

Migrating Oracle BI Discoverer to Oracle BI Enterprise Edition

August 2008

*Includes details of using the Discoverer Metadata Conversion Assistant
shipped with Oracle BI EE version 10.1.3.4*

ORACLE®

Contents

CONTENTS	2
1. INTRODUCTION	4
2. WHY MIGRATE TO ORACLE BI ENTERPRISE EDITION?	5
2.1 ORACLE BI SERVER.....	5
2.2 ORACLE BI ADMINISTRATION TOOL	5
2.3 ORACLE BI INTERACTIVE DASHBOARDS	6
2.4 ORACLE BI ANSWERS.....	6
2.5 ORACLE BI DELIVERS	6
2.6 ORACLE BI PUBLISHER.....	6
2.7 ORACLE BI DISCONNECTED ANALYTICS	6
2.8 ORACLE BI APPLICATIONS	6
3. DISCOVERER METADATA CONVERSION ASSISTANT	7
3.1 SUITABILITY OF THE CONVERSION ASSISTANT	7
3.2 PRE-REQUISITES	7
3.3 INSTRUCTIONS FOR USING THE CONVERSION ASSISTANT	7
3.3.1 Export Discoverer metadata.....	7
3.3.2 Using the command line executable	8
3.3.3 Migration Log Files	8
3.3.4 Configuration options	10
3.4 DESCRIPTION OF OUTPUT FROM THE CONVERSION ASSISTANT	10
3.4.1 End User Layer	10
3.4.2 Business Areas	10
3.4.3 Simple Folders	10
3.4.4 Complex Folders.....	11
3.4.5 Custom Folders	12
3.4.6 Items.....	12
3.4.7 Joins	14
3.4.7.1 Multiple Join Paths	15
3.4.7.2 Circular Joins	16
3.4.7.3 Duplicate Joins.....	17
3.4.7.4 Notes on Migrating Joins	18
3.4.8 Conditions	18
3.4.8.1 Mandatory Conditions	19
3.4.8.2 Optional Conditions	19
3.4.9 Aggregate Calculated Items.....	19
3.4.10 Item hierarchies.....	20
3.4.11 Discoverer Date Hierarchies	20
3.4.12 Item Classes	20
3.4.13 Summary Folders	20
3.5 HOW THE MIGRATION ASSISTANT CONVERTS DISCOVERER METADATA	21
3.5.1 Measures	21
3.5.2 Facts and Dimensions	21
3.5.3 Key Columns	21
3.5.4 Migrating Folders without Joins	21
3.5.5 Creation of Alias Objects.....	22
3.5.6 Aggregates and Calculations.....	22

3.5.7	User Privileges	22
3.6	EVALUATING THE OUTPUT FROM THE ASSISTANT	22
3.6.1	Use of the EUL_DATE_TRUNC function.....	23
3.6.2	Joins Involving Different Data Types.....	23

1. Introduction

This document provides a single source of information about migrating an Oracle BI Discoverer system to Oracle BI Enterprise Edition (Oracle BI EE).

Oracle BI EE is an innovative and comprehensive BI platform that delivers the full range of BI capabilities on a next-generation architecture designed for true enterprise deployment. It enables organizations to define a single, logical view of all enterprise data, whether in a single data warehouse or across multiple operational and analytic sources. Business users benefit from new levels of self-sufficiency to access, interact with, and utilize this information to increase effectiveness.

Pervasive – Oracle BI EE offers a highly intuitive and interactive 100% Web UI that delivers rich and powerful BI from across multiple sources to any audience, providing insight where and when users need it to drive actions, decisions, and business processes. All without requiring special BI expertise or power user skills

Comprehensive without Compromise – Oracle BI EE offers the full range of BI capabilities, including interactive intelligence dashboards, ad hoc analysis, proactive detection and alerts, intelligent workflow, enterprise reporting, disconnected analytics, MS Office integration, real-time predictive analytics, and more. It provides innovative next-generation capabilities, including a model-driven enterprise semantic layer, heterogeneous data access and integration, modern Services Oriented architecture, and mission critical scalability and performance.

Hot-Pluggable – Oracle BI EE can leverage your existing IT investments, including Oracle and non-Oracle databases, applications, and custom solutions.

2. Why Migrate to Oracle BI Enterprise Edition?

It is assumed that readers of this document will be familiar with the components of Oracle BI Discoverer. This section describes the comparable components of Oracle BI EE and highlights the functional benefits.

2.1 Oracle BI Server

The Oracle BI Server sits at the heart of Oracle BI EE generating optimized queries across multiple data sources and presenting end users with a single, easy to use, logical model of their business. This can benefit end users in many ways such as being able to present a single graph with order data, receivables data and forecast all coming from different sources. In addition, the Oracle BI Server can act as an ODBC data source, allowing querying of the metadata through tools such as Excel or other third party applications.

The Oracle BI Server includes a multi-user data cache that provides high scalability and improved query performance. Cache administration is carried out from the Oracle BI Administration Tool where metadata objects can be flagged as being cacheable in the physical layer. Other benefits of the Oracle BI Server include clustering capability enabling high availability and ability to use a variety of enterprise security models.

2.2 Oracle BI Administration Tool

The metadata that is used by the Oracle BI Server is created using the Oracle BI Administration Tool. This is comparable to Discoverer Administrator. The Oracle BI EE metadata is contained within a file known as an .RPD file. This is the equivalent of a Discoverer EUL. A Discoverer business areas is roughly equivalent to a presentation layer catalog that are called subject areas within Oracle BI Answers.

There is a key difference between the metadata models of the two products whereby BI EE metadata consists of three layers. A **Physical Layer** contains the mappings to the objects that hold the data e.g. a database, MS Excel spreadsheet, XML file etc. This layer also contains information about how the mappings relate to each other in the form of primary/foreign keys. The **Logical or Business Model Layer** contains mappings of how objects in the physical layer relate to each other. Key differences with Discoverer are the creation of dimensions and facts leading to the ability to define level based measures. The **Presentation Layer** contains the view of the metadata that the end users see. It is possible to view a diagrammatic representation of both the physical model (i.e. database model) and logical business model layers. The benefit of this is a clear and quick understanding of how the metadata objects are related.

The Oracle BI Administration tool provides support for multi-user BI metadata administration (including a check in/out model) and multi-language metadata.

2.3 Oracle BI Interactive Dashboards

Oracle BI Interactive Dashboards provide a fully interactive collection of analytic content with a rich variety of visualizations in a pure thin client. Guided navigations and alerts drive the user to greater insight to take the right actions at the right time for their business. This is comparable to the Discoverer Portlet provider. The BI Interactive dashboards are easier to use and offer a greater presentation and visualization capabilities.

2.4 Oracle BI Answers

Oracle BI Answers is a pure thin client ad-hoc query and analysis tool with a simple point and click interface for building queries. Users interact with a business friendly view of their data. This is comparable to Discoverer Plus and Viewer. In Answers, a query is known as a request. This is equivalent to a Discoverer worksheet. Answers requests are stored in a catalog either in a user's personal area or in a shared area. There is full support in the catalog for organizing requests in folders and subfolders.

2.5 Oracle BI Delivers

Oracle BI Delivers enables pro-active notification, monitoring and alerting and report distribution to multiple channels such as email, dashboard alerts and mobile devices. It includes a web based interface for creation of alerts (known as iBots) and the ability to execute other iBots, scripts or Java programs to build up analytic workflows. There is no comparable alerting and distribution functionality in Discoverer other than the ability to schedule worksheets and share the results with other users of the system.

2.6 Oracle BI Publisher

Oracle BI Publisher enables pixel-perfect reports to be generated using Answers requests as the source. MS Word or Adobe Acrobat can be used to layout reports and create templates. There is no equivalent functionality in Discoverer. Note that BI Publisher also integrates with Discoverer as a source.

2.7 Oracle BI Disconnected Analytics

Oracle BI Disconnected enables the capabilities of Interactive Dashboards and Answers but available offline via data synchronization between the mobile client and central BI Server. There is no equivalent in Discoverer.

2.8 Oracle BI Applications

The Oracle BI Applications are a complete prebuilt BI solutions that deliver intuitive, role based intelligence for everyone in an organization. They consist of ETL adapters, metadata, dashboards and KPIs built to industry best practices to gain insight from applications including Siebel, Oracle E-Business Suite, PeopleSoft Enterprise, JD Edwards and SAP.

3. Discoverer Metadata Conversion Assistant

This section describes and provides instructions for using the Discoverer Metadata Conversion Assistant (Assistant). This is a command line utility that greatly accelerates the migration of Discoverer metadata from the EUL to Oracle BI EE.

3.1 Suitability of the Conversion Assistant

There are several types of system that Discoverer can be used to report against:

- Data warehouse e.g. star schema. This is the optimum type of metadata to use with the conversion assistant and due to the nature of the Oracle BI EE business model layer generates metadata that in the majority of cases needs few manual changes before being able to create queries.
- Custom built OLTP schema. This type of metadata may need some manual changes after being run through the conversion assistant as the Discoverer metadata can contain join mappings that do not directly convert to the Oracle BI EE business model layer e.g. multiple join paths between folders and circular joins. The conversion assistant can cope with these scenarios by the optional creation of additional objects in the BI EE metadata logical layer e.g. alias dimension or fact tables. Further details can be found later in this document.
- Discoverer reporting on Oracle Applications e.g. Oracle BIS or custom built systems. Oracle recommends customers evaluate the Oracle BI Applications in these scenarios as they offer the fastest time to value using prebuilt data warehouse star schemas, ETL routines and Oracle BI EE metadata and dashboards built to best practices.

3.2 Pre-requisites

The **minimum** version of Discoverer EUL that can be used with the Assistant is **10.1.2.n**. The version of the RPD file that is generated is dependent upon the installed version of Oracle BI EE that the Assistant is run against but must be a minimum of 10.1.3.4. Note that earlier versions of the Oracle BI Administration tool cannot open RPD files that have been generated with the Assistant running against later versions.

3.3 Instructions for using the conversion assistant

3.3.1 Export Discoverer metadata

The first stage in the migration process is to export the Discoverer metadata you wish to convert to an .eex file using the Discoverer Administration tool. This document assumes familiarity with the Discoverer product but details of this process can be found online in the Discoverer Administration Guide:

http://download.oracle.com/docs/pdf/B13916_04.pdf

3.3.2 Using the command line executable

The Assistant is provided as a command line executable that must be run on the same machine as an Oracle BI EE installation. The executable file 'MigrateEUL.exe' can be found in the following directory of an Oracle BI EE install: OracleBI\server\bin

The format for using the command line is as follows:

```
> MigrateEUL.exe filename.eex
```

Where 'filename.eex' is the name of the Discoverer Administrator export file (.eex file) that was generated in the previous step. Note that the RPD file that the Assistant generates will be created in the directory containing the Discoverer EUL export file.

After the RPD file has been generated, open this file using the Oracle BI EE Administration Tool to update the connection pool settings to point to the data source being used, run a consistency check and save these changes to the RPD file.

3.3.3 Migration Log Files

The migration assistant generates two log files that contain the following information:

<name of original file>.migration.log - captures the progress of migration at a high level

<name of original file>.exception.log - captures the items that could not be migrated such as skipped joins and folders.

For each of the business areas that are converted, these files contain the following details:

- The name of business area being processed
- Details of any joins in the Discoverer metadata that may be skipped due to the existence of multiple join paths. For each skipped join, a 'folder ID' is generated that can be used as a reference for a second run of the Assistant with the required folder IDs added to the 'MigrationConfig.properties' file in order to generate folder aliases. The folder IDs are listed with the configuration file option 'IncludePathsForFolders'.

An example of the log file output is included below:

```
Repository creation started...
*****
                Business Area : Circular Joins Example
*****
-- Processing EUL Joins....
-- Processing Folder(s) based on EUL Joins....
** CASE OF MULTIPLE JOIN PATH
```

```

Join : Time -> Inventory has been skipped for folder Inventory
## List of Folder(s) with SKIPPED JOINS / PATHS
FOLDER_ID      SELECTED = ###      FOLDER_NAME      SKIPPED JOIN_NAME
-----
1      ---      "Inventory"      Join : Time -> Inventory
-- Processing Complex Folder(s)....
-- Processing Dimension(s)....
***** CREATING PHYSICAL LAYER *****
-- Creating Database...[DONE]
-- Creating Connection Pool...[DONE]
-- Creating Physical Table(s)...
-- Physical Table(s) creation...[DONE]
-- Creating Physical Join(s)...
-- Physical Join(s) creation...[DONE]
***** PHYSICAL LAYER CREATION DONE *****
***** CREATING LOGICAL LAYER *****
-- Creating Subject Area...[DONE]
-- Creating Logical Table(s)...
-- Creating Logical Join(s)...
-- Logical Join(s) creation...[DONE]
-- Creating Calculation(s)...
-- Creating Complex Folder Calculation(s)...
-- Calculation(s) creation...[DONE]
-- Logical Table(s) creation...[DONE]
-- Creating Dimension(s)...
Dimension : Shipments: Ship date: Default Date Hierarchy
Dimension : Time: Date key: Default Date Hierarchy
-- Dimension(s) creation...[DONE]
***** LOGICAL LAYER CREATION DONE *****
***** CREATING PRESENTATION LAYER *****
-- Creating Presentation Folder(s)...
Folder : Inventory
Folder : Shipments
Folder : Time
-- Presentation Folder(s) creation...[DONE]
***** PRESENTATION LAYER CREATION DONE *****
----- All Business Area(s) processed -----
-- Creating Security Filter(s)...
-- Security Filter(s) creation...[DONE]
-- Creating User(s)...
MIGRATE
PUBLIC
VIDEO5
-- User(s) creation...[DONE]
      The migrated repository is saved at
D:\migration\CircularJoins\circular_join.rpd
-----
                        EUL MIGRATION SUCCESSFUL
-----

```

3.3.4 Configuration options

A configuration file **MigrationConfig.properties** is provided for the user to control the migration behavior. The options available are as below:

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
CreateSeperateRPDs	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

3.4 Description of output from the conversion assistant

This section describes the mappings used by the metadata conversion assistant for the metadata properties in Discoverer Administrator. Where there is no equivalent property in Oracle BI EE this is indicated by 'n/a'. It also describes any assumptions that the conversion assistant makes when creating the three layers that make up the Oracle BIEE metadata model.

3.4.1 End User Layer

The Discoverer End User Layer (EUL) maps to a BIEE metadata repository file (.rpd file).

3.4.2 Business Areas

Business Areas map to a presentation catalog also known as a subject area in BI Answers.

3.4.3 Simple Folders

Discoverer simple folders are mapped against database tables or views. They correspondingly migrate to physical tables in the physical layer, logical tables in the

business model layer and to presentation tables in the presentation layer. The tables only migrate to the presentation layer if they are set to be visible to users in Discoverer. The following table summarizes how the properties are migrated.

Discoverer Property	Target Property In Presentation Layer	Target Property In Logical Layer	Target Property In Physical Layer
Name	Table name	Table name	Physical table
Description	Table description	Table description	Table description
Visible to User	Will not appear in this layer if set to NO	n/a	n/a
Database	n/a	n/a	DB user in connection pool
Owner	n/a	n/a	Physical layer schema
Object	n/a	n/a	Physical layer table
Optimizer Hints	n/a	n/a	Physical table 'hint' property in 'general tab of table properties
Identifier	n/a	n/a	n/a

3.4.4 Complex Folders

Discoverer complex folders are used to combine items from multiple simple folders. The analogy often used to describe complex folders is that of a database view. From a Discoverer administration perspective, they are useful for combining items into logical groups to simplify the end users view of the metadata. From an end users perspective, this can make life easier as they only need to go to a single folder to get all the items they need for a report rather than multiple folders.

Discoverer Property	Target Property In Presentation Layer	Target Property In Logical Layer	Target Property In Physical Layer
Name	Table name	Table name	n/a
Description	Table description	Table description	n/a
Visible to User	Will not appear in this layer if set to NO	n/a	n/a
Optimizer Hints	n/a	n/a	n/a
Identifier	n/a	n/a	n/a

Complex folders will be mapped to a logical table in the Logical Layer having a Logical Table Source that contains the base folders and the joins between them. This folder in turn will be joined to those base folders, which are Dimensions.

Complex folders appear in the Presentation Layer with the following mapping:

- The item references of the complex folder will be picked from the respective base folders in the logical layer.
- For creating the Discoverer ‘admin calculations’ within a complex folder, a logical table corresponding to the complex folder will be created in the logical layer.. Those calculations involving items from more than one base table will be created in the Complex Folder. The Logical Folder will then be moved to the presentation layer. However, if the ‘admin calculations’ are based on a single base folder, they will be migrated to the corresponding Logical folder and not the Complex Folder.

Note that complex folders based on items from another complex folder cannot be migrated automatically.

3.4.5 Custom Folders

Custom folders enable flexibility in folder creation – e.g. SQL statements using set operators (e.g. UNION, INTERSECT, MINUS). On entering a SQL statement in the UI for creating a custom folder, a folder is created containing items that are referenced in the SQL statement. In Oracle BI EE the custom folders are migrated to the physical layer with a table type of ‘select’. This is also known as an opaque view.

Discoverer Property	Target Property In Presentation Layer	Target Property In Logical Layer	Target Property In Physical Layer
Name	Table name	Table name	Table name (type = select)
Description	Description	Description	Description
Visible to User	Will not appear in this layer if set to NO	n/a	n/a
Valid	Will not appear in this layer if not valid	n/a	Only created if its valid
Custom SQL	n/a	n/a	Default initialization string
Identifier	n/a	n/a	n/a

3.4.6 Items

Items are the basic building block for queries. They are mapped to columns in database tables or views or created from calculations in Discoverer Administrator. These calculations can be based upon PL/SQL functions. Discoverer items migrate to physical columns in the physical layer, logical columns in the business model layer and presentation columns in the presentation layer of Oracle BI EE. Only items that are not hidden from end users will appear in the presentation layer.

Discoverer Property	Target Property In Presentation	Target Property In Logical Layer	Target Property In Physical
---------------------	---------------------------------	----------------------------------	-----------------------------

	Layer		Layer
Name	Column name	Column name	Column name
Auto generate name	n/a	n/a	n/a
Description	Description	Description	Description
Data type	n/a	Datatype	Datatype
Formula	n/a	Column formula in the logical table source (LTS) physical mapping	n/a
Database column	n/a	n/a	n/a
Visible to user	Will not appear in this layer if set to NO	n/a	n/a
Item class	n/a	n/a	n/a
Date hierarchy	n/a	n/a	n/a
Default position	n/a	n/a	n/a
Default aggregate	n/a	Default aggregation rule for the column.	n/a
Heading	n/a	n/a	n/a
Format mask	n/a	n/a	n/a
Alignment	n/a	n/a	n/a
Word wrap	n/a	n/a	n/a
Case storage	n/a	n/a	n/a
Display case	n/a	n/a	n/a
Default width	n/a	n/a	n/a
Replace NULL with	n/a	n/a	n/a
Content type	n/a	n/a	n/a
Alternative display value	n/a	n/a	n/a
Max char fetched	n/a	n/a	n/a
Indexed item	n/a	n/a	n/a
Identifier	n/a	n/a	n/a

In Discoverer, when building a query, end users are able to use either the default aggregation for an item or select from a list of available aggregation functions. In Oracle BI EE metadata, it is possible to specify the default aggregation for a particular column in the logical layer, however this aggregation cannot be changed (in a similar manner to Discoverer) during the creation of the answers worksheet. The user needs to create another column and define the required aggregation.

In order to offer end users a similar experience for selecting aggregations the migration has an option to create a separate column in the logical layer for each default aggregation that is supported in Discoverer. This configuration option is CreateAggregatedCols. If set to TRUE while migrating, all aggregations supported by

Discoverer will be generated. If set to FALSE then a column with its aggregation function set to the Discoverer 'default aggregation' will be created.

Discoverer calculated items based on Oracle PL/SQL functions or Oracle analytic functions will be migrated to Oracle BI EE metadata that uses the EVALUATE and EVALUATE_AGGR functions. These calculations will be created as a logical column with its formula being set in the Physical Mapping of the Logical Table Source.

3.4.7 Joins

Joins in Discoverer metadata define relationships between the folders that are used for building queries. Usually joins are defined using the corresponding key columns of the underlying database objects. Due to the differences in the metadata models between Discoverer and Oracle BIEE there are some differences in the types of joins that can be migrated automatically.

It is in this area of metadata that the differences between Discoverer and Oracle BIEE metadata become apparent. The main difference is that the logical business model layer needs to be based around one or more star schema models (this is a common data model for data warehouse design). A variation on the star schema model is known as a snowflake model (commonly represents the hierarchy levels of a dimension as separate tables). In these cases, the migration assistant 'collapses' the snowflake dimensions to their lowest level of dimension above the fact table.

The physical metadata layer does not need to be modeled around a star schema so this layer is created using the join information from the Discoverer metadata.

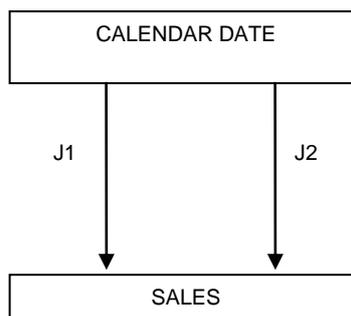
Discoverer Property	Target Property In Presentation Layer	Target Property In Logical Layer	Target Property In Physical Layer
Name	n/a	Logical join name	Physical foreign key name
Auto generate name	n/a	n/a	n/a
Description	n/a	n/a	n/a
Master	n/a	Cardinality = 1	The master table is the one containing a primary key
Detail	n/a	Cardinality = N	The detail table is the one containing the foreign key
Formula	n/a	n/a	Expression for the foreign key relationship
Outer join on	n/a	Type Default Join Type is InnerJoin OuterJoinOnDetail is mapped as LeftOuterJoin	Type. It is always set automatically and cannot be changed in the Oracle BI EE Admin

		OuterJoinOnMaster is mapped as RightOuterJoin	
Join details	n/a	n/a	n/a
Join relationship	n/a	<p>Cardinality.</p> <p>OneToOne join relationships are migrated as 1:N join since 1:1 joins are not supported</p> <p>DetailItemValuesAlwaysExistInMaster is mapped as 1:N</p> <p>DetailItemValuesMightNotExistInMaster is mapped as 0,1:N</p>	<p>Cardinality.</p> <p>It is always set automatically and cannot be changed in the Oracle BI EE Admin</p>
Identifier	n/a	n/a	n/a

3.4.7.1 Multiple Join Paths

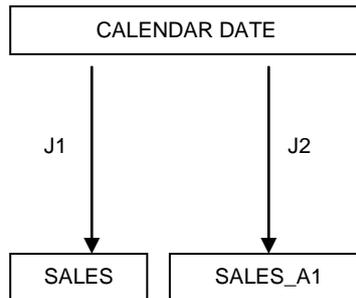
Discoverer folders having Multiple Join paths to another folder will be supported by creating object aliases that are based on the same underlying physical object but have the required alternative join paths.

In the case of multiple joins to the same folder, the detail folder will be aliased:

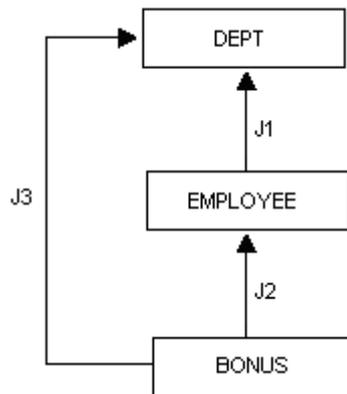


Join Name	Join Condition
J1	CALENDAR_DATE.DATE_COL = SALES.TRANSACTION_DATE
J2	CALENDAR_DATE.DATE_COL = SALES.ORDER_DATE

This alias folder SALES_A1 is created by the conversion assistant to accommodate the join J2 to CALENDAR_DATE:

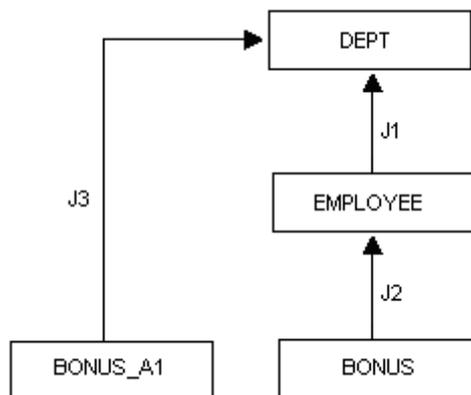


Multiple paths exist to the same folder:



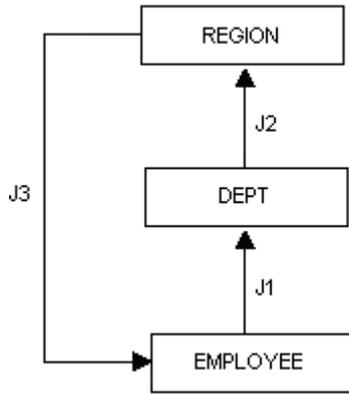
Join Name	Join Condition
J1	EMPLOYEE.DEPT_NO = DEPT.DEPT_NO
J2	BONUS.EMP_ID = EMPLOYEE.EMP_ID
J3	BONUS. DEPT_NO = DEPT.DEPT_NO

The BONUS folder is aliased to accommodate the join J3 to DEPT.



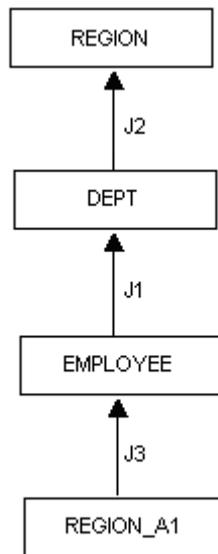
3.4.7.2 Circular Joins

In this case, a circular join path exists between the folders:



Join Name	Join Condition
J1	EMPLOYEE . DEPT_NO = DEPT.DEPT_NO
J2	DEPT.REGION_ID = REGION.REGION_ID
J3	REGION.REGION_HEAD = EMPLOYEE . EMP_ID

The REGION folder is aliased (REGION_A1) to accommodate the join J3 to EMPLOYEE.



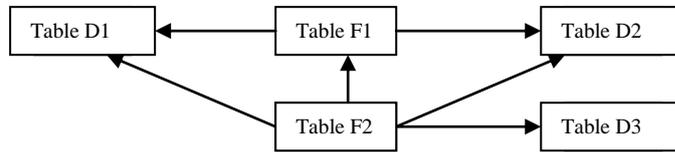
3.4.7.3 Duplicate Joins

In the case where duplicate join definitions exist in the Discoverer metadata, only one of the duplicate joins created in Discoverer will be considered for migration. Any duplicate joins that have been detected will be noted in the migration log files.

3.4.7.4 Notes on Migrating Joins

This section describes how various joins are treated by the migration assistant due to differences between Discoverer and the Oracle BI Server.

- The join between any two folders is skipped when they share more than one common dimension, for example see below:



This heuristic is applied to simplify the model..In the case above, table F1 and F2 are better treated as facts and D1, D2,D3 as dimensions in a classical star schema. The join between F1 and F2 is redundant for most practical situations. In the cases where it is determined to be important, it can be manually restored by aliasing table F2.

- Folder A has a join to Folder B and vice-versa. In such a scenario one of the joins will be skipped randomly. The skipped join is not available for further inclusion
- Joins involving calculations in the join condition will not be migrated.
- Only one join path is available between the component folders of a complex folder
- In the case of non-equi joins in Discoverer metadata, the resulting object in the physical layer is a complex join
- Oracle BIEE does not support 1:1 joins

In the following cases, Oracle recommends checking the results of the migration assistant:

- Joins based on columns where the column in the master and the detail tables have different data type
- Calculations involving EUL_DATE_TRUNC function
- Admin mandatory conditions based on date columns

3.4.8 Conditions

Discoverer allows the creation of conditions in the metadata. Conditions can be both mandatory and optional. Mandatory conditions are not visible to the end user and have the effect of limiting the data that can be queried. Optional conditions can be defined for end user convenience, i.e. the user (who may not be familiar with SQL syntax) can optionally use pre-defined conditions, perhaps containing complicated logic in their reports by dragging them into their report from a list of available conditions.

3.4.8.1 Mandatory Conditions

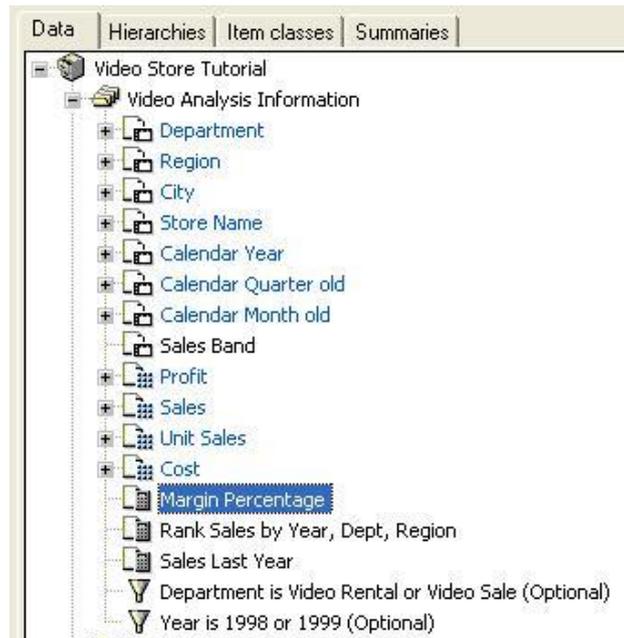
The migration assistant converts mandatory conditions depending on whether it is a simple, custom or complex folder. For simple and custom folders, mandatory conditions are migrated as “Content Filters” in the where clause section of the Logical Table Source. For complex folders, all users migrated from the Discoverer metadata will be of the BI EE group ‘everyone’ so mandatory conditions on complex folders will be applied to this user group as a security filter.

3.4.8.2 Optional Conditions

The optional conditions that can be defined in Discoverer metadata will not be migrated to the Oracle BI EE metadata in this initial release of the migration assistant. NOTE: It is expected that Optional conditions will be persisted as saved filters in the Web Catalog when Workbook Migration is available.

3.4.9 Aggregate Calculated Items

Discoverer aggregate calculations (also known as ‘admin calcs’) are calculated items that contain an aggregate function (e.g. SUM). They are represented by a different icon to other items in a folder (see below):



The formula for the ‘margin percentage’ item highlighted above is:

$$\frac{\text{SUM (Video Analysis Information.Profit)} * 100}{\text{SUM (Video Analysis Information.Sales)}}$$

These formulas will be migrated using a variation of the EVALUATE function available in BI EE.

3.4.10 Item hierarchies

Discoverer item hierarchies provide end users with a drill path through related data. A simple example is drilling through a geography hierarchy using data aggregated at a country level through region and city. These item hierarchies are migrated to Oracle BIEE dimensions. The levels in the Discoverer item hierarchy are migrated to the associated dimension levels.

Discoverer has the ability to allow the creation of multiple hierarchy drill paths. This migrates to entries in the preferred drill path property of the BI EE dimension level property.

All hierarchies based on a folder will be migrated to a single Dimension created on the folder. Item Hierarchies based on Complex Folders will not be migrated since a Dimension in Oracle BIEE must be associated with a Dimension table. Hierarchies spanning tables will be migrated by setting the Preferred Drill Path appropriately

Discoverer Property	Target Property In Presentation Layer	Target Property In Logical Layer	Target Property In Physical Layer
Name	n/a	Dimension name	n/a
Auto generate name	n/a	n/a	n/a
Description	n/a	Dimension description	n/a

3.4.11 Discoverer Date Hierarchies

There is no equivalent in Oracle BI EE to Discoverer date hierarchy templates so these will not be migrated. However, the resulting date hierarchies will be migrated.

3.4.12 Item Classes

There is no equivalent metadata object in Oracle BI EE so item classes are not migrated. When creating filters in Oracle BI Answers lists of values (LOVs) are generated at runtime.

3.4.13 Summary Folders

Discoverer summary folders are not migrated to Oracle BI EE.

3.5 How the Migration Assistant Converts Discoverer Metadata

3.5.1 Measures

Measures are identified by their placement and default aggregation settings in the Discoverer metadata. Any attribute having placement value as data point and default aggregation value that is neither 'none' nor 'detail' is defined as a measure. Any non-measure is considered a dimension

3.5.2 Facts and Dimensions

A folder is a Fact folder when either it doesn't join to any folder or is joined to some folder(s) and is the Detail table in the relationship.

A folder is a Dimension folder when it is involved in a relationship with some folder(s) where it is the Master table.

The leaf node for any dimension is always the level containing the folder key columns.

The content of a logical table source for a folder is mapped to its linked dimension levels.

3.5.3 Key Columns

Keys are created for tables in the Physical layer based on the columns involved in the join. A primary key is based on the columns involved in the join in which the table acts as a Master. A foreign key is based on the columns involved in the join in which the table acts as a Detail.

Logical folder key is based on: -

- Dimension folder – columns referenced by another folder
- Degenerated Dimension – foreign key columns
- Standalone folder – all non-measure columns if they exist else all the columns of the folder

3.5.4 Migrating Folders without Joins

The Oracle BIEE metadata requires that folders must be joined to another folder in the logical layer. In cases where the Discoverer metadata contains a standalone folder, a fact alias folder is created with just the measure columns from the standalone folder. If no measure columns exist then a COUNT column based on one of its key columns is created in the alias folder.

- A COUNT and COUNT_DISTINCT column is created in the logical folder for those columns that form part of the primary key and foreign key respectively.

3.5.5 Creation of Alias Objects

A dimension alias is created for a fact folder when there exists a hierarchy based on it.

A fact alias is created for a dimension folder if it has measure attribute(s).

An aliased folder is propagated to the presentation layer only in case of a folder having more than one join to the same folder [multiple join scenario].

The description of an aliased folder contains the usage information and the join for which it was created. The original folder must be used for all those joins which don't have an aliased folder based on it.

3.5.6 Aggregates and Calculations

Aggregated columns are created for measure attributes based on the default aggregation setting in the EUL.

Calculations involving columns from only one folder is created in its own folder whereas those calculations that involve columns from multiple folders are created in the complex folder.

All calculations are created using either the Evaluate or Evaluate_Aggr function in the logical table source containing the column mappings.

3.5.7 User Privileges

The migration of roles and privileges is limited to user creation only. All users found in the EUL export file will be migrated to the Oracle BIEE Group "Everyone" with a password that is username in UPPERCASE.

Discoverer 'Admin Mandatory' conditions based on simple folders are migrated as "where clause" for the logical table source corresponding to the simple folder.

Admin Mandatory conditions based on complex folders are migrated as security filters set on the Group "Everyone". This is set on the presentation folder corresponding to the complex folder.

3.6 Evaluating the Output from the Assistant

The goal of the conversion assistant is to accelerate the migration of Discoverer metadata to Oracle BIEE. This section describes scenarios where the results of the conversion assistant will need to be reviewed and validated as described below.

3.6.1 Use of the EUL_DATE_TRUNC function

The date hierarchies that Discoverer can automatically generate make use of a built in function called EUL_DATE_TRUNC. For example, the following calculation in Discoverer returns a 'Quarter' for a date column e.g. 'Transaction Date'. The valid values are 'Q1', 'Q2', 'Q3', 'Q4' and the result is a 'date' data type:

```
EUL_DATE_TRUNC(Transaction Date, '"Q"Q')
```

The result of passing this calculation through the conversion assistant is:

```
Evaluate( DECODE( %1, NULL,  
TO_DATE('01011900', 'MMDDYYYY'), TRUNC( %1,  
'Q')), TIMES.TRANSACTION_DATE)
```

Again, the resultant column is a 'date' data type. However, if this column is used in another calculation:

```
DECODE(Times."Transaction Date: Quarter", 'Q1', 'First  
Quarter', 'Q2', 'Second Quarter', 'Q3', 'Third  
Quarter', 'Q4', 'Fourth Quarter', 'Unknown')
```

The result of passing this calculation through the conversion assistant is:

```
EVALUATE('DECODE(DECODE( %1, NULL,  
TO_DATE(''01011900'', 'MMDDYYYY'), TRUNC( %1,  
'Q')), Q1, 'First Quarter', Q2, 'Second  
Quarter', Q3, 'Third Quarter', Q4, 'Fourth  
Quarter', 'Unknown')' AS VARCHAR ( 50 ),  
"Test"."". "VIDEO5". "TIMES". "TRANSACTION_DATE")
```

that can result in a data type mismatch.

3.6.2 Joins Involving Different Data Types

In the case of a join that involves columns of different data type, e.g.

```
Folder A.Column 1 [number] = Folder B.Column 1 [varchar]
```

The migration assistant will not apply any data type conversions but will migrate the join 'as is'. The conversions need to be added as a post migration assistant step.