

Your Choice Regarding Cookies on this Site

By using this site, you accept that Oracle uses cookies to remember your log-in details, collect statistics to optimize site functionality, and deliver marketing based on your interests.

[Cookie Preferences](#)

Menu

Account

Country

Call

Oracle Technology Network Articles Business Intelligence

- Application Development Framework
- Application Express
- Big Data
- Business Intelligence
- Cloud Computing
- Communications
- Database Performance & Availability
- Data Warehousing
- Database
- .NET
- Dynamic Scripting Languages
- Embedded
- Digital Experience
- Enterprise Architecture
- Enterprise Management
- Identity & Security
- Java
- Linux
- Mobile
- Service-Oriented Architecture
- Solaris
- SQL & PL/SQL
- Systems - All Articles
- Virtualization

Migrating Oracle Discoverer Data to Oracle Business Intelligence 11g
by Edelweiss Kammermann

Plan your migration project, step by step.

Published February 2012

In January 2009, Oracle published a [statement of direction](#) for the Oracle Discoverer product strongly encouraging users to migrate to [Oracle Business Intelligence Enterprise Edition Plus](#) (OBIEE).

For organizations that have a significant investment in Discoverer and are considering such a migration, reusing as much of that work in OBIEE as possible is important. Doing so can minimize the impact on IT staff and end users, and also allow them to take full advantage of the functionality in OBIEE.

In this article, we will review the steps required to perform this migration while reusing much of the code implemented in Oracle Discoverer. As an overview, to accomplish the migration from Discoverer to OBIEE, we need to address the following objects:

The EUL (End User Layer), which contains all the metadata used by Discoverer to perform queries

The Discoverer Workbooks and worksheets containing queries

Migrating the EUL

As of OBIEE 10g (10.1.3.4), Oracle includes the Oracle Discoverer Metadata Conversion Assistant, a utility that lets you generate the BI repository file (.Rpd) from a Discoverer EUL export (.EEX) . The file is called MigrateEUL.exe and is found in the directory OracleBI/server/bin for OBIEE 10g, and in the Oracle_BI1/bifoundation/server/bin directory with OBIEE 11g.

The steps to perform this migration are:

Export of the EUL from Discoverer into a .EEX file.

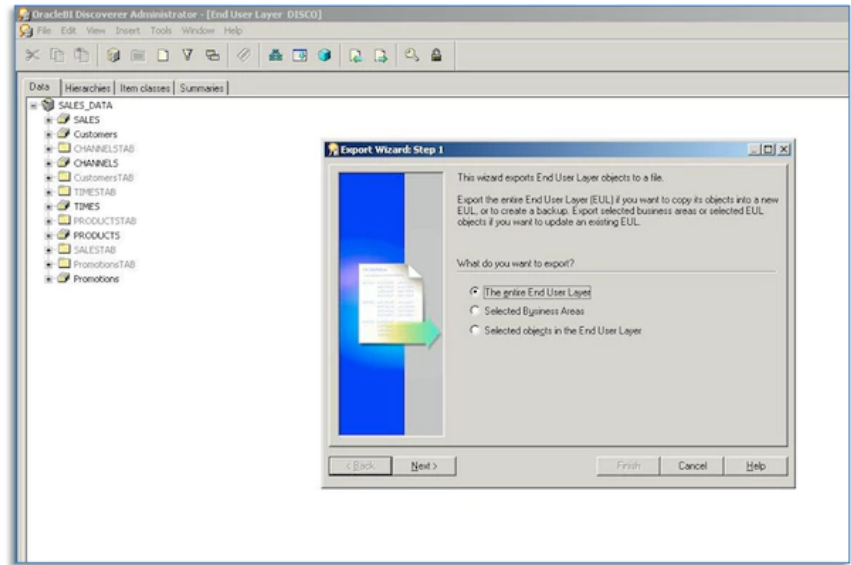
Run the utility in Oracle BI (either 10g or 11g) to convert the .EEX to a repository (.Rpd) file. If the target is OBI 11g (11.1.1.5), you must assign a password to the repository.

Enable the repository using Enterprise Manager if you are in 11g, or modify the NQSCfg.ini in version 10g.

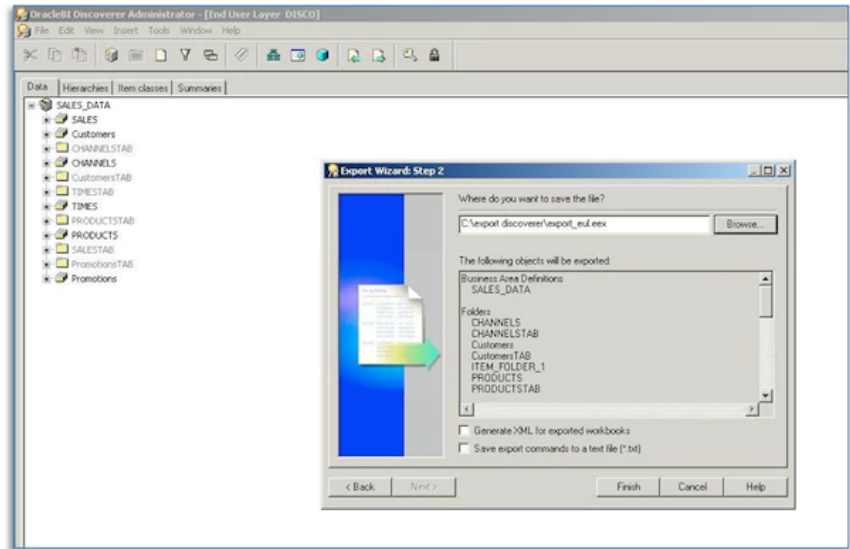
To demonstrate, we will migrate a Discoverer Version 10.1.2.1EUL to 11.1.1.5 Oracle BI repository. The EUL contains a typical star schema consisting of fact table, SALES, and five dimensions that are called CHANNELS, PRODUCTS, PROMOTIONS, CUSTOMERS, and TIME.

Export the EUL

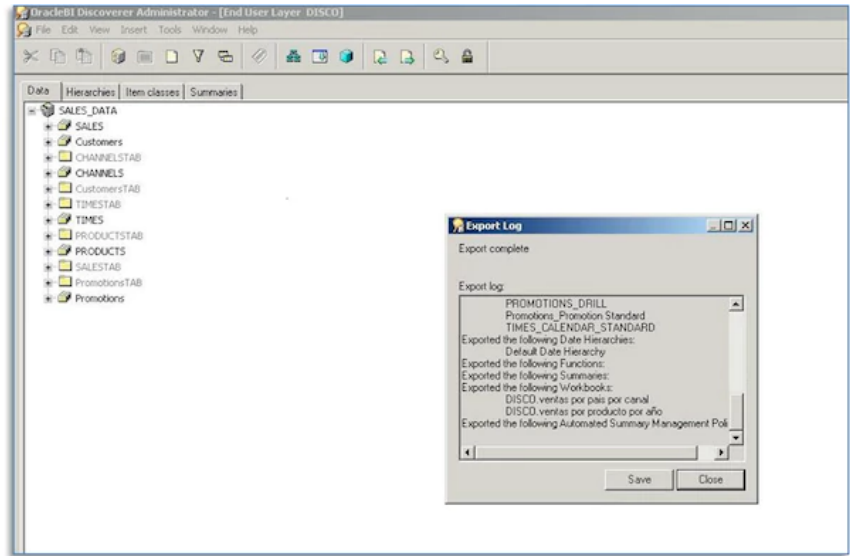
In Oracle Discoverer Administrator, go to the **File -> Export**. When you open the Export Wizard select the "The Entire End User Layer" option to export the entire EUL.



Press the **Next** button, and the "Step 2" dialog appears. Select the location and file name (.EEX) that will contain the export.



Then click **Finish** to create the export file. A progress window will be displayed, and you can capture the log as a text file to review at a later time.



Run the Oracle Discoverer Metadata Conversion Assistant

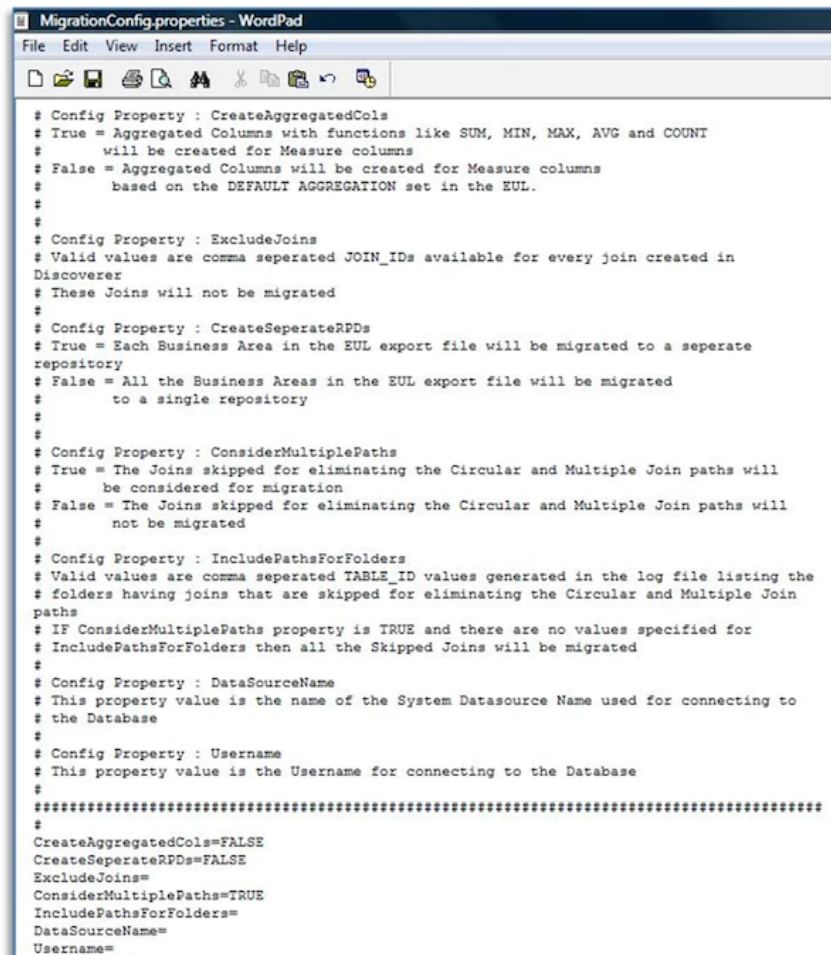
The Oracle Discoverer Metadata Assistant utility is run from command line. In the same directory or folder in which it is found (as described previously), there is also a configuration properties file called MigrationConfig.properties. We can refine the implementation of migration by settings in the properties file. (Note: the utility must be run on the same machine on which the OBIEE is installed.)

Documentation is found in file DiscovererMetadataConversionAssistant.pdf, which is located in the Oracle_BI1/bifoundation/server/document directory. This 30-page document provides more detail on migration and transformation of the respective objects.

Some flexibility is available in the migration tool, which can be adjusted using the configuration options found in the MigrationConfig.properties file. Some of the options are:

Property	Settings
CreateAggregatedCols	TRUE - Columns with aggregations like SUM, MIN, MAX, AVG and COUNT will be created for measure columns. FALSE - Aggregated Columns will be created for measure columns based on the DEFAULT AGGREGATION property set in the EUL
CreateSeparateRPDs	TRUE - Separate repository is generated for each business area. FALSE - All the business areas are migrated to a single repository.
ExcludeJoins	A comma separated list of JOIN_ID to be skipped during migration. To be used in conjunction with ConsiderMultiplePaths = TRUE
ConsiderMultiplePaths	TRUE - The migration assistant will accommodate the joins that would be otherwise skipped during migration. FALSE - The migration assistant will not accommodate the joins skipped during migration.
IncludePathsForFolders	A comma separated list of folder_id [available in the Filename.exception.log] for which the skipped joins must be accommodated during migration.
Connection pool parameters	DataSourceName, Username

The following image shows an example of the configuration file:



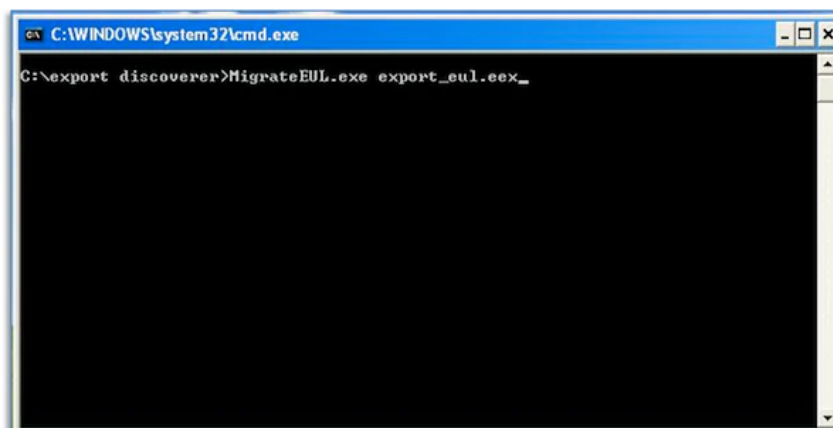
```

# MigrationConfig.properties - WordPad
File Edit View Insert Format Help

# Config Property : CreateAggregatedCols
# True = Aggregated Columns with functions like SUM, MIN, MAX, AVG and COUNT
# will be created for Measure columns
# False = Aggregated Columns will be created for Measure columns
# based on the DEFAULT AGGREGATION set in the EUL.
#
#
# Config Property : ExcludeJoins
# Valid values are comma separated JOIN_IDs available for every join created in
# Discoverer
# These Joins will not be migrated
#
# Config Property : CreateSeparateRPDs
# True = Each Business Area in the EUL export file will be migrated to a separate
# repository
# False = All the Business Areas in the EUL export file will be migrated
# to a single repository
#
#
# Config Property : ConsiderMultiplePaths
# True = The Joins skipped for eliminating the Circular and Multiple Join paths will
# be considered for migration
# False = The Joins skipped for eliminating the Circular and Multiple Join paths will
# not be migrated
#
# Config Property : IncludePathsForFolders
# Valid values are comma separated TABLE_ID values generated in the log file listing the
# folders having joins that are skipped for eliminating the Circular and Multiple Join
# paths
# IF ConsiderMultiplePaths property is TRUE and there are no values specified for
# IncludePathsForFolders then all the Skipped Joins will be migrated
#
# Config Property : DataSourceName
# This property value is the name of the System Datasource Name used for connecting to
# the Database
#
# Config Property : Username
# This property value is the Username for connecting to the Database
#
#####
#
CreateAggregatedCols=FALSE
CreateSeparateRPDs=FALSE
ExcludeJoins=
ConsiderMultiplePaths=TRUE
IncludePathsForFolders=
DataSourceName=
Username=

```

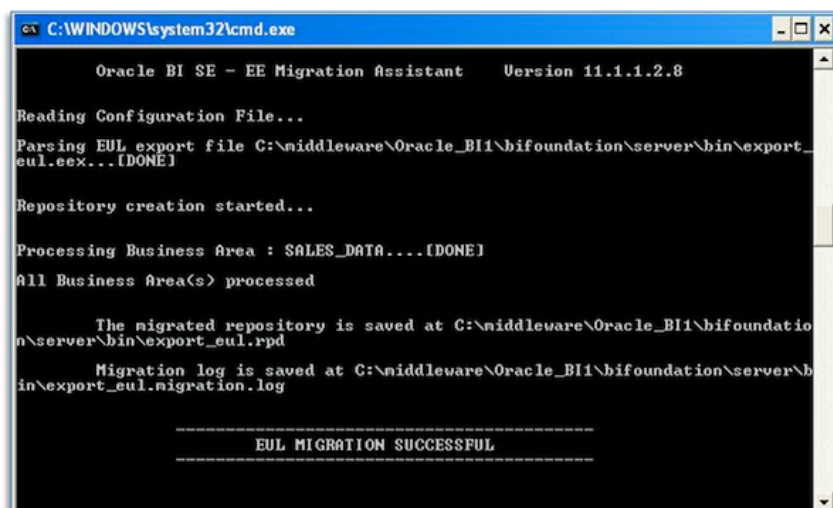
After modifying the configuration file, run the utility on the .eex file to export the EUL.



```

C:\WINDOWS\system32\cmd.exe
C:\export discoverer>MigrateEUL.exe export_eul.eex_

```



```

C:\WINDOWS\system32\cmd.exe

Oracle BI SE - EE Migration Assistant      Version 11.1.1.2.8

Reading Configuration File...
Parsing EUL export file C:\middleware\Oracle_BI1\bifoundation\server\bin\export_
eul.eex...[DONE]

Repository creation started...

Processing Business Area : SALES_DATA....[DONE]
All Business Area(s) processed

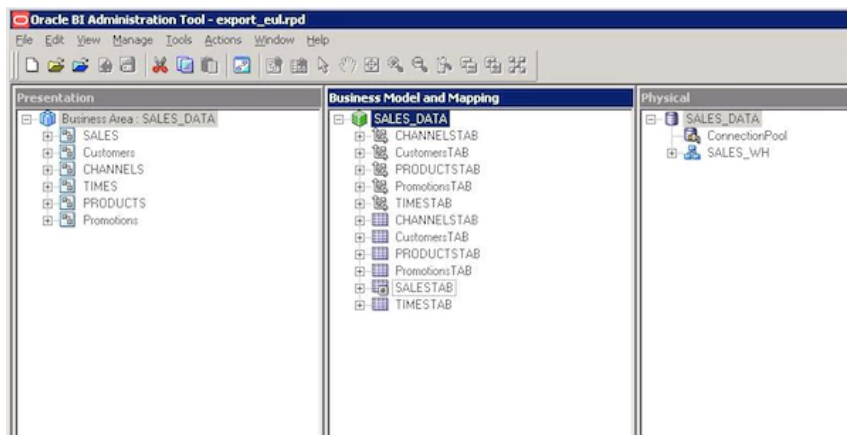
The migrated repository is saved at C:\middleware\Oracle_BI1\bifoundation
n\server\bin\export_eul.rpd
Migration log is saved at C:\middleware\Oracle_BI1\bifoundation\server\b
in\export_eul.migration.log

=====
EUL MIGRATION SUCCESSFUL
=====

```

Executing the Migration Assistant generates the BI Repository 11g (export_eul.rpd). In addition, it creates two log files: the export_eul.migration.log, which has information about the process of migration to high level; and export_eul.exception.log, which indicates the items that were not migrated.

To open the repository file we use the tool OBI Administration Tool and open the repository offline. The following figure shows how the repository is a result of migration.

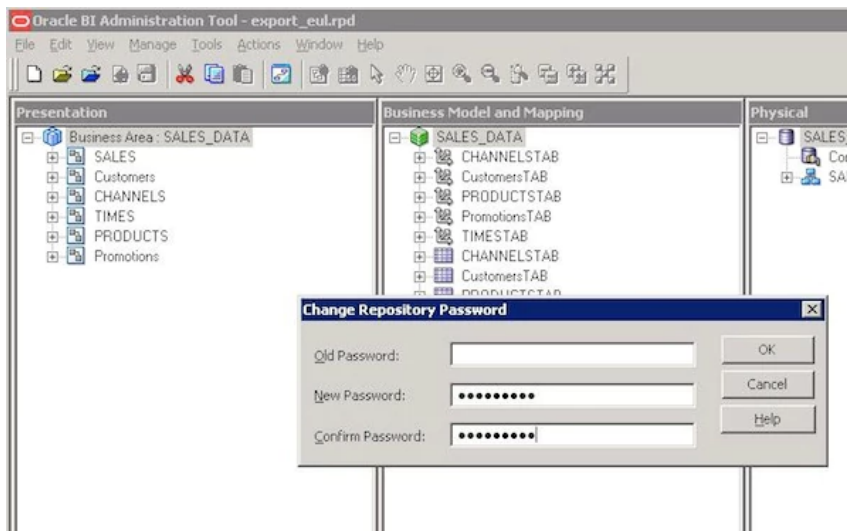


Assigning a Value to the Repository Password

As mentioned previously, as of OBIEE 11g version 11.1.5, the repository file is encrypted. A repository password is required in addition to the username and password with administrator privileges.

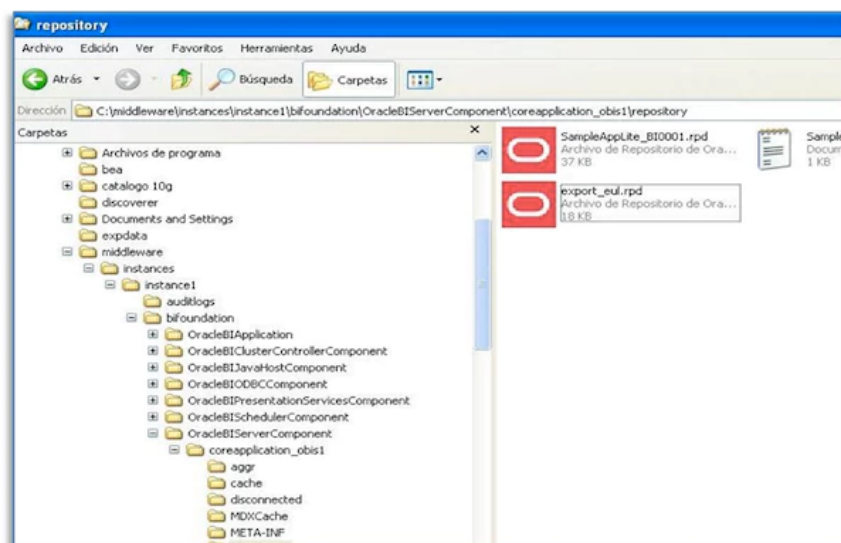
The new repository export_eul.rpd is assigned a null password that is generated by default. There is no option in the configuration file to change this prior to migration.

Since our repository has been created with a null password, to enable end user queries we need to assign a value. To do this task we must use the Oracle BI Administration Tool and open the repository in the offline mode. Once opened, go to **File -> Change Password**.

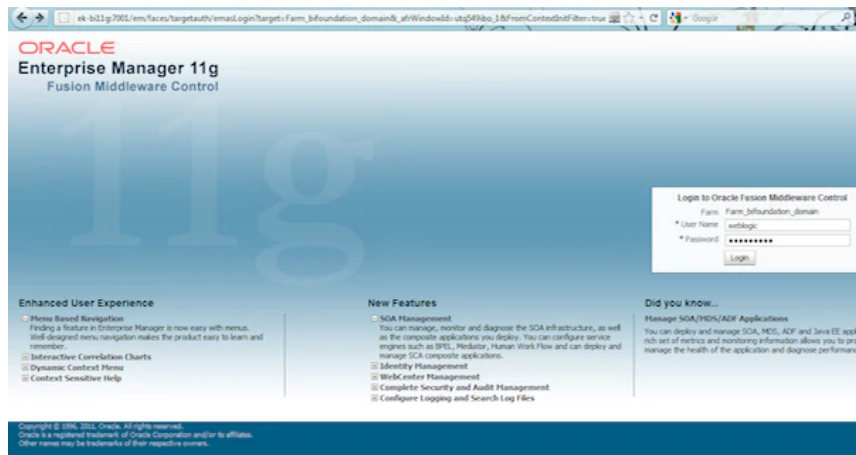


Enable the repository

OBIEE requires that active repositories be in a registered directory – by default, in instances\instance1\bifoundation\OracleBI\ServerComponent\coreapplication_obis1\repository. Therefore, to enable the repository we have just created, first copy/move it to that directory as shown in the following figure:



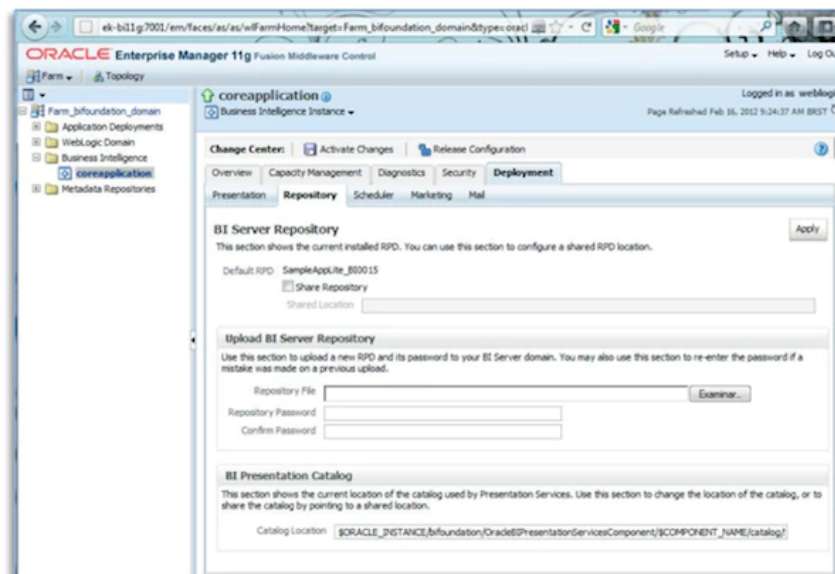
Once the newly created repository is copied to that directory, we can start Oracle Enterprise Manager 11g Fusion Middleware Control to enable the repository:



On the left navigation panel choose the instance of "Business Intelligence: core application".



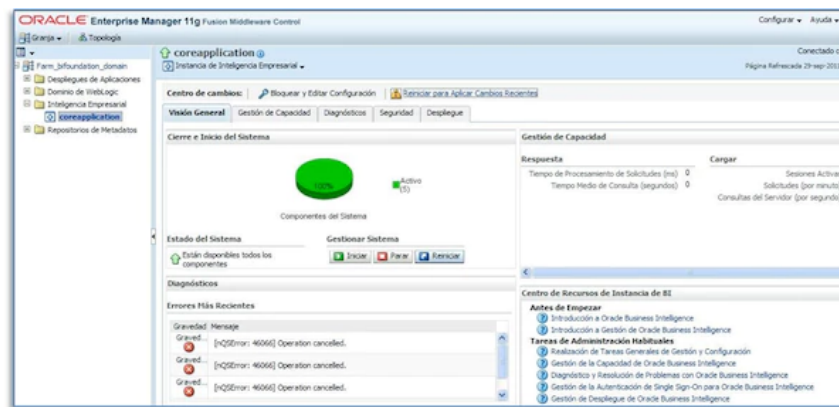
In the right panel select the Deployment tab and Repository option. To load the new repository, click the **Lock and Edit Configuration** button, which will allow us to modify the settings.



Then press the **Browse** button under the Upload BI Server Repository section, select the repository, and enter the password. Then press **Apply** and then **Activate Changes**. This will apply the changes.



After applying these changes we will see a message that tells us that we must restart the BI services for these changes to take effect.



We press the **Restart** button to restart services, which activates our changes. Then the new repository becomes visible and ready to be accessed by the users. (ote: To enable the repository in OBIEE 10g, the repository has to be saved in the OracleBI\server\repository folder. It is 'registered' as available by changing the NQConfig.ini found in OracleBI\server\config, adding the entry **Star = export_eul.rpd , DEFAULT** in the [REPOSITORY] section.)

Now that the EUL has been migrated, here are the elements that are affected:

Business Areas: Each Business Area is converted to a Subject Area in the presentation layer (visible to the user from OBI Answers).

Simple Folders: For each folder, a physical table is created in the physical layer and a logical table is created in the Business Model layer. When the simple folder has the property 'Visible to User' = Yes, a display table is created in the Presentation Layer of the repository.

Complex Folders: Each complex folder becomes a logical table in the logical layer, referencing Logical Table Source tables describing the joins between them. This complex folder will also join with those base folders which are dimensions. As in the case of simple folders, create a presentation table in the Presentation Layer if the property indicates it should be visible to the user.

Custom Folders: These folders are migrated to the physical layer of the repository as tables of type "Select". For each, a logical table in the business layer will be created and a table in the presentation layer will be created if it is set to be visible to the user.

Columns: Each column is created in the physical layer table to which it belongs and again in the presentation layer if the Presentation property is set to be visible to the user.

Joins: Each join remains as a foreign key in the physical layer and a logical join in the Business Model layer, with some exceptions:

- In the case of Circular Joins and Multiple Joins Path, the utility generates an alias for the table, as OBI does not allow them natively.
- A join between two tables is not migrated when they share between them more than one dimension in common.
- Duplicate joins (Folder A has a join to the folder B and vice versa) will be reduced to a single join, selected at random.
- Joins with calculations in the join condition are not migrated.

Conditions: Mandatory and Optional conditions are handled differently, as follows:

- **Mandatory:** If the condition is over a Simple or Custom folder, the condition is applied to the logical table source of that folder. For complex folders, conditions will be applied to OBIEE group "Everyone" as a security filter.
- **Optional:** There are not defined in the OBI repository and should be migrated manually to the catalog of OBI.

Hierarchy:

- Hierarchies based on Simple folders migrated; however, the ones based on Complex folders are not migrated.
- Date Hierarchies templates have no direct equivalent in Oracle BI. However the hierarchies result of use them are migrated.
- User privileges: All EUL users are migrated to the repository as part of the group "Everyone" and the initial password is the username in uppercase.

And here are the objects that are not affected:

The Item class: There is no equivalent in Oracle BIEE for the item class, therefore it can not be migrated. The lists of values in Oracle BIEE are generated at runtime when creating a filter.

Summary folders are not migrated.

Complex folders based on other complex folders are not migrated.

The system cannot migrate complex folders that are dimensions. A corresponding message will appear in the log.

Hierarchies based on Complex Folders are not migrated.

Optional conditions would need to be registered in the OBI catalog and are not part of the repository.

And here are the objects that are not affected:

The Item class: There is no equivalent in Oracle BIEE for the item class, therefore it can not be migrated. The lists of values in Oracle BIEE are generated at runtime when creating a filter.

Summary folders are not migrated.

Complex folders based on other complex folders are not migrated.

The system cannot migrate complex folders that are dimensions. A corresponding message will appear in the log.

Hierarchies based on Complex Folders are not migrated.

Optional conditions would need to be registered in the OBI catalog and are not part of the repository.

And here are the objects that are not affected:

The Item class: There is no equivalent in Oracle BIEE for the item class, therefore it can not be migrated. The lists of values in Oracle BIEE are generated at runtime when creating a filter.

Summary folders are not migrated.

Complex folders based on other complex folders are not migrated.

The system cannot migrate complex folders that are dimensions. A corresponding message will appear in the log.

Hierarchies based on Complex Folders are not migrated.

Optional conditions would need to be registered in the OBI catalog and are not part of the repository.

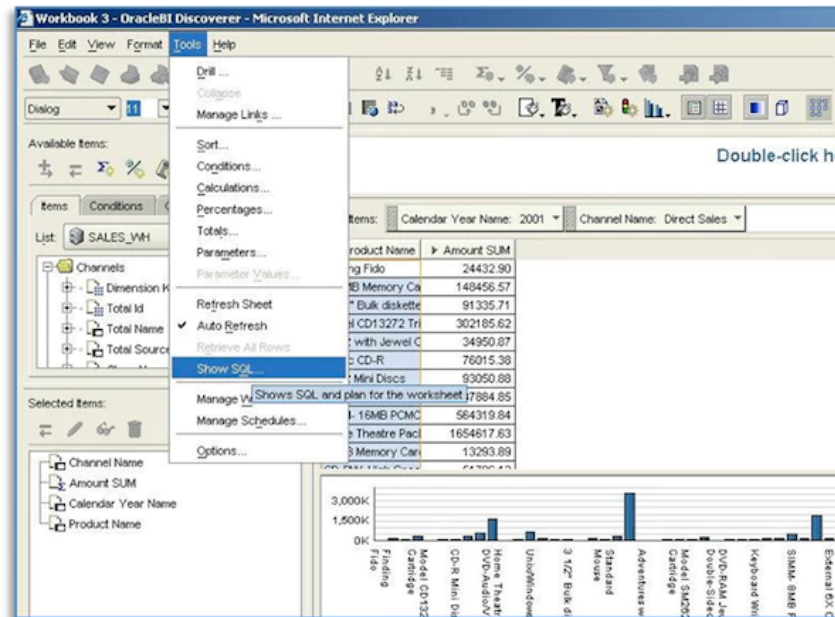
Migrating Workbooks

As of this writing, there is no tool available that would make workbook migration automatic. According to Oracle documentation, you need to re-create queries from scratch. That would force us to re-do the requests, which doubles the previous work.

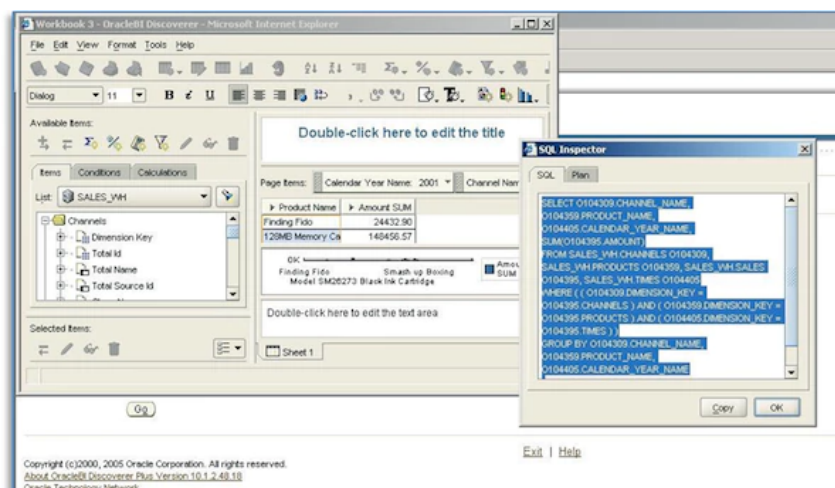
The alternative that this document proposes is to reuse the SQL queries generated by Discoverer and apply them directly into Oracle BI, significantly reducing the workload for creating queries. For this we apply the following steps for each of the Workbooks or Worksheets in Oracle Discoverer:

Open the workbook in Oracle Discoverer, using either Discoverer Plus or Discoverer Desktop.

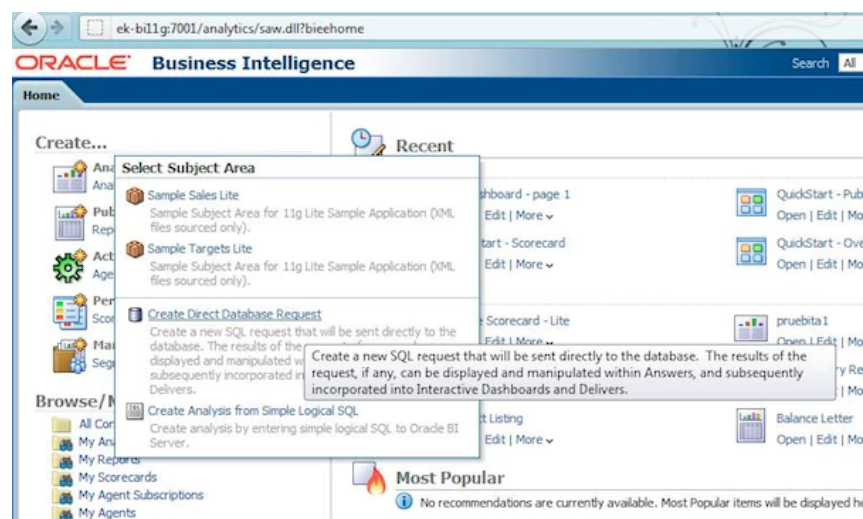
Once opened, go to the **Tools (Tools) -> Show SQL**.



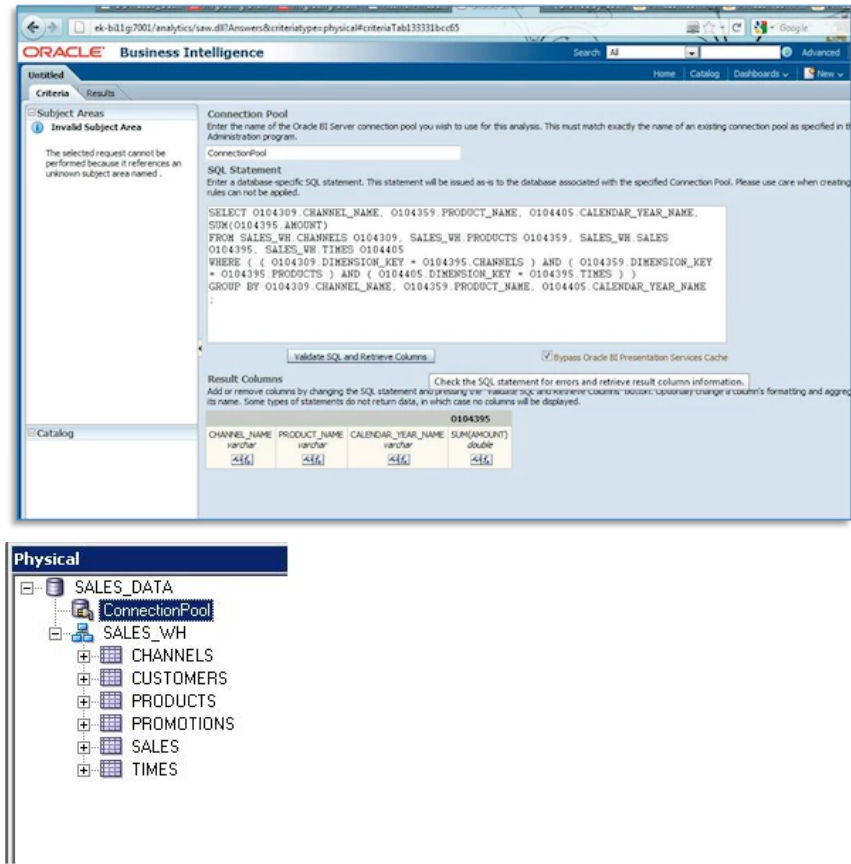
Choosing this option opens a window that displays the SQL statement that is generated by the workbook. Use the copy button in the window to copy the entire statement.



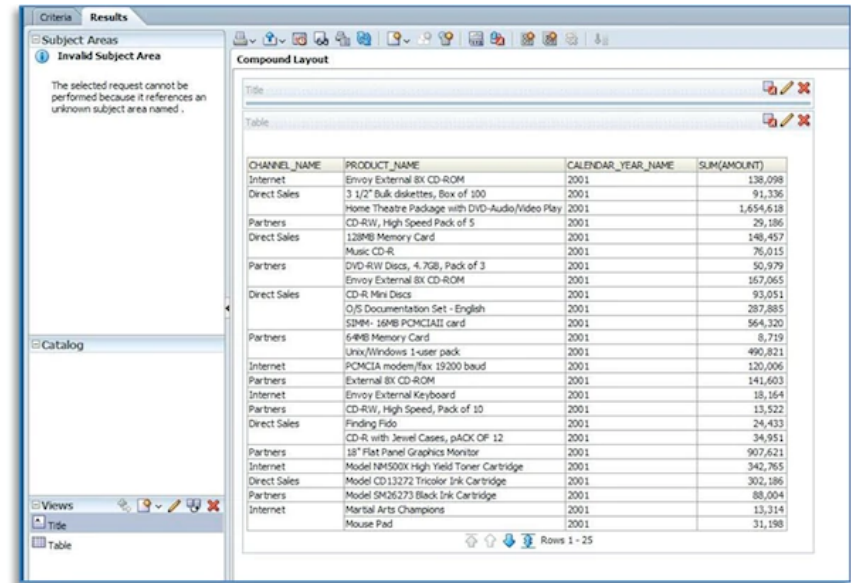
Go to OBIEE and select "Create Analysis" and then "Create Direct Database Request".



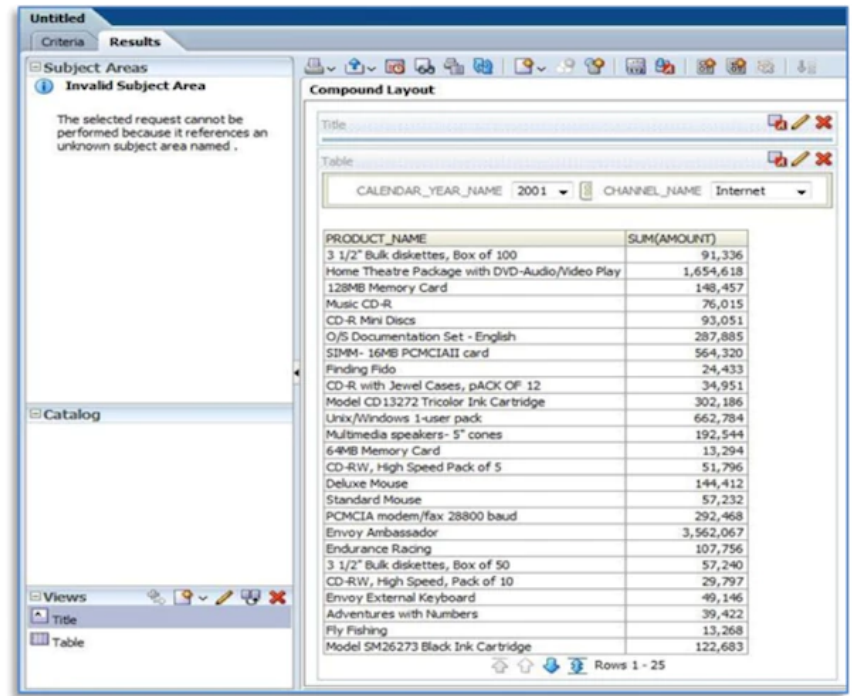
In the next window, enter the name of the Connection Pool and paste the SQL statement copied from Discoverer. Press the button **Validate SQL and retrieve columns** to select columns. (Note: the name and details of the Connection Pool are in the repository at the physical layer.)



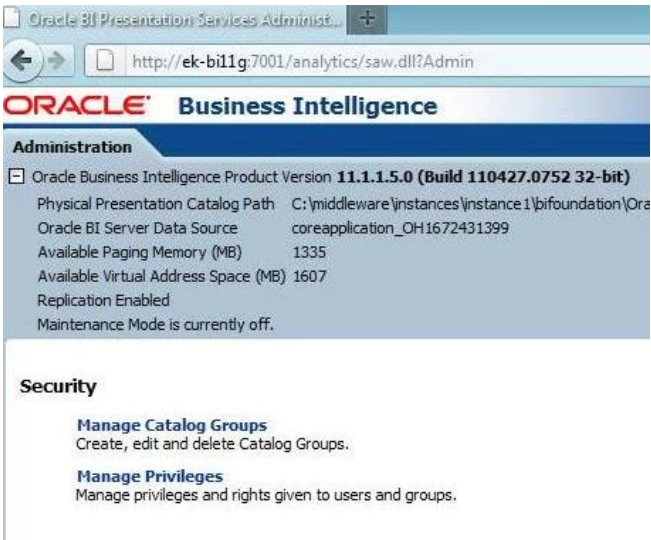
Switching to the Results tab displays the data for this query. In our case it becomes:



To get the results in the same format we had in Discoverer, simply drag the fields Channel Name and Calendar_Year_Name to the Table Prompts area.



It is important to remember that a user requires special privileges to perform a query directly against a database. The admin of the catalog would select the Manage Privileges option from the menu in OBIEE Administration.



Then, in the privileges section for Answers, give the follow permissions to the user:

Administration		
Manage Privileges		
Formatting	Save System-Wide Column Formats	BI Administrator Role
My Account	Access to My Account	BI Consumer Role
	Change Preferences	BI Consumer Role
	Change Delivery Options	BI Consumer Role
Answers	Create Views	BI Author Role
	Create Prompts	BI Author Role
	Access Advanced Tab	BI Author Role
	Edit Column Formulas	BI Author Role
	Save Content with HTML Markup	BI Administrator Role
	Enter XML and Logical SQL	BI Author Role
	Edit Direct Database Analysis	BI Administrator Role
	Create Analysis From Simple SQL	BI Administrator Role
	Create Advanced Filters and Set Operations	BI Author Role
	Save Filters	BI Author Role
	Execute Direct Database Analysis	BI Administrator Role

Important Considerations

Your migration is now complete! However, there are some important additional considerations.

For example, there is a fundamental difference between OBIEE and Discoverer in the preferred relationship model. In the Business Model layer of the OBIEE repository, the relationships between tables should represent a star schema - as a minimum requirement there is a dimension table and a fact table. In comparison, Oracle Discoverer doesn't demand any special type of relationship between data. For that reason, if the Discoverer data model doesn't represent star schemas, changes may be needed after migration in order to respect this structure. Otherwise the repository will not be available and enabled for queries.

Note also: although Oracle indicates that the metadata conversion utility only migrates to version 10g and that the Upgrade Assistant must be run to obtain a repository for version 11g, the repository version generated using the utility is actually in the correct format for 11g with version 11.1.1.3 and 11.1.1.5.

Edelweiss Kammermann is a BI and BPM specialist with over 15 years of consulting and project management experience in Oracle technology. Her work with both private and public organizations has spanned much of South America, especially Uruguay, Bolivia, Panamá and Honduras. Edelweiss is a very active member of the Oracle community; she is co-founder and vice president of the Oracle User Group of Uruguay (UYOUG).

 E-mail this page  Printer View

Contact Us

US Sales: +1.800.633.0738
Have Oracle Call You
Global Contacts
Support Directory

About Oracle

Company Information
Communities
Careers

Cloud

Overview of Cloud Solutions
Software (SaaS)
Platform (PaaS)
Infrastructure (IaaS)
Data (DaaS)
Free Cloud Trial

Events

Oracle OpenWorld
Oracle Code
JavaOne
All Oracle Events

Top Actions

Download Java
Download Java for Developers
Try Oracle Cloud
Subscribe to Emails

News

Newsroom
Magazines
Customer Success Stories
Blogs

Key Topics

ERP, EPM (Finance)
HCM (HR, Talent)
Marketing
CX (Sales, Service, Commerce)
Supply Chain
Industry Solutions
Database
MySQL
Middleware
Java
Engineered Systems