WITH LOOKUPS

5865483 CANNOT DISABLE ANSI SQL SYNTAX FOR SOME MAPPINGS

5864264 AFTER UPGRADING TO OWB10.2.0.2.0 GETTING VLD-1108: OPERATOR X IS NOT PROPERLY CO

5861252 IMPORTED PACKAGE: CODE LOST AFTER DEPLOYMENT

5858086 CC_MIGRATE.SQL FAILS IF OBJECT LISTED IN TSUPGRADE_DROP_ OBJECTS.SQL DO NOT EXIST

5854738 PERFORMANCE PROBLEMS WITH DELETE MAPPINGS CONTINUE AFTER APPLYING OWB 10.2.0.2

5853639 OMBINSTALL FAILS BECAUSE OF OWBCC_HOME AFTER OWB 10.2.0.2

5845656 NEED EFFECTIVE DATES FOR INITIAL VERSION AND NEW VERSION FOR EPM

5845640 OWB DOES NOT VERSION MEMBERS WHEN PARENT IS VERSION FOR TYPE 2 SCD

5754997 GUI FREEZE (ENQ: TX - ROW LOCK CONTENTION) DURING INCREASING MULTI-USER LOAD

5747781 CONDITION MATCH RULES ALWAYS PASS WHEN VALUE IS BLANK

5741112 WITH CC SERVICE ON AIX PLATFORM,BI DEPLOYMENT THROWS RPE-02150 ERROR

5690957 MATERIALIZED VIEW OPERATOR PROPERTIES DEFAULT INDEX DEFAULT INDEX TABLESPACE

5680431 INPUT PARAMETER OF DATA TYPE INTEGER NOT PROMPTED AT EXECUTION

5677249 SPARCITY SHOULD BE SPARSITY, PLUS TECHNICAL ERRORS

5658777 RAC: OMBREGISTER: JAVA.SQL.SQLEXCEPTION: EXHAUSTED RESULTSET

5650875 DEPLOYMENT UPGRADE DOES NOT HANDLE COLUMN DEFAULT SYSDATE

5609212 TABLE PARTITIONS NOT ORDERED AT CODE GENERATION

5593312 RTC-5270 UPGRADE OF TABLE FAILS RPE-2081ALTER TABLE NO UPGRADE SCRIPT GENERATED

5575379 NO QUERY GENERATED UNDER QUERY TAB IN DATA OBJECT EDITOR FOR NEW VIEW

5563921 TABLE PARTITION DEPLOYABLE SET TO FALSE STILL DEPLOYS

5518205 REPO INSTALL FAILS DURING SEEDED DATA INSTALLATION ON 11GBETA1 RDBMS

5496418 EXTRACTION HINT IS REPEATED IN GENERATED CODE FOR MAPPING

5472814 UNABLE TO PASS DATE PARAMETERS USING RUN_TASK OR SQLPLUS_EXEC_TEMPLATE

5467660 CREATE THE TEST SCRIPT FOR EXPERT EXECUTION

5458229 ORA-12705: ENCOUNTERED WHILE TRYING TO LOGIN INTO OWB REPOSITORY

5403648 PLS-00103 WHEN DEPLOYING MAPPING WITH PRE-MAPPING OPERATOR

5394180 CHOOSE AN INPUT ATTRIBUTE THAT HOLDS THE HISTORIC DATE - ERROR

5383961 MIGR MAP'S FROM 10.1 TO 10.2 GENERATE ER VLD-1141 WHEN DEPLOYING

5338439 DPF-0009 UNABLE TO GENERATE DATA PROFILE WHEN LOGGED ON WITH OWB USER

5309802 CANNOT UPGARDE MDL FILE FROM 10.1.0.4 TO 10.2

5293048 CASE STATEMENT FAILS IF THERE ARE MORE THAN THREE WHERE CLAUSES

5292941 CANNOT COMPILE NESTED CASE FUNCTION IN BRACKETS IN EXPRESSION

5233636 TREE WALKING LOGIC WITHIN OWB MAP.

5218220, 5187382: CANNOT CREATE A FISCAL TIME DIM IN A LOCALE OTHER THAN ENGLISH

5051838 DERIVE FROM MANUALLY BOUND SNOWFLAKE DIMENSIONS

2627237 QUERY TEXT FROM INFORMIX VIEWS IS NOT SHOWN COMPLETELY

for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.