Oracle® Business Intelligence Applications

Installation and Configuration Guide for Oracle Data Integrator Users

Version 7.9.5.2

E13669-01

February 2009

Oracle Business Intelligence Applications Installation and Configuration Guide for Oracle Data Integrator Users, Version 7.9.5.2

E13669-01

Contents

5 Configuring Common Areas and Dimensions

6 Configuring Oracle Procurement and Spend Analytics

7 Configuring Oracle Financial Analytics

8 Configuring Oracle Supply Chain and Order Management Analytics

Preface

Oracle Business Intelligence Applications are comprehensive prebuilt solutions that deliver pervasive intelligence across an organization, empowering users at all levels — from front line operational users to senior management — with the key information they need to maximize effectiveness. Intuitive and role-based, these solutions transform and integrate data from a range of enterprise sources, including Siebel, Oracle, PeopleSoft, SAP, and corporate data warehouses — into actionable insight that enables more effective actions, decisions, and processes.

Oracle Business Intelligence Applications are built on Oracle Business Intelligence Suite Enterprise Edition, a comprehensive next-generation BI and analytics platform.

Oracle Business Intelligence Applications include the following application families:

- Sales
- Service and Contact Center
- Marketing
- Financial
- Supply Chain and Supplier
- HR/Workforce

This guide explains how to install, set up, configure, and customize Oracle Business Intelligence Applications Version 7.9.5.2. For a high level road map of the steps required, see Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle Business Intelligence Applications With ODI".

Oracle recommends reading *Oracle Business Intelligence Applications Release Notes* before installing or using Oracle Business Intelligence Applications. The *Oracle Business Intelligence Applications Fusion Edition Release Notes* are available:

- On the Oracle Business Intelligence Applications CD-ROM.
- On the Oracle Technology Network at `http://www.oracle.com/technology/documentation/bi_apps.html` (to register for a free account on the Oracle Technology Network, go to `http://www.oracle.com/technology/about/index.html`).

Audience

This document is intended for BI managers and implementors of Oracle Business Intelligence Applications.

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.

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 Documents

For more information, see the following documents in the Oracle Business Intelligence Applications Release 7.9.5.2 documentation set (available at `http://www.oracle.com/technology/documentation/bi_apps.html`):

- *Oracle Business Intelligence Applications Release Notes for Oracle Data Integrator Users*
- *System Requirements and Supported Platforms for Oracle Business Intelligence Applications for Oracle Data Integrator Users*
- *Oracle Business Intelligence Applications Security Guide*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Part I

Getting Started

Part 1 helps you get started with Oracle Business Intelligence Applications, and contains the following chapters:

- Chapter 1, "What's New in This Release"
- Chapter 2, "Overview of Oracle Business Intelligence Applications"

Note: For a high level road map for installation, configuration, and customization steps for Oracle Business Intelligence Applications, see Chapter 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle Business Intelligence Applications With ODI".

1

What's New in This Release

This section summarizes the adapters that are supported in this release of Oracle Business Intelligence Applications.

1.1 What’s New In Oracle Business Intelligence Applications Version 7.9.5.2?

The main changes in Oracle Business Intelligence Applications Version 7.9.5.2 are:

- Oracle Data Integrator support.

  Oracle Business Intelligence Applications Version 7.9.5.2 uses Oracle BI Applications Configuration Manager in conjunction with Oracle Data Integrator (ODI) to perform Extract-Load Transform (E-LT).

  Oracle Business Intelligence Applications Version 7.9.5.2 does not support Informatica PowerCenter or Oracle Data Warehouse Console (DAC).

- Source System support.

  Oracle Business Intelligence Applications Version 7.9.5.2 supports Oracle EBS 11.5.10 data sources.

- Applications support.

  Oracle Business Intelligence Applications Version 7.9.5.2 supports the following application product families:

  – Oracle Financial Analytics

  – Oracle Human Resources Analytics

  – Oracle Procurement and Spend Analytics

  – Oracle Supply Chain and Order Management Analytics

2

Overview of Oracle Business Intelligence Applications

This chapter provides an overview of Oracle Business Intelligence Applications, and contains the following topics:

2.1 What is Oracle Business Intelligence Applications?

Oracle Business Intelligence Applications is a prebuilt business intelligence solution. Oracle Business Intelligence Applications Version 7.9.5.2 supports Oracle EBS 11.5.10 source systems, and uses Oracle Data Integrator (ODI) Version 10.1.3.5 as its E-LT environment.

If you already own an Oracle EBS 11.5.10 source system, you can purchase Oracle Business Intelligence and Oracle Business Intelligence Applications to work with the application.

Oracle Business Intelligence Applications deployed with Oracle Data Integrator (ODI) consists of the components shown in the table below.

Table 2–1 Oracle Business Intelligence Applications Components (with ODI)

Component	Description
Oracle Data Integrator	This is the Oracle data integration tool (E-LT tool) that performs the extract, load, transform operations for the data warehouse. Oracle Data Integrator works in conjunction with Oracle BI Applications Configuration Manager.
Oracle BI Applications Configuration Manager	This is a Web tool that enables you to set E-LT parameters, create and execute Execution Plans, and monitor ELT executions. Oracle BI Applications Configuration Manager works in conjunction with Oracle Data Integrator.

Table 2–1 (Cont.) Oracle Business Intelligence Applications Components (with ODI)

Component	Description
Prebuilt ODI content	This content includes Extract-Load-Transform (E-LT) repository objects (that is, scenarios, packages, and interfaces), which are contained in the ODI repository.
Prebuilt metadata content	This metadata content is contained in the Oracle Business Intelligence Applications repository file (OracleBIAnalyticsApps.rpd).
Prebuilt reports and dashboard content	This content is contained in the Oracle BI Presentation Services Catalog.
Oracle Business Analytics Warehouse	The prebuilt data warehouse that holds data extracted, loaded, and transformed from the transactional database (for more information, see Section 2.2, "Oracle Business Analytics Warehouse Overview").

2.2 Oracle Business Analytics Warehouse Overview

The Oracle Business Analytics Warehouse is a unified data repository for all customer-centric data, which supports the analytical requirements of the supported source systems.

The Oracle Business Analytics Warehouse includes the following:

- A complete relational enterprise data warehouse data model with numerous pre-built star schemas encompassing many conformed dimensions and several hundred fact tables.

  For more information about the data warehouse data model, please see the *Oracle Business Analytics Fusion Edition Data Model Reference*.

- An open architecture to allow organizations to use third-party analytical tools in conjunction with the Oracle Business Analytics Warehouse using the Oracle Business Intelligence Server.
- Prebuilt data extractors to incorporate data from external applications into the Oracle Business Analytics Warehouse.
- A set of E-LT (extract, load and transform) processes that takes data from an Oracle EBS 11.5.10 source system to create the Oracle Business Analytics Warehouse tables.
- A set of easy-to-use tools for the set up, configuration, administration, loading, and monitoring of the Oracle Business Analytics Warehouse. For example, ODI Designer, and Oracle BI Applications Configuration Manager.

Tip: Once you have installed Oracle Business Intelligence Applications, use the Repository Documentation option in Oracle BI Administration Tool to create a list of the repository objects in a text file (for more information, see Section 2.5, "Using Oracle BI-EE Repository Documentation".

2.3 Oracle Business Analytics Warehouse Architecture

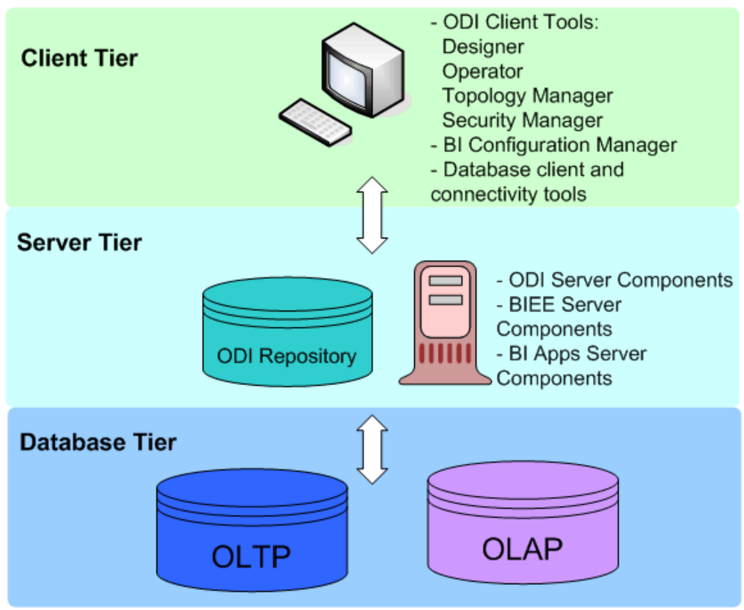
High-level analytical queries, like those commonly used in Oracle Business Intelligence, scan and analyze large volumes of data using complex formulas. This process can take a long time when querying a transactional database, which impacts overall system performance.

For this reason, the Oracle Business Analytics Warehouse was constructed using dimensional modeling techniques to allow for fast access to information required for decision making. The Oracle Business Analytics Warehouse derives its data from operational applications, and uses Oracle Data Integrator to extract, transform, and load data from various supported transactional database systems (OLTP) into the Oracle Business Analytics Warehouse.

2.3.1 Oracle Business Analytics Warehouse Architecture Components

The figure below illustrates the Oracle Business Analytics Warehouse architecture when deployed with ODI.

Figure 2–1 An example Oracle Business Analytics Warehouse architecture with ODI



The figure above shows the following Oracle Business Analytics Warehouse components:

- The Client Tier contains the ODI Client Tools, Oracle BI Applications Configuration Manager, and database client tools.
- The Server Tier contains the ODI Server and Repository, Oracle BI-EE server components, and Oracle Business Intelligence Applications server components.
- The Database Tier contains the OLTP (source) and OLAP (data warehouse) databases.

2.4 Roadmap To Installing, Configuring, and Customizing Oracle Business Intelligence Applications With ODI

To install, configure, and customize Oracle Business Intelligence Applications with ODI, do the following:

- Optimize your Oracle database, as described in Chapter 3, "Pre-installation and Predeployment Requirements for Oracle Business Intelligence Applications".

- Install and set up the Oracle Business Intelligence Applications components and Oracle Data Integrator components as described in Chapter 4, "Installing and Setting Up Oracle Business Intelligence Applications".

  Note: For an example that shows a typical deployment topology for Oracle Business Intelligence Applications, see Section 4.1, "About Oracle Business Intelligence Applications Topologies".

- (Optional) Perform any required configuration steps for the applications that you deploy, from the appropriate chapters below:
  - All applications - see Chapter 5, "Configuring Common Areas and Dimensions".
  - Procurement and Spend applications - see Chapter 6, "Configuring Oracle Procurement and Spend Analytics".
  - Financial applications - see Chapter 7, "Configuring Oracle Financial Analytics".
  - Supply Chain and Order Management applications - see Chapter 8, "Configuring Oracle Supply Chain and Order Management Analytics".
  - HR applications - see Chapter 9, "Configuring Oracle Human Resources Analytics".
- (Optional) If you want to modify the out-of-the-box Oracle Business Intelligence Repository (RPD file), see Chapter 10, "Configuring the Oracle Business Intelligence Applications Repository".
- (Optional) If you want to customize the out-of-the-box Oracle Business Intelligence Applications functionality, follow the steps described in Chapter 11, "Customizing the Oracle Business Analytics Warehouse".
- (Optional) If you want to modify the out-of-the-box Oracle Business Intelligence Applications security, follow the steps described in *Oracle Business Intelligence Applications Security Guide*.

Once you have installed and configured Oracle Business Intelligence Applications components, configured the modules (optional), and customized Oracle Business Intelligence Applications (optional), you are ready to start running ELT processes. For more information about performing ELT, see Section 4.7, "Loading Source Data Using an Execution Plan".

For a detailed example of how to run an E-LT process for Oracle Financials with an Oracle EBS OLTP data source, see Section 4.7.4, "Example of Running A Full Load E-LT in Oracle BI Applications Configuration Manager".

2.5 Using Oracle BI-EE Repository Documentation

When you deploy Oracle Business Intelligence Applications, you can use the following documentation and tools to manage your metadata:

- Oracle BI-EE Repository Documentation

  Using Oracle BI Administration Tool, you can generate repository documentation that lists the mapping from the presentation columns to the corresponding logical and physical columns. You might use this information for gap-analysis, or to create a record of your repository that you can use to compare with other repositories.

To generate Repository Documentation into a text or comma-separated file, log into Oracle BI Administration Tool and choose Tools, then Utilities, then Repository Documentation.

For more information about generating repository documentation, see *Oracle Business Intelligence Server Administration Guide*.

- Oracle BI-EE Presentation Services Catalog

  Using the Catalog Manager, you can view the names of the prebuilt dashboards and requests in the Presentation Services Catalog.

  To view the Presentation Catalog, select Catalog Manager from the Windows\Start\Programs\Oracle Business Intelligence menu.

Part II

Installation and Setup

Part II explains how to install and set up Oracle Business Intelligence Applications, and contains the following chapters:

- Chapter 3, "Pre-installation and Predeployment Requirements for Oracle Business Intelligence Applications"
- Chapter 4, "Installing and Setting Up Oracle Business Intelligence Applications"

Note: For a high level road map for installation, configuration, and customization steps for Oracle Business Intelligence Applications, see Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle Business Intelligence Applications With ODI".

3

Pre-installation and Predeployment Requirements for Oracle Business Intelligence Applications

This chapter provides information about preparing to install and deploy Oracle Business Intelligence Applications with Oracle Data Integrator, and contains the following topics:

- Section 3.1, "Oracle-Specific Database Guidelines for Oracle Business Analytics Warehouse"
- Section 3.2, "Database Client Connectivity Software Requirements"

You should review this information before you begin the installation and deployment process. For example, as a minimum you should read the appropriate database-specific guidelines for the source OLTP databases that you are using.

3.1 Oracle-Specific Database Guidelines for Oracle Business Analytics Warehouse

To configure your Oracle Business Analytics Warehouse on Oracle databases more easily, use the init10gR2.ora parameter template file that is installed with Oracle Business Intelligence Applications. The init10gR2.orafile is used to specify initialization parameters.

The init10gR2.ora is contained in the biapps_odi.zip file on the Oracle Business Intelligence Applications installation machine. When you unzip the biapps_odi.zip file onto your ODI machine, the init10gR2.ora file is located in the <ODI Home>\oracledi\biapps_odi\odifiles\dbfiles\ directory. For detailed instructions on unzipping and copying the init10gR2.ora file, see Section 4.4.2.2, "How to set up the Oracle BI Applications files for ODI").

The init10gR2.ora parameter template file provides guidelines based on the cost-based optimizer for Oracle 10g. Use these guidelines as a starting point. You will need to make changes based on your specific database sizes, data shape, server size (CPU and memory), and type of storage. The database administrator should make changes to the settings based on performance monitoring and tuning.

Rename the init10gR2.ora file, as follows:

- On Windows, rename the file as:

  %ORACLE_HOME%\database\init<SID>.ora

- On UNIX or Linux, rename the file as:

$ORACLE_HOME/dbs/init<SID>.ora

Note: SID is the Oracle System Identifier for the database machine.

Copy the renamed init10gR2.ora file into your $ORACLE_HOME/dbs directory, review the recommendations in the template file, and make the changes based on your specific database sizes, data shape, server size (CPU and memory), and type of storage. The database administrator should make changes to the settings based on performance monitoring and tuning considerations.

The table below describes some of the main initialization parameters for optimization that you can set in the init10gR2.ora file. For a complete list of initialization parameters, refer to *Oracle Database Reference 10g Release 2 (10.2)*.

Table 3–1 Parameters in database configuration file init<version>.ora

Parameter Name	Parameter Description
background_dump_dest	Specifies the path name (directory or disc) where debugging trace files for the background processes (for example, LGWR, DBWn) are written during Oracle operations.
control_files	Every database has a control file, which contains entries that describe the structure of the database (for example, its name, the timestamp of its creation, and the names and locations of its data files and redo files). CONTROL_FILES specifies one or more names of control files, separated by commas.
core_dump_dest	Specifies the directory where Oracle dumps core files. CORE_DUMP_DEST is primarily a UNIX parameter and might not be supported on your platform.
db_name	Specifies a database identifier of up to eight characters.
plsql_native_library_dir	Specifies the name of the directory where the shared objects produced by the native compiler are stored. This parameter used by the PL/SQL compiler.
user_dump_dest	Specifies the path name for a directory where the server writes debugging trace files on behalf of a user process.
utl_file_dir	Specify one or more directories that Oracle should use for PL/SQL file input and output (I/O). If you are specifying multiple directories, you must repeat the UTL_FILE_DIR parameter for each directory on separate lines of the initialization parameter

3.2 Database Client Connectivity Software Requirements

Make sure that you have suitable database client and connectivity software installed on the machine that will host the Oracle Business Analytics Warehouse (that is, the data warehouse).

4 Installing and Setting Up Oracle Business Intelligence Applications

This chapter explains how to install and set up the Oracle Business Intelligence Applications components to create a working Extract-Load Transform (E-LT) environment. It contains the following main topics:

- Section 4.1, "About Oracle Business Intelligence Applications Topologies"
- Section 4.2, "Installation and Set up Process Task List"
- Section 4.3, "Mandatory Requirements and Pre-installation Tasks"
- Section 4.4, "Installing Oracle Business Intelligence Applications and Oracle Data Integrator Software"
- Section 4.5, "Setting Up the Oracle Business Intelligence Applications and ODI Components"
- Section 4.6, "Miscellaneous and Supporting Tasks"
- Section 4.7, "Loading Source Data Using an Execution Plan"

For information about supported platform versions, see *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*.

To find out about other possible tasks required to deploy Oracle Business Intelligence Applications, see Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle Business Intelligence Applications With ODI".

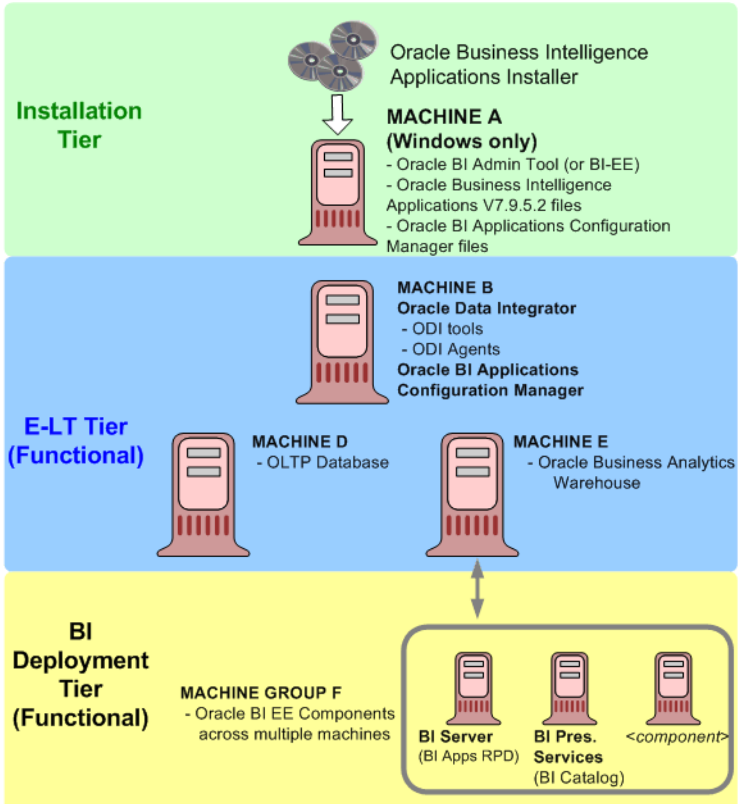
4.1 About Oracle Business Intelligence Applications Topologies

Oracle Business Intelligence Applications and Oracle Data Integrator (ODI) can be deployed flexibly across a wide range of topologies on different platforms and combinations of platform.

This section uses an example to explain a typical topology for an Oracle Business Intelligence Applications deployment. The figure below describes a typical deployment topology, which includes an installation tier, an E-LT tier, and an Oracle BI deployment tier.

Note: The Oracle Business Intelligence Applications installer only installs onto a Windows machine. To install Oracle Business Intelligence Applications components on Unix platforms, you must first install the components on a Windows machine, then manually copy over the components to a Unix machine.

Figure 4–1 A typical topology for an Oracle Business Intelligence Applications deployment with ODI



In the figure above, note the following:

- **Installation Tier**
  - MACHINE A (Windows-only)

    MACHINE A is a machine that has installed Oracle Business Intelligence Administration Tool, on which you run the Oracle Business Intelligence Applications installer to install the Oracle Business Intelligence Applications files. You can also install Oracle Business Intelligence Applications on a machine that has Oracle Business Intelligence Enterprise Edition installed.

    When the Oracle Business Intelligence Applications installation is complete, you manually copy the following files from the installation machine (MACHINE A) to the Business Intelligence Deployment Tier (MACHINE GROUP E), as follows:

    - You manually copy the OracleBI\Server\Repository\OracleBIAnalyticsApps.rpd file from MACHINE A to the machine that runs the BI Server in MACHINE GROUP E.

    - You manually copy the OracleBIData\Web\Catalog\EnterpriseBusinessAnalytics*.* files from MACHINE A to the machine that runs the BI Presentation Services Catalog in MACHINE GROUP E.

    You typically develop the RPD and Presentation Catalog and perform customization changes to fit your business requirements.

- You manually copy the biapps_odi.zip file from MACHINE A to the Oracle Data Integrator machine (that is, MACHINE B) and unzip the file into the \oracledi\ directory.

- You manually copy the Oracle BI Applications Configuration Manager files in the \OracleBI\dwrep\biapps_configmgr\ directory from MACHINE A to the Oracle Data Integrator machine (that is, MACHINE B).

- **E-LT Tier (Functional)**
  - MACHINE B (Windows, Unix, Linux)

    MACHINE B is a machine on which ODI is installed, and which runs the ODI production environment (for example, ODI tools, ODI Agents). You manually copy the biapps_odi.zip to this machine, and unzip the file into the \oracledi\ directory. In addition, you manually copy the Oracle BI Applications Configuration Manager files in the \OracleBI\dwrep\biapps_configmgr\ directory from MACHINE A to this machine.
  - MACHINE C (Windows, Unix, Linux)

    MACHINE C is a machine that hosts the transactional (OLTP) database.
  - MACHINE D (Windows, Unix, Linux)

    MACHINE D is a machine that hosts the Oracle Business Analytics Warehouse database.
- **BI Deployment Tier (Functional)**

  The BI Deployment Tier (Functional) tier is used to deploy the business intelligence dashboards, can have either of the following:
  - MACHINE GROUP E (Windows, Unix, Linux)

    MACHINE GROUP E is a group of machines that runs the Oracle Business Intelligence Enterprise Edition components. For example, one machine might run the BI Server and BI Applications RPD, and another machine might run the BI Presentation Services and the Oracle Business Analytics Warehouse.

Notes

- In an ODI environment, you typically maintain different environments (known as Contexts) for QA, Development, and Production. For information about Contexts, and about moving from a test to a production environment in ODI, see Section 4.6.10, "About deploying ODI across multiple environments".
- To maintain a back-up of the ODI project for recovery purposes, use the Import and Export option in ODI Designer. For example, you can select File, then Export to export the Work Repository to a directory or ZIP file. You can select File, then Import to import the Work Repository from a directory or ZIP file. For information about how to import a Work Repository, see Section 4.5.5.5, "How to import the Oracle BI Applications Work Repository".

4.2 Installation and Set up Process Task List

The Oracle Business Intelligence Applications installation and set up process consists of the following tasks. Complete each of these tasks in the order listed below.

1. Before you install Oracle Business Intelligence Applications and Oracle Data Integrator, make sure that you have satisfied the following requirements:

a. Make sure that you satisfy the Oracle Business Intelligence Infrastructure requirements that are specified in Section 4.3.1, "Oracle Business Intelligence Infrastructure Requirements".

b. Make sure that you satisfy the Oracle Data Integrator requirements that are specified in the *Oracle Data Integrator Installation Guide*. In addition, before you install ODI, Oracle recommends that you read the ODI documentation in detail to enable you to plan your ODI topology. ODI documentation is available on the BI media pack.

c. Make sure that you satisfy the Code Page requirements that are specified in Section 4.3.2, "Code Page and Data Movement Requirements").

d. Make sure that you perform that mandatory pre-installation tasks that are specified in Section 4.3.3, "Pre-installation Tasks").

2. Install the Oracle Business Intelligence Applications Version 7.9.5.2 software (for more information, see Section 4.4, "Installing Oracle Business Intelligence Applications and Oracle Data Integrator Software"), as follows:

   a. Run the Oracle installer to install the Oracle Business Intelligence Applications software. For more information, see Section 4.4.1, "How to Run the Oracle Business Intelligence Applications Installer (Windows)".

   b. Run the Oracle Data Integrator installer to install the Oracle Data Integrator software. For more information, see Section 4.4.2, "Installing Oracle Data Integrator Version 10.1.3.5".

3. (Optional) Copy the BI server components to appropriate Unix or Windows machines Section 4.5.1, "(Optional) How to Copy Over Installation Files".

4. Set up the Oracle Business Intelligence Applications and Oracle Data Integrator components (for more information, see Section 4.5, "Setting Up the Oracle Business Intelligence Applications and ODI Components"), as follows:

   a. Edit the ODI parameter file to set your deployment-specific settings (for more information, see Section 4.5.2, "How to configure the ODI parameter file").

   b. Create and start the ODI Agents (for more information, see Section 4.5.3, "How to start the ODI Agents").

   c. Set up the ODI Repository (for more information, see Section 4.5.4, "How to configure the ODI Repository").

   d. Set up the ODI Topology (for more information, see Section 4.5.5, "How to set up the ODI Topology").

   e. Create the ODI Scenarios (for more information, see Section 4.5.6, "How to generate the required ODI Scenarios").

   f. Set up the Data Warehouse Target Schema Objects (for more information, see Section 4.5.7, "How to install the Schema Objects for the Oracle Business Analytics Warehouse").

   g. Setup Oracle BI Applications Configuration Manager. For more information, see Section 4.5.8, "Setting Up Oracle BI Applications Configuration Manager".

   h. Set up the Master Packages in ODI to run Execution Plans (for more information, see Section 4.7.1, "How to set up Master Packages to run an Execution Plan").

5. Review and perform any post-installation tasks that are applicable to your deployment. For more information, see Section 4.6, "Miscellaneous and Supporting Tasks".

6. Perform a test full load E-LT (for more information, see Section 4.7, "Loading Source Data Using an Execution Plan").

   Note: Before you load your OLTP data, you typically configure your applications (for more information, see Section III, "Configuring Your Analytical Applications") and make customizations if required (for more information, see Section IV, "Customizing Oracle Business Intelligence Applications").

   For an example of performing a full load of OLTP data, see Section 4.7.4, "Example of Running A Full Load E-LT in Oracle BI Applications Configuration Manager".

4.3 Mandatory Requirements and Pre-installation Tasks

This section includes mandatory requirements that you must satisfy and pre-installation tasks that you must perform before you can deploy Oracle Business Intelligence Applications, and contains the following topics:

- Section 4.3.1, "Oracle Business Intelligence Infrastructure Requirements"
- Section 4.3.2, "Code Page and Data Movement Requirements"
- Section 4.3.3, "Pre-installation Tasks"

4.3.1 Oracle Business Intelligence Infrastructure Requirements

Oracle Business Intelligence Applications has the following infrastructure requirements:

- To install Oracle Business Intelligence Applications files, at a minimum you must have installed on the installation machine Oracle Business Intelligence Administration Tool.

  Alternatively, you can run the Oracle Business Intelligence Applications installer on a machine that has installed Oracle Business Intelligence Enterprise Edition. If so, Oracle Business Intelligence Enterprise Edition must have been installed using the Complete setup type option. This infrastructure does not need to be the functional version of Oracle Business Intelligence Enterprise Edition that you will use to deploy reports and dashboards in your live system. This infrastructure is only required to enable the Oracle Business Intelligence Applications installer to install the Oracle Business Intelligence Applications files onto a machine. For more information about topologies, see Section 4.1, "About Oracle Business Intelligence Applications Topologies".

- To run Oracle Business Intelligence Applications, you must have installed Oracle Business Intelligence Enterprise Edition as the infrastructure. In an Oracle Business Intelligence Applications deployment, Oracle Business Intelligence Enterprise Edition uses the Oracle Business Analytics Warehouse as a data source, uses the prebuilt OracleBIAnalyticsApps.rpd as the repository for the Oracle BI Server, and provides users with the Oracle BI Applications reports and dashboards.

  To determine the minimum version of Oracle Business Intelligence infrastructure that is supported for this release of Oracle Business Intelligence Applications, refer to *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*.

4.3.2 Code Page and Data Movement Requirements

The Oracle Business Analytics Warehouse can be deployed in various code page environments and supports global deployments. Oracle EBS data sources typically use UTF8 encoding.

Data movement in the following source database and data warehouse configuration modes are supported:

- Unicode to Unicode
- Code page (multi- or single-byte) to Unicode
- Code page to Code page (where the code pages are the same)

Oracle Business Intelligence Applications uses ODI to perform E-LT routines to move data from source database(s) to the Oracle Business Analytics Warehouse.

During the installation and Set up procedures described in this chapter, you will make various settings to enable accurate data movement. Use the guidelines and references noted below to determine values for these settings that are appropriate for your environment:

- Consult your database administrator to determine the code page your source OLTP database uses. Based on the type of data that will be moved from one or more source databases to the Oracle Business Analytics Warehouse, determine what code page you will need to use for the Oracle Business Analytics Warehouse database. Consider future requirements for storing data when determining what code page to use for the Oracle Business Analytics Warehouse.
- If your environment uses Oracle or DB2 database, you need to set environment variables NLS_LANG or DB2CODEPAGE. For information on how to set these environment variables see Section 4.3.3.3.1, "Setting the NLS_LANG Environment Variable for Oracle Databases".

4.3.3 Pre-installation Tasks

This section explains the mandatory tasks that you must perform for an Oracle Business Intelligence Applications deployment, and contains the following topics:

4.3.3.1 Creating the Required Databases and Tablespaces

This section provides information on creating database accounts that are required by Oracle Business Intelligence Applications components. You must create the database accounts listed in this section in a single Oracle 10gR2 database instance.

For example SQL commands that you can use to create database accounts, see "Example SQL Commands For Creating Database Users and Tablespaces".

Note: Make sure that you set up the database correctly, using the init<*Version*>.ora initialization file provided (for more information, see Section 3.1, "Oracle-Specific Database Guidelines for Oracle Business Analytics Warehouse").

Note: For information about changing the default database passwords or changing the schema names, see Section 4.6.4, "About Resetting the Default Passwords, Schema Names, and ODI Connection Details".

Before you install Oracle Business Intelligence Applications, use your target database tool to create the following database users and tablespaces with the specified privileges:

Table 4–1 Required Database Users and Tablespaces

Database User ID	Description	Default Password	Tablespace	Privileges
ODI_REP_7952	ODI Master and Work Repositories database schema.	ODI_REP_7952	BIAPPS_REP	CONNECT, RESOURCE
TEMP_BIAPPS	ODI staging database schema. TEMP_BIAPPS is used by ODI to create and drop segments during ELT processes. This schema must have a dedicated and locally managed tablespace with a uniform extent size.	TEMP_BIAPPS	BIAPPS_TEMP	CONNECT RESOURCE

Table 4–1 (Cont.) Required Database Users and Tablespaces

Database User ID	Description	Default Password	Tablespace	Privileges
DATA_BIAPPS	Data Warehouse database schema.	DATA_ BIAPPS	BIAPPS_ DATA	CONNECT RESOURCE CREATE DATABASE LINK CREATE ANY DIRECTORY CREATE ANY INDEX CREATE ANY PROCEDURE CREATE ANY SEQUENCE CREATE ANY SYNONYM CREATE ANY TABLE CREATE ANY TRIGGER CREATE ANY VIEW DELETE ANY TABLE DROP ANY DIRECTORY DROP ANY INDEX DROP ANY PROCEDURE DROP ANY SEQUENCE DROP ANY SYNONYM DROP ANY TABLE DROP ANY TRIGGER DROP ANY VIEW INSERT ANY TABLE SELECT ANY SEQUENCE SELECT ANY TABLE UPDATE ANY TABLE ANALYZE ANY

Table 4–1 (Cont.) Required Database Users and Tablespaces

Database User ID	Description	Default Password	Tablespace	Privileges
Index Tablespace	Index Tablespace. This is a mandatory Tablespace that is used as the INDEX_ TABLESPACE FlexField value in ODI (for more information, see Section 4.5.7.5, "Verifying the INDEX_ TABLESPACE Setting").	Not applicable.	BIAPPS_ INDEX	Not applicable.
DATA_ BIAPPSTX	Oracle BI Applications Configuration Manager database schema. Additional privileges are created as part of the setup (for more information, see Section 4.5.8.3, "How to Set Up Oracle BI Applications Configuration Manager on Windows").	DATA_ BIAPPSTX	BIAPPS_ DATA	CONNECT RESOURCE

Example SQL Commands For Creating Database Users and Tablespaces

- To create a database user named ODI_REP_7952:

  ```
  GRANT CONNECT, RESOURCE TO ODI_REP_7952 IDENTIFIED BY ODI_REP_7952;
  ```

  To make sure that the database object is created in the correct tablespace, use the following command:

  ```
  ALTER USER ODI_REP_7952 DEFAULT TABLESPACE BIAPPS_REP;
  ```

- To create a database user named TEMP_BIAPPS:

  ```
  GRANT CONNECT, RESOURCE TO TEMP_BIAPPS IDENTIFIED BY TEMP_BIAPPS;
  ```

- To create a database user named DATA_BIAPPS:

  ```
  GRANT CONNECT,RESOURCE,CREATE DATABASE LINK,CREATE ANY DIRECTORY,CREATE ANY
  INDEX,CREATE ANY PROCEDURE,CREATE ANY SEQUENCE,CREATE ANY SYNONYM,CREATE ANY
  TABLE,CREATE ANY TRIGGER,CREATE ANY VIEW,DELETE ANY TABLE,DROP ANY DIRECTORY,
  DROP ANY INDEX,DROP ANY PROCEDURE,DROP ANY SEQUENCE,DROP ANY SYNONYM,DROP ANY
  TABLE,DROP ANY TRIGGER,DROP ANY VIEW,INSERT ANY TABLE,SELECT ANY SEQUENCE,
  SELECT ANY TABLE,UPDATE ANY TABLE,ANALYZE ANY TO DATA_BIAPPS IDENTIFIED BY
  DATA_BIAPPS;
  ```

- To create a tablespace named BIAPPS_INDEX:

  ```
  CREATE TABLESPACE BIAPPS_INDEX DATAFILE 'tmp_tablespace.dat' SIZE 10M REUSE;
  ```

- To create a database user named DATA_BIAPPSTX:

```
GRANT CONNECT, RESOURCE TO DATA_BIAPPSTX IDENTIFIED BY DATA_BIAPPSTX;
```

4.3.3.2 How to create a TNSNAMES Entry for your Oracle EBS 11.5.10 data source

Oracle EBS 11.5.10 source data is accessed through a Database Link (DBLINK) drawn from the ODI_REP_7952 schema on the Oracle Business Analytics Warehouse database instance to the schema on the EBS 11.5.10 database instance. You need to enable this connection by creating an entry in the tnsnames.ora file in your Oracle Business Analytics Warehouse database instance.

The file tnsnames.ora is located in <ORACLE_HOME>\NETWORK\ADMIN directory on the machine hosting your Oracle Business Analytics Warehouse database instance.

Example TNS entries:

```
ipc-ora=(DESCRIPTION=
             (ADDRESS=
                 (PROTOCOL=IPC)
                 (KEY=ORCL)
             )
             (CONNECT_DATA=(SID=ORA102))
            (HS=)
             )
ipc-gw=(DESCRIPTION=
           (ADDRESS=
               (PROTOCOL=IPC)
               (KEY=ORCL)
           )
           (CONNECT_DATA=(SID=drdahoa1))
                                (HS=)
           )
```

You can update the TNS file manually using a text editor, or use the configuration tool, Oracle Net Configuration Assistant. For more information about the configuration tool, refer to *Oracle Net Services Administrator's Guide* in the Oracle database documentation library.

4.3.3.3 Configuring the Language Environment

On the machines that will host the databases, you need to configure the language environment using the NLS_LANG variable.

4.3.3.3.1 Setting the NLS_LANG Environment Variable for Oracle Databases Follow this procedure to set the NLS_LANG environment variable for Oracle databases.

Note: You need to set the NLS_LANG environment variable on each machine that has the Oracle client installed.

To set the NLS_LANG environment variable for Oracle databases:

1. Determine the NLS_LANG value.

   a. In the data warehouse database, run the command

      ```
      SELECT * FROM V$NLS_PARAMETERS
      ```

   b. Make a note of the NLS_LANG value, which is in the format [NLS_LANGUAGE]_[NLS_TERRITORY].[NLS_CHARACTERSET].

For example: `American_America.UTF8`

2. For Windows:

   a. Navigate to Control Panel > System and click the Advanced tab. Click Environment Variables.

   b. In System variables section, click New.

   c. In the Variable Name field, enter `NLS_LANG`.

   d. In the Variable Value field, enter the NLS_LANG value that was returned in Step 1.

   The format for the NLS_LANG value should be [NLS_LANGUAGE]_[NLS_TERRITORY].[NLS_CHARACTERSET].

   For example: `American_America.UTF8`.

   Note: The NLS_LANG character set should reflect the setting of the operating system character set of the client. For example, if the database character set is AL32UTF8 and the client is running on a Windows operating system, then you should not set AL32UTF8 as the client character set in the NLS_LANG parameter because there are no UTF-8 WIN32 clients. Instead, the NLS_LANG setting should reflect the code page of the client. For example, on an English Windows client, the code page is 1252. An appropriate setting for NLS_LANG is AMERICAN_AMERICA.WE8MSWIN1252.

   Setting NLS_LANG correctly allows proper conversion from the client operating system character set to the database character set. When these settings are the same, Oracle assumes that the data being sent or received is encoded in the same character set as the database character set, so character set validation or conversion may not be performed. This can lead to corrupt data if the client code page and the database character set are different and conversions are necessary

   ---

3. For UNIX and Linux, set the variable as shown below:

```
setenv NLS_LANG <NLS_LANG>
echo $NLS_LANG
```

   For example: `setenv NLS_LANG American_America.UTF8`. Use the echo $NLS_LANG command to verify that you have set the environment correctly.

   If your data is 7-bit or 8-bit ASCII and ODI is running on UNIX or Linux, then set `NLS_LANG <NLS_LANGUAGE>_<NLS_TERRITORY>.WE8ISO8859P1`

4.3.3.4 Opening Firewall Ports

Depending on your network setup, you might have to open ports on your firewall. For example, you might open a port in your firewall for the connection from ODI to the source system database or data warehouse database.

4.4 Installing Oracle Business Intelligence Applications and Oracle Data Integrator Software

This section explains how to install Oracle Business Intelligence Applications software and Oracle Data Integrator software, and contains the following topics:

- Section 4.4.1, "How to Run the Oracle Business Intelligence Applications Installer (Windows)"
- Section 4.4.2, "Installing Oracle Data Integrator Version 10.1.3.5"

Note: After installation, you must follow the set up instructions specified in Section 4.5, "Setting Up the Oracle Business Intelligence Applications and ODI Components".

The Oracle Business Intelligence Applications installer runs on Windows, and requires an Oracle Business Intelligence infrastructure to be installed. For more information on Oracle Business Intelligence infrastructure requirements, including versions supported for this release of Oracle BI Applications, see Section 4.3.1, "Oracle Business Intelligence Infrastructure Requirements".

4.4.1 How to Run the Oracle Business Intelligence Applications Installer (Windows)

This section explains how to install the Oracle Business Intelligence Applications files using the Oracle Business Intelligence Applications Installation Wizard.

Note: The Oracle Business Intelligence Applications installer only installs onto a Windows machine. To install Oracle Business Intelligence Applications components on Unix platforms, you must first install the components on a Windows machine, then manually copy over the components to one or more Unix machines.

When you run the Oracle Business Intelligence Applications Installation Wizard, the Oracle Business Intelligence Applications files are installed into the existing Oracle Business Intelligence infrastructure directory (for example, <DRIVE>:\OracleBI\).

Note: If you have a previous version of Oracle Business Intelligence Applications installed, you must uninstall this version before you run the installer for Oracle Business Intelligence Applications Version 7.9.5.2. Oracle recommends that before you uninstall the old version you make a back-up of the RPD and Presentation Services Catalog.

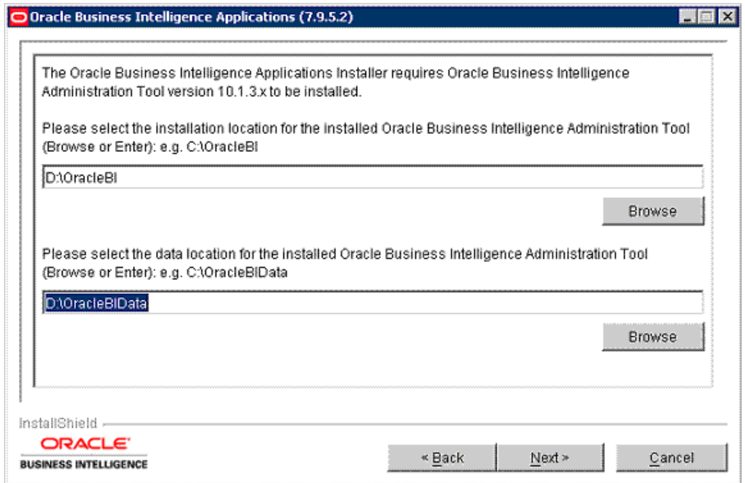
To install Oracle Business Intelligence Applications on Windows:

1. Access the installation files on the installation CD-ROM, and then run the program setup.exe to display the Welcome page.

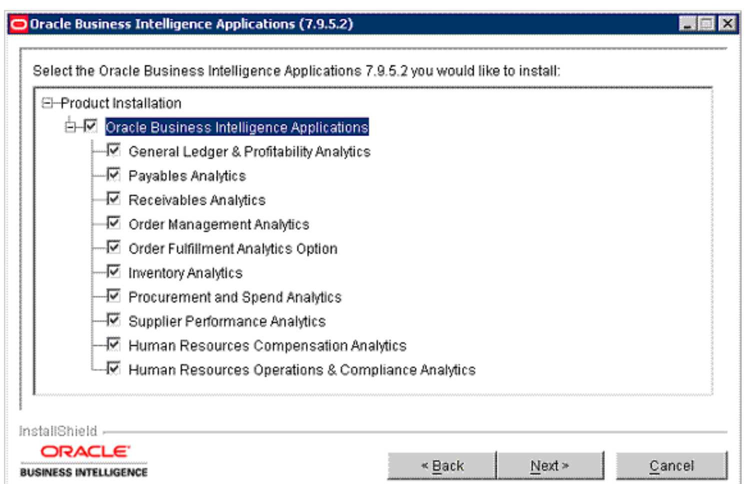
Note: To run the installer in console (or text) mode, run the command setup.exe -console. You do not see following dialogue screens in console installation mode. Instead, you enter input as plain text in the terminal window when prompted.

---

2. Click Next to display the Oracle Business Intelligence infrastructure selection page.



3. Enter or browse for the location for the Oracle Business Intelligence infrastructure (for example, <DRIVE>:\OracleBI\), the Oracle Business Intelligence infrastructure data (for example, <DRIVE>:\OracleBIData\).
4. Click Next to display the Select Oracle Business Intelligence Applications page.
5. At the Select Oracle Business Intelligence Applications page, select the applications that you want to install, then click Next.



To determine the applications you should install for the Oracle Business Intelligence Applications licenses that you have purchased, see the spreadsheet entitled '2-Installer Options' in *Oracle Business Intelligence Applications Licensing and Packaging Guide*. This guide is part of the Oracle Business Intelligence Media Pack.

6. If you selected more than one application on the Select Oracle Business Intelligence Applications page, an information page is displayed. Click Next.
7. At the summary page, review the summary information and click Next to start the installation.

   The installer installs the Oracle Business Intelligence Applications directories and files into the Oracle Business Intelligence infrastructure installation (for example, <DRIVE>:\OracleBI\), and installs the Oracle BI Repository file and Presentation Catalog.

   Note: Even when the progress bar on the installer reports 100% complete, you must wait until the **Finish** button is displayed.
8. Click Finish.

   > **Tip:** To check that you have installed the correct version of Oracle Business Intelligence Applications, look in the <DRIVE>:\OracleBI\Document\version_apps.txt file in the Oracle Business Intelligence infrastructure directory, which contains the installed version number.

9. When the installation is complete, unzip the \OracleBIData\web\catalog\EnterpriseBusinessAnalytics.zip file into the \OracleBIData\web\catalog\ directory, which creates a new directory called EnterpriseBusinessAnalytics.

   You typically copy the \OracleBIData\web\catalog\EnterpriseBusinessAnalytics directory from the installation machine to the machine on which the Oracle Presentation Services is running. For more information, see Section 4.1, "About Oracle Business Intelligence Applications Topologies".

When the installation is complete, verify that the following directories or files are installed in the \OracleBI\ directory of the Oracle Business Intelligence Applications installation machine:

- The \OracleBI\dwrep\biapps_odi\ directory.
- The Oracle Business Intelligence Applications repository file named OracleBIAnalyticsApps.rpd in the \OracleBI\server\Repository directory. You typically copy the OracleBIAnalyticsApps.rpd file to the machine on which Oracle BI Server is running (for more information, see Section 4.1, "About Oracle Business Intelligence Applications Topologies").

Notes

- To log into the OracleBIAnalyticsApps.rpd file using the Oracle Business Intelligence Administration Tool, use the following login details:
  – Username: Administrator
  – Password: SADMIN
- Refer to the Oracle Business Intelligence Enterprise Edition documentation for more information on working with the repository and Presentation Services Catalog.

4.4.2 Installing Oracle Data Integrator Version 10.1.3.5

This section explains how to install Oracle Data Integrator for an Oracle Business Intelligence Applications deployment. Before you install Oracle Data Integrator, make sure that you have satisfied the requirements specified in *Oracle Data Integrator Installation Guide*.

To install Oracle Data Integrator Version 10.1.3.5 for Oracle Business Intelligence Applications, do the following:

- Install the ODI software (for more information, see Section 4.4.2.1, "How to Run the Oracle Data Integrator Installer").
- Manually copy the Oracle BI Applications ZIP file biapps_odi_zip into the ODI installation directory and unzip the file (for more information, see Section 4.4.2.2, "How to set up the Oracle BI Applications files for ODI").

When you run the Oracle Data Integrator installer, you will install the following ODI components:

- Agent
- Designer
- Operator
- Repository Management
- Security Manager
- Topology Manager

You can add other ODI components later if required (for example, Oracle Data Profiling, Metadata Navigator) by running the ODI installer again on a machine. For more information, refer to the Oracle Data Integrator Documentation Library.

4.4.2.1 How to Run the Oracle Data Integrator Installer

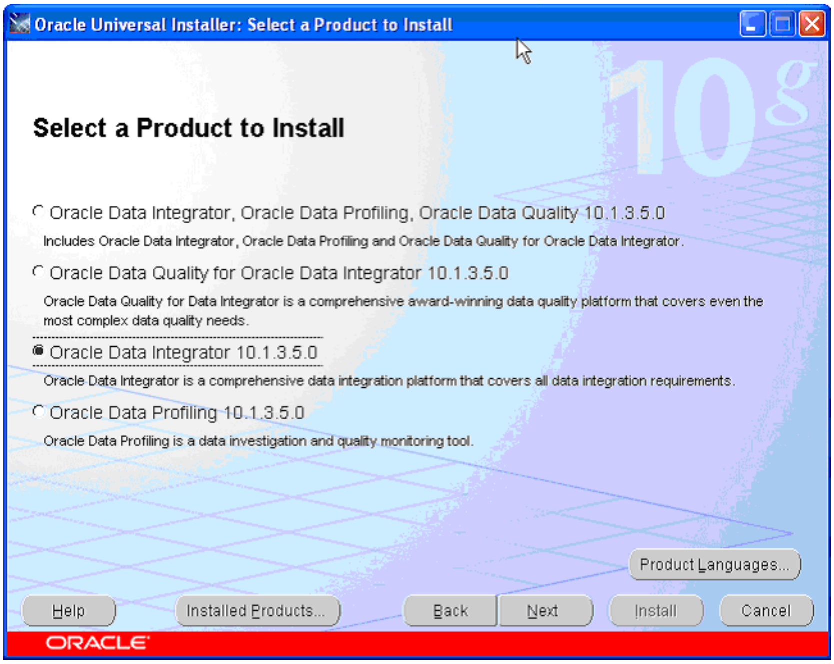
To start the Oracle Data Integrator installer:

1. Access the ODI installation files on the installation CD-ROM, then:
   - on Windows, double-click `setup.bat` to start the installer.
   - on UNIX, enter the `./runInstaller` command to start the installer.

   Note: To run the installer in console (or text) mode, run the command setup.exe -console. You do not see following dialogue screens in console installation mode. Instead, you enter input as plain text in the terminal window when prompted.

   ---

2. At the Select a Product to Install screen, select the **Oracle Data Integrator 10.1.3.5.0** radio button.



3. Click Next and follow the remaining on-screen instructions to start the ODI installation.

   The Oracle Data Integrator installer creates an Oracle Home directory that contains a \oracledi\ directory. For example, you might have an ODI installation directory called D:\OraHome_1\. The ODI installation directory is referred to as $ODI_HOME.

Note: On UNIX platforms, add the following environment variable for the user who has installed Oracle Data Integrator: `ODI_JAVA_HOME=$ODI_HOME/jre/1.4.2`. Refer to *Oracle Data Integrator Installation Guide* for a full list of environment variables required for other components.

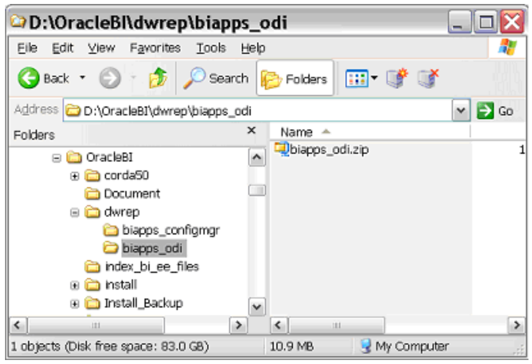
4.4.2.2 How to set up the Oracle BI Applications files for ODI

After you have installed Oracle Data Integrator Version 10.1.3.5, you need to manually add some Oracle Business Intelligence Applications files to the ODI installation machine, as follows:

To set up the required Oracle BI Applications files for ODI:

1. On the Oracle Business Intelligence Applications installation machine, locate the biapps_odi.zip file in the \OracleBI\dwrep\biapps_odi\ directory.

   For example, D:\OracleBI\dwrep\biapps_odi\biapps_odi.zip.



2. On the Oracle Data Integrator machine, locate the \oracledi\ directory in the Oracle Home directory in which ODI is installed.

   For example, D:\OraHome_1\oracledi\.

3. On the Oracle Data Integrator machine, copy the \OracleBI\dwrep\biapps_odi\biapps_odi.zip file from the Oracle Business Intelligence Applications installation machine to the \oracledi\ directory on the Oracle Data Integrator installation machine.

4. On the Oracle Data Integrator installation machine, unzip the $ODI_HOME\oracledi\biapps_odi.zip file into the $ODI_HOME\oracledi\ directory.

   When you unzip the biapps_odi.zip file, the following new directories are created in the $ODI_HOME\oracledi\ directory:

   - \biapps_odi\odifiles\binfiles\
   - \biapps_odi\odifiles\libfiles\
   - \biapps_odi\odifiles\importfiles\
   - \biapps_odi\odifiles\dbfiles\
   - \biapps_odi\odifiles\odidatafiles\

     Note: The \odidatafiles\ directory contains a 'lkpfiles' directory (that contains the domain value look-up files) and a 'srcfiles' directory (that contains metadata definitions for the source flat-file data stores).

5. On the Oracle Data Integrator machine, copy the files in the $ODI_HOME\oracledi\biapps_odi\odifiles\binfiles\ directory to the $ODI_HOME\oracledi\bin\ directory.

6. On the Oracle Data Integrator machine, copy the files in the $ODI_HOME\oracledi\biapps_odi\odifiles\libfiles\ directory to the $ODI_HOME\oracledi\lib\ directory.

7. On the Oracle Data Integrator machine, copy the init10gR2.ora file in the $ODI_HOME\oracledi\biapps_odi\dbfiles\ directory to the machine that hosts the data warehouse (for example, in the $ORACLE_HOME/dbs directory).

   Rename the init10gR2.ora file, as follows:

   - On Windows, rename the file as:

     %ORACLE_HOME%\database\init<SID>.ora

   - On UNIX or Linux, rename the file as:

     $ORACLE_HOME/dbs/init<SID>.ora

For more information about setting the parameters in the *.ora file, see Section 3.1, "Oracle-Specific Database Guidelines for Oracle Business Analytics Warehouse".

8. If required (for example, to conserve disk space), you can delete the biapps_odi.zip file from the Oracle Data Integrator machine.

4.5 Setting Up the Oracle Business Intelligence Applications and ODI Components

This sections explains how to set up the Oracle Business Intelligence Applications and Oracle Data Integrator components to create an operational E-LT environment. Perform the tasks in this section in the sequence specified. This section contains the following topics:

- Section 4.5.1, "(Optional) How to Copy Over Installation Files"
- Section 4.5.2, "How to configure the ODI parameter file"
- Section 4.5.3, "How to start the ODI Agents"
- Section 4.5.4, "How to configure the ODI Repository"
- Section 4.5.5, "How to set up the ODI Topology"
- Section 4.5.6, "How to generate the required ODI Scenarios"
- Section 4.5.7, "How to install the Schema Objects for the Oracle Business Analytics Warehouse"
- Section 4.5.8, "Setting Up Oracle BI Applications Configuration Manager"

Notes

- Before following the steps in this section, you must have installed the Oracle Business Intelligence Applications software and the Oracle Data Integrator software as specified in Section 4.4, "Installing Oracle Business Intelligence Applications and Oracle Data Integrator Software".
- The installation and set up steps in this chapter assume the following recommended ODI configuration:
  - The ODI Master and Work Repositories are installed in a single database schema.
  - The Data Warehouse database schema is hosted on the same database instance as the ODI Repository.
  - The ODI Work (temporary) database schema is hosted on the same database instance as ODI Repository.
  - There is only one Oracle Applications 11.5.10 data source for each environment (for example, the Production context, for more information about Contexts, refer to the ODI documentation).

4.5.1 (Optional) How to Copy Over Installation Files

A typical deployment environment for Oracle Business Intelligence Applications is to have client tools running on Windows, and have servers running on Unix. After installing Oracle Business Intelligence Applications files onto a Windows machine, you might copy the BI server components to appropriate Unix machines, as described in this section.

To copy over installation files:

1. Copy the OracleBI\Server\Repository\OracleBIAnalyticsApps.rpd file from the Oracle Business Intelligence Applications installation machine to the machine that runs the BI Server.

2. Copy the OracleBIData\Web\Catalog\EnterpriseBusinessAnalytics*.* files from the Oracle Business Intelligence Applications installation machine to the machine that runs the BI Presentation Services Catalog.

   The Presentation Services Catalog is installed as a ZIP file named EnterpriseBusinessAnalytics.zip. Make sure that you un-zip the EnterpriseBusinessAnalytics.zip file into the OracleBIData\Web\Catalog\ directory.

For more information about deploying Oracle Business Intelligence Applications on multiple machines, see Section 4.1, "About Oracle Business Intelligence Applications Topologies".

4.5.2 How to configure the ODI parameter file

You configure the ODI parameter file to specify configuration values that are specific to your Oracle Business Intelligence Applications deployment. For example, you need to specify the ODBC URL.

To configure the odiparams.bat file:

1. On the ODI machine, open the odiparams.bat or odiparams.sh file in a text editor.

   The odiparams.bat and odiparams.sh files are located in the $ODI_HOME\oracledi\bin\ directory. For example, D:\OraHome_1\oracledi\bin\.

2. Set the value of the following parameter:

   ```
   set ODI_SECU_URL=jdbc:oracle:thin:@<host>:<port>:<sid>
   ```

   Replace <host>, <port>, and <sid> with your database specific parameters.

   For example, 'jdbc:oracle:thin:@US12345.us.company.com:1521:US12345'.

3. Save the odiparams file.

4.5.3 How to start the ODI Agents

To deploy ODI, you must start the ODI Agents named 'WORKFLOW' and 'INTERFACE' as background processes. You must start the ODI Agents each time the machine on which you are running the ODI Agents is started. This task explains how to start the ODI Agents using the default port numbers 20910 (for the WORKFLOW Agent) and 20911 (for the INTERFACE Agent).

Note: If you encounter a port conflict when you run the ODI Agents, you might need to use different port numbers (for more information, see (Section 4.6.13, "How to Resolve Conflicts in ODI Agent Port Numbers").

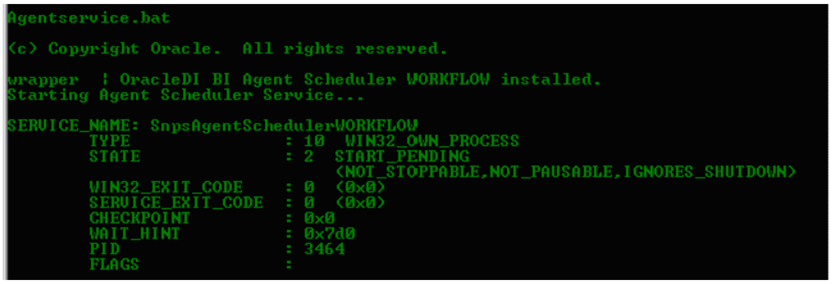
When you use ODI Designer to execute Packages and Interfaces, you specify the INTERFACE Agent.

To start the ODI Agents on Windows:

1. On the ODI machine, open a command window and change directory to the $ODI_HOME\oracledi\bin directory.

2. Execute the following commands one at a time:

   ```
   agentservice -i -s WORKFLOW 20910
   agentservice -i -a INTERFACE 20911
   ```



The above commands create the following agents:

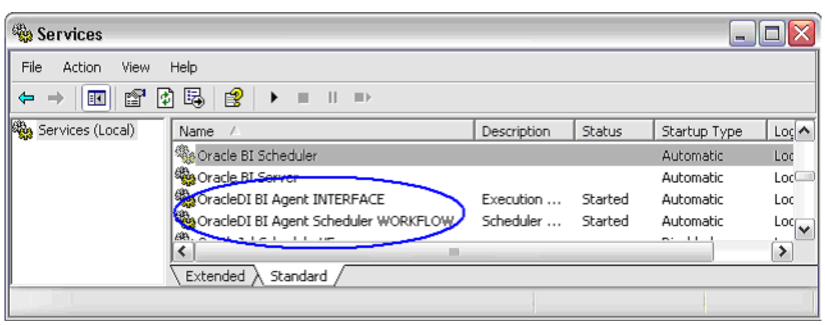
Table 4–2   ODI Agents for Oracle Business Intelligence Applications

Name of Agent	Windows Service Name	Description	Mode	Port
WORKFLOW	OracleDI BI Agent WORKFLOW	Scheduler agent for ODI sessions.	Scheduler	20910
INTERFACE	OracleDI BI Agent INTERFACE	Listener agent for ODI sessions. When you use ODI Designer to execute Packages and Interfaces, you always specify the INTERFACE Agent when prompted.	Listener	20911

3. Display the Windows Services dialog.

   For example, run 'services.msc' from the Windows > Run dialog.

4. Start the 'OracleDI BI Agent WORKFLOW' service and the 'OracleDI BI Agent INTERFACE' service.



Note: Command log information about Agents is stored in the file $ODI_HOME\oracledi\bin\agentservice.log. For a full list of log files, see Section 4.6.15, "List of Log Files".

To install and start the ODI Agents on Unix:

1. On the ODI machine, open a command window and change directory to the $ODI_HOME/oracledi/bin directory.

2. Execute the following commands one at a time:

```
./agentscheduler.sh -NAME=WORKFLOW -PORT=20910 &
```

```
./agent.sh -NAME=INTERFACE -PORT=20911 &
```

Notes

- For the agentscheduler and agent commands, you can set the trace level using -v <trace level>, and use '> trace.txt' to output to a log file. For example:

  ```
  agent.bat "-v=5" > trace.txt
  agent.sh -v=5 . trace.txt
  ```
- If you restart the machine on which you are running the ODI Agents, you must re-start the ODI Agents.

4.5.4 How to configure the ODI Repository

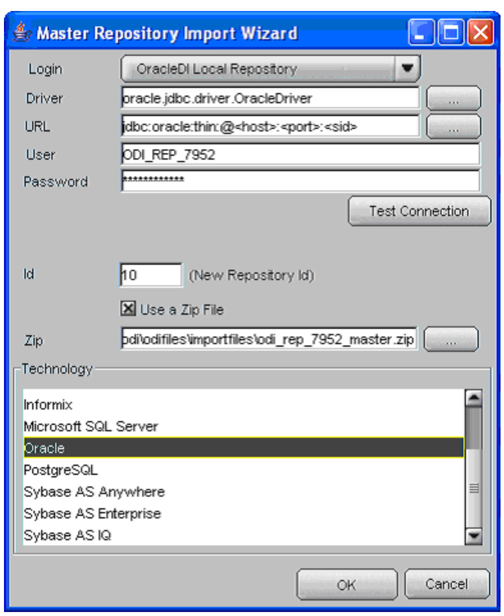
To configure the ODI Repository, do the following:

- Import the ODI Master Repository (for more information, see Section 4.5.4.1, "How to import the ODI Master Repository").
- Configure the Topology Manager connection to the ODI Master Repository (for more information, see Section 4.5.4.2, "How to configure the Topology Manager connection to the ODI Master Repository").
- Create the ODI Work Repository (for more information, see Section 4.5.4.3, "How to create the blank ODI Work Repository").

4.5.4.1 How to import the ODI Master Repository

To import the ODI Master Repository:

1. On the ODI machine, choose Start, then All Programs, then Oracle Data Integrator, then Repository Management, then Master Repository Import to display the Master Repository Import Wizard dialog.

   On a UNIX or Linux machine, you can display the Master Repository Import Wizard by running the `$ODI_HOME/oracledi/bin/mimport.sh` command.

2. In the Master Repository Import Wizard dialog, enter the appropriate information, as described in the table below.

Table 4–3 Master Repository Import Wizard dialog fields

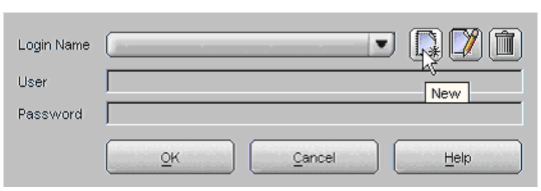
Field	Description
Login	Do not change the default option (OracleDI Local Repository).
Driver	Specify 'oracle.jdbc.driver.OracleDriver'.
URL	Specify the JDBC URL to the Oracle Business Analytics Warehouse in the format jdbc:oracle:thin:@<host>:<port>:<sid>. Replace <host>, <port> and <sid> with your database installation specific values. For example, 'jdbc:oracle:thin:@US12345.us.company.com:1521:US12345'.
User	Specify ODI_REP_7952. This is the ODI Repository database user.
Password	Specify ODI_REP_7952. This is the default password for the ODI Repository database user.
Id	Specify a non-zero integer (the recommended value is 10).
Use a zip file	Select this check box.
Zip	Specify $ODI_HOME\oracledi\biapps_ odi\odifiles\importfiles\odi_rep_7952_master.zip
Technology	Select Oracle from the list.

3. Click Test Connection to verify the details, then click OK to close the Information dialog.
4. Click OK to save the details and start the import.

4.5.4.2 How to configure the Topology Manager connection to the ODI Master Repository

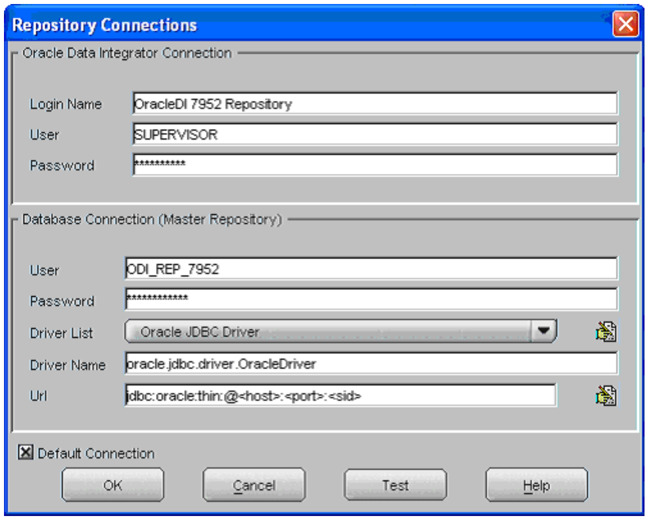
To configure the Topology Manager connection to the ODI Master Repository:

1. On the ODI machine, choose Start, then All Programs, then Oracle Data Integrator, then Topology Manager to display the Topology Manager login dialog.



   On a UNIX or Linux machine, you can display the Topology Manager by running the `/oracledi/bin/topology.sh` command.

2. Click New to display the Repository Connections dialog.



3. In the Repository Connections dialog, enter the appropriate information, as described in the table below.

Table 4–4 Repository Connections dialog fields

Field	Description
Login Name	Specify an identifier for the repository connection.
User	Specify SUPERVISOR. This is the ODI Administrator database user.
Password	Specify SUPERVISOR. This is the default password for the ODI Administrator database user.
User	Specify ODI_REP_7952. This is the ODI Repository database user name.
Password	Specify ODI_REP_7952. This is the default password for the ODI Repository database user name.
Driver List	Select 'Oracle JDBC Driver'.
Driver Name	Specify 'oracle.jdbc.driver.OracleDriver'.
URL	Specify the JDBC URL to the Oracle Business Analytics Warehouse in the format jdbc:oracle:thin:@<host>:<port>:<sid>. Replace <host>, <port> and <sid> with your database installation specific values. For example, 'jdbc:oracle:thin:@US12345.us.company.com:1521:US12345'.
Default Connection	Select this check box.

4. Click Test to verify the details.
5. Click OK on the Information dialog.
6. Click OK on the Security Repository Connections dialog to log in to Topology Manager.

4.5.4.3 How to create the blank ODI Work Repository

You create a blank ODI Work Repository to enable you to import the Oracle Business Intelligence Applications Work Repository in a later setup task (for more information, see Section 4.5.5.5, "How to import the Oracle BI Applications Work Repository").

To create the blank ODI Work Repository:

1. Log into Topology Manager as user SUPERVISOR as described in Section 4.5.4.2, "How to configure the Topology Manager connection to the ODI Master Repository".
2. In the Topology Manager, display the Repositories tab.
3. In the Repositories pane, right click on Work Repositories node.
4. Select the **Insert Work Repository** menu option to display the Data Server: New dialog.
5. Display the Definition tab, and enter the appropriate information, as described in the table below.

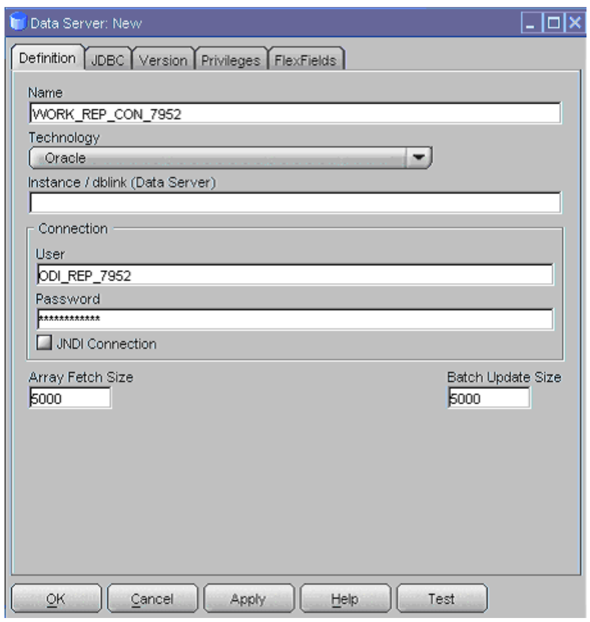


Table 4–5 Data Server: New dialog, Definition tab fields

Field	Description
Name	Specify WORK_REP_CON_7952. This is the identifier for the repository connection.
Technology	Select Oracle from the drop down list
Instance/dblink (Data Server)	Leave this field blank.
User	Specify 'ODI_REP_7952'. This is the repository database user name.
Password	Specify 'ODI_REP_7952'. This is default password for the repository database user name.

Table 4–5 (Cont.) Data Server: New dialog, Definition tab fields

Field	Description
JNDI Connection	Clear this check box.
Array Fetch Size	Specify a value suitable to your environment (a typical value is 5000).
Batch Update Size	Specify a value suitable to your environment (a typical value is 5000).

6. Display the JDBC tab, and enter the appropriate information, as described in the table below.

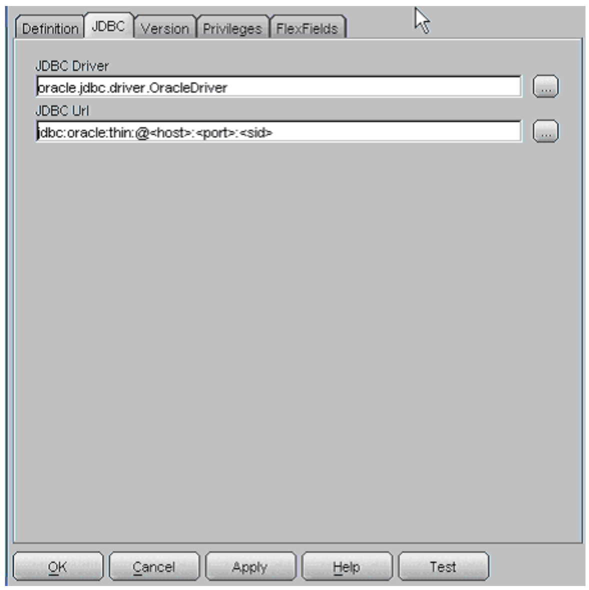


Table 4–6 Data Server: New dialog, JDBC tab fields

Field	Description
JDBC Driver	Specify oracle.jdbc.driver.OracleDriver.
JDBC Url	Specify the JDBC URL to the Oracle Business Analytics Warehouse in the format jdbc:oracle:thin:@<host>:<port>:<sid>. Replace <host>, <port> and <sid> with the values for the database hosting the ODI Repositories. For example, 'jdbc:oracle:thin:@US12345.us.company.com:1521:US12345'.

7. Click Test to display the Test Connection for: <connection> dialog.
8. From the **Agent** drop down list, select Local (No Agent).
9. Click Test to verify the details.
10. Click OK on the Information dialog.
11. Click OK on the Data Server: New dialog to display the Work Repository: New dialog.
12. On the Work Repository: New dialog, enter the appropriate information, as described in the table below.

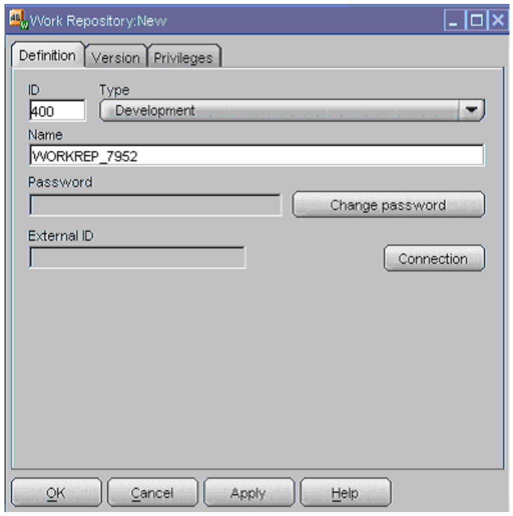
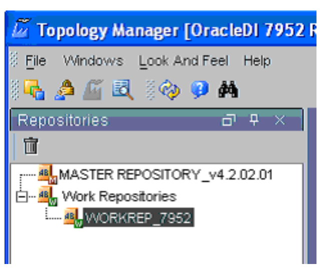


Table 4–7 Work Repository: New dialog fields

Field	Description
ID	Specify a unique ID (for example, 400).
Type	Select Development from the drop down list.
Name	Specify WORKREP_7952. This is the name of the work repository.

13. Click Apply, then click OK to save the details.

    Topology Manager displays the WORKREP_7952 repository in the Work Repositories list in the Repositories pane.



4.5.5 How to set up the ODI Topology

To set up the ODI Topology, you need to do the following:

- Set up the Oracle Data Servers (for more information, see Section 4.5.5.1, "Setting up the Oracle Data Servers").
- Set up the Agents (for more information, see Section 4.5.5.2, "How to set up the Agents").
- Set up the Data Source Number (for more information, see Section 4.5.5.3, "How to set up the Data Source Number").
- Set up the ODI Designer connection to the ODI Master Repository (for more information, see Section 4.5.5.4, "How to set up the ODI Designer connection to the ODI Master Repository").

- Create the ODI Work Repository (for more information, see Section 4.5.5.5, "How to import the Oracle BI Applications Work Repository").

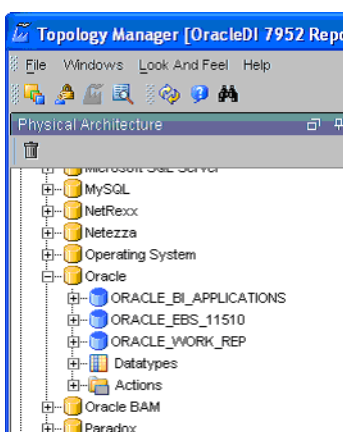
4.5.5.1 Setting up the Oracle Data Servers

To set up the Oracle Data Servers for Oracle Business Intelligence Applications, do the following:

- Locate the Physical Architecture tab in ODI Topology Manager (for more information, see Section 4.5.5.1.1, "How to display the Physical Architecture tab in Topology Manager").
- Set up the ORACLE_BI_APPLICATIONS Data Server (for more information, see Section 4.5.5.1.2, "How to set up the ORACLE_BI_APPLICATIONS Data Server").
- Set up the Physical Schema for the ORACLE_BI_APPLICATIONS Data Server (for more information, see Section 4.5.5.1.3, "How to set up the Physical Schema for the ORACLE_BI_APPLICATIONS Data Server").
- Set up the ORACLE_EBS_1510 Data Server (for more information, see Section 4.5.5.1.4, "How to set up the ORACLE_EBS_11510 Data Server").
- Set up the Physical Schema for the ORACLE_EBS_11510 Data Server (for more information, see Section 4.5.5.1.5, "How to set up the Physical Schema for the ORACLE_EBS_11510 Data Server").
- Set up the ORACLE_WORK_REP Data Server (for more information, see Section 4.5.5.1.6, "How to set up the ORACLE_WORK_REP Data Server").
- Set up the Physical Schema for the ORACLE_WORK_REP Data Server (for more information, see Section 4.5.5.1.7, "How to set up the Physical Schema for the ORACLE_WORK_REP Data Server").
- Set up the Logical Data Servers (for more information, see Section 4.5.5.1.8, "How to set up the Logical Data Servers").

4.5.5.1.1 How to display the Physical Architecture tab in Topology Manager To display the Physical Architecture tab in Topology Manager:

1. Start ODI Topology Manager.
2. Display the Physical Architecture tab.
3. Expand the Technologies node.
4. Expand the Oracle node to display the Physical Data Servers.



4.5.5.1.2 How to set up the ORACLE_BI_APPLICATIONS Data Server

To set up the ORACLE_BI_APPLICATIONS Data Server:

1. Start Topology Manager and display the Oracle node in the Physical Architecture tab (for more information, see Section 4.5.5.1.1, "How to display the Physical Architecture tab in Topology Manager").
2. Double click the ORACLE_BI_APPLICATIONS node to display the Data Server: <Name> dialog.
3. Display the Definition tab, and enter the appropriate information, as described in the table below.

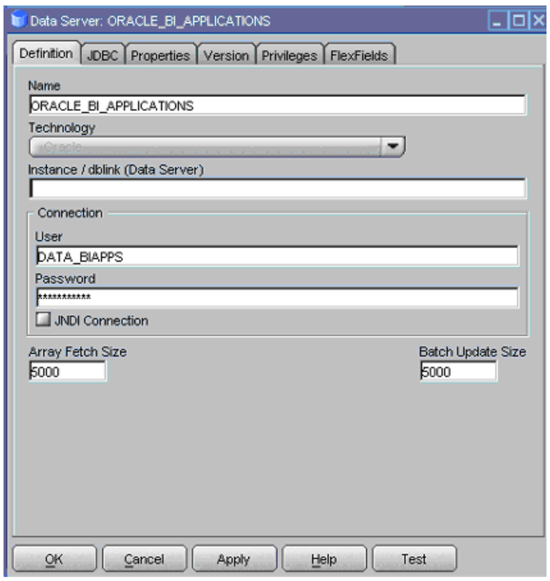


Table 4–8 Data Server: ORACLE_BI_APPLICATIONS dialog, Definition tab fields

Field	Description
Name	Do not change the default value ORACLE_BI_APPLICATIONS.
Technology	Do not change the default value Oracle.
Instance/dblink (Data Server)	Specify a database instance name. Use the Oracle SID name.
User	Specify 'DATA_BIAPPS'. This is the warehouse database user name.
Password	Specify 'DATA_BIAPPS'. This is default password for the warehouse database user name.
Array Fetch Size	Specify a value suitable to your environment (a typical value is 5000).
Batch Update Size	Specify a value suitable to your environment (a typical value is 5000).

4. Display the JDBC tab, and enter the appropriate information, as described in the table below.

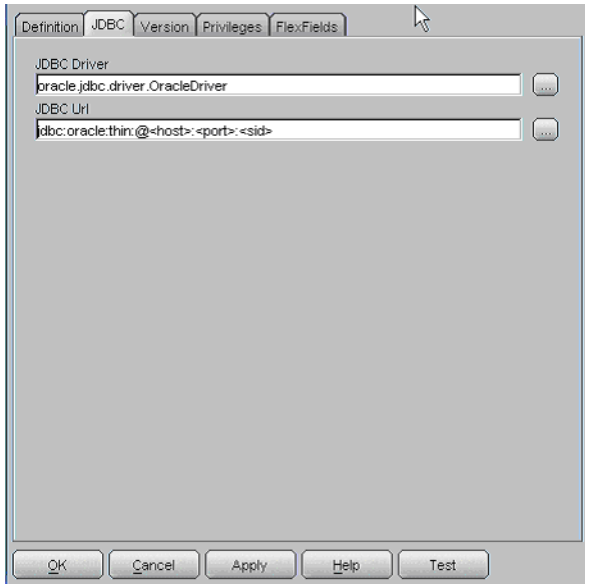


Table 4–9 Data Server: New dialog, JDBC tab fields

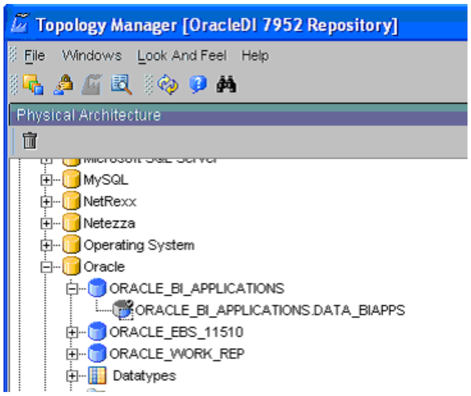
Field	Description
JDBC Driver	Specify oracle.jdbc.driver.OracleDriver.
JDBC Url	Specify in the format jdbc:oracle:thin:@<host>:<port>:<sid>. Replace <host>, <port> and <sid> with the values for the database hosting the ODI Repositories. For example, 'jdbc:oracle:thin:@US12345.us.company.com:1521:US12345'.

5. Click Test to display the Test Connection for: <Connection> dialog.
6. From the **Agent** drop down list, select Local (No Agent).
7. Click Test to verify the details.
8. Click OK on the Information dialog.
9. Click Apply, then click OK.

4.5.5.1.3 How to set up the Physical Schema for the ORACLE_BI_APPLICATIONS Data Server

To set up the Physical Schema for the ORACLE_BI_APPLICATIONS Data Server:

1. Start Topology Manager and display the Oracle node in the Physical Architecture tab (for more information, see Section 4.5.5.1.1, "How to display the Physical Architecture tab in Topology Manager").
2. Expand the ORACLE_BI_APPLICATIONS node.



3. Double click on ORACLE_BI_APPLICATIONS.DATA_BIAPPS to display the Physical Schema: <Name> dialog.

4. Display the Definition tab, and enter the appropriate information, as described in the table below.

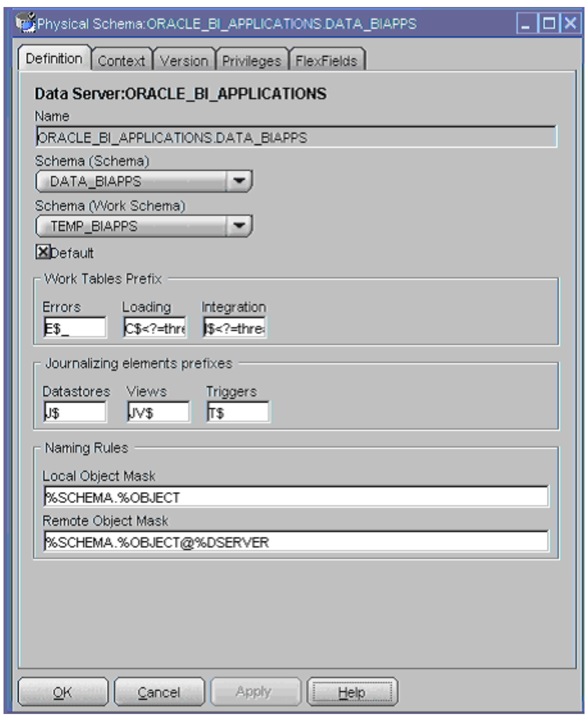


Table 4–10 Physical Schema: <Name> dialog, Definition tab fields

Field	Description
Schema (Schema)	Make sure that DATA_BIAPPS is selected from the drop down list.
Schema (Work Schema)	Make sure that TEMP_BIAPPS is selected from the drop down list.

Note: Do not change the other field values.

5. Click OK to save the details.

4.5.5.1.4 How to set up the ORACLE_EBS_11510 Data Server To set up the ORACLE_EBS_11510 Data Server:

1. Start Topology Manager and display the Oracle node in the Physical Architecture tab (for more information, see Section 4.5.5.1.1, "How to display the Physical Architecture tab in Topology Manager").
2. Double click the ORACLE_EBS_11510 node to display the Data Server: <Name> dialog.
3. Display the Definition tab, and enter the appropriate information, as described in the table below.

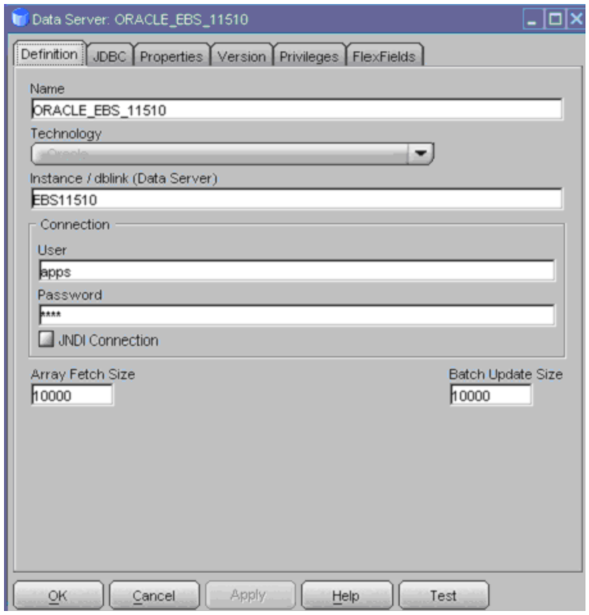


Table 4–11 Data Server: ORACLE_EBS_11510 dialog, Definition tab fields

Field	Description
Name	Do not change the default value ORACLE_EBS_11510.
Technology	Do not change the default value Oracle.
Instance/dblink (Data Server)	Specify a database instance name. Use the EBS 11.5.10 Oracle SID name. For example, EBS11510.
User	Specify 'APPS'. This is the EBS 11.5.10 APPS user name.
Password	Specify the APPS schema password (for example, APPS, or appropriate password).
Array Fetch Size	Specify a value suitable to your environment (a typical value is 5000).
Batch Update Size	Specify a value suitable to your environment (a typical value is 5000).

4. Display the JDBC tab, and enter the appropriate information, as described in the table below.

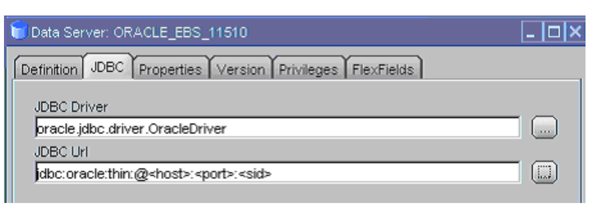


Table 4–12 Data Server: ORACLE_EBS_11510 dialog, JDBC tab fields

Field	Description
JDBC Driver	Specify oracle.jdbc.driver.OracleDriver.
JDBC Url	Specify in the format jdbc:oracle:thin:@<host>:<port>:<sid>. Replace <host>, <port> and <sid> with the values for the database hosting the EBS instance. For example, 'jdbc:oracle:thin:@US12345.us.company.com:1521:US11510'.

5. Click Test to display the Test Connection for: <connection> dialog.
6. From the **Agent** drop down list, select Local (No Agent).
7. Click Test to verify the details.
8. Click OK on the Information dialog.
9. Click Apply, then click OK.

4.5.5.1.5 How to set up the Physical Schema for the ORACLE_EBS_11510 Data Server

To set up the Physical Schema for the ORACLE_EBS_11510 Data Server:

1. Start Topology Manager and display the Oracle node in the Physical Architecture tab (for more information, see Section 4.5.5.1.1, "How to display the Physical Architecture tab in Topology Manager").
2. Expand the ORACLE_EBS_11510 node.
3. Double click on ORACLE_EBS_11510.APPS to display the Physical Schema: <Name> dialog.
4. Display the Definition tab, and enter the appropriate information, as described in the table below.

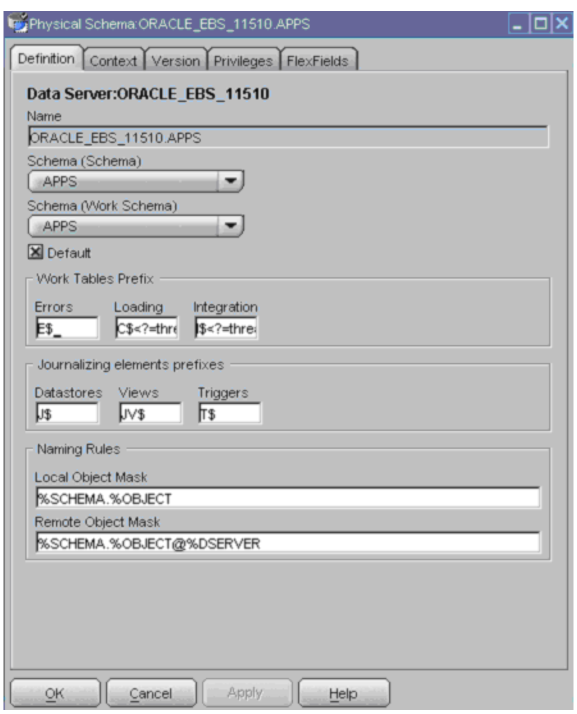


Table 4–13 Physical Schema: <Name> dialog, Definition tab fields

Field	Description
Schema (Schema)	Make sure that APPS is selected from the drop down list.
Schema (Work Schema)	Make sure that APPS is selected from the drop down list.

Note: Do not change the other field values.

5. Click OK to save the details.

4.5.5.1.6 How to set up the ORACLE_WORK_REP Data Server To set up the ORACLE_WORK_REP Data Server:

1. Start Topology Manager and display the Oracle node in the Physical Architecture tab (for more information, see Section 4.5.5.1.1, "How to display the Physical Architecture tab in Topology Manager").
2. Double click the ORACLE_WORK_REP node to display the Data Server: <Name> dialog.
3. Display the Definition tab, and enter the appropriate information, as described in the table below.

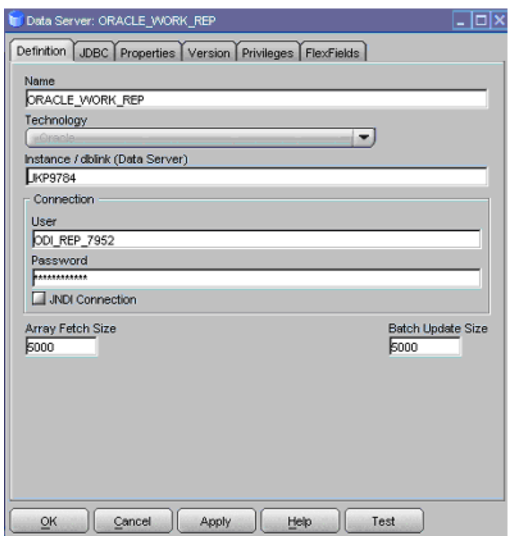


Table 4–14 Data Server: ORACLE_WORK_REP dialog, Definition tab fields

Field	Description
Name	Do not change the default value ORACLE_WORK_REP.
Technology	Do not change the default value Oracle.
Instance/dblink (Data Server)	Specify the name of the machine that hosts the ODI Master Repository. Use the Oracle SID name.
User	Specify 'ODI_REP_7952'. This is the ODI Repository user name.
Password	Specify 'ODI_REP_7952'. This is default password for the ODI Repository user name.

Table 4–14 (Cont.) Data Server: ORACLE_WORK_REP dialog, Definition tab fields

Field	Description
Array Fetch Size	Specify a value suitable to your environment (a typical value is 5000).
Batch Update Size	Specify a value suitable to your environment (a typical value is 5000).

4. Display the JDBC tab, and enter the appropriate information, as described in the table below.

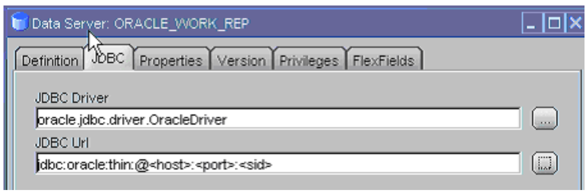


Table 4–15 Data Server: ORACLE_WORK_REP dialog, JDBC tab fields

Field	Description
JDBC Driver	Specify oracle.jdbc.driver.OracleDriver.
JDBC Url	Specify in the format jdbc:oracle:thin:@<host>:<port>:<sid>. Replace <host>, <port> and <sid> with the values for your database installation specific values. For example, 'jdbc:oracle:thin:@US12345.us.company.com:1521:US12345'.

5. Click Test to display the Test Connection for: <connection> dialog.
6. From the **Agent** drop down list, select Local (No Agent).
7. Click Test to verify the details.
8. Click OK on the Information dialog.
9. Click Apply, then click OK.

4.5.5.1.7 How to set up the Physical Schema for the ORACLE_WORK_REP Data Server To set up the Physical Schema for the ORACLE_WORK_REP Data Server:

1. Start Topology Manager and display the Oracle node in the Physical Architecture tab (for more information, see Section 4.5.5.1.1, "How to display the Physical Architecture tab in Topology Manager").
2. Expand the ORACLE_WORK_REP node.
3. Double click on ORACLE_WORK_REP.ODI_REP_7952 to display the Physical Schema: <Name> dialog.
4. Display the Definition tab, and enter the appropriate information, as described in the table below.

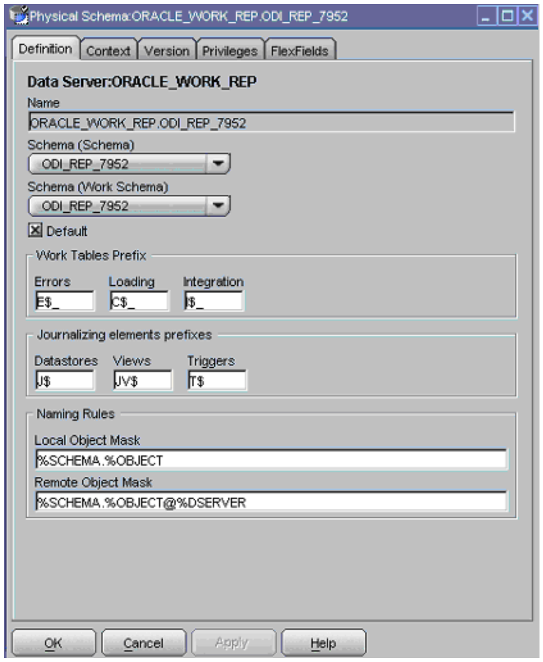


Table 4–16 Physical Schema: <Name> dialog, Definition tab fields

Field	Description
Schema (Schema)	Make sure that ODI_REP_7952 is selected from the drop down list.
Schema (Work Schema)	Make sure that ODI_REP_7952 is selected from the drop down list.

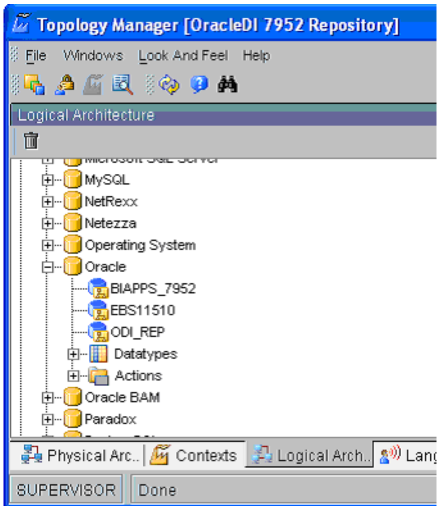
Note: Do not change the other field values.

5. Click OK to save the details.

4.5.5.1.8 How to set up the Logical Data Servers

To set up the Logical Data Servers:

1. Start ODI Topology Manager.
2. Display the Logical Architecture tab.
3. Expand the Technologies node.
4. Expand the Oracle node to display the Logical Data Servers.



The Oracle node should contain the following Logical Data Servers:

- BIAPPS_7952
- EBS11510
- ODI_REP

5. Edit the BIAPPS_7952 Logical Data Server and make sure that for the appropriate Context (for example, Development), the value in the **Physical Schemas** column is set to 'ORACLE_BI_APPLICATIONS.DATA_BIAPPS'.
6. Edit the EBS11510 Logical Data Server and make sure that for the appropriate Context (for example, Development), the value in the **Physical Schemas** column is set to 'ORACLE_EBS_11510.APPS'.
7. Edit the ODI_REP Logical Data Server and make sure that for the appropriate Context (for example, Development), the value in the **Physical Schemas** column is set to 'ORACLE_WORK_REP.ODI_REP_7952'.

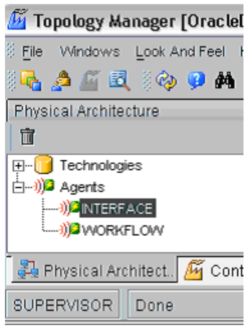
4.5.5.2 How to set up the Agents

This section explains how to set up the INTERFACE agent and the WORKFLOW agent in Topology Manager. To set up the Agents, do the following:

1. Set up the INTERFACE Agent (for more information, see Section 4.5.5.2.1, "How to set up the INTERFACE Agent").
2. Set up the WORKFLOW Agent (for more information, see Section 4.5.5.2.2, "How to set up the WORKFLOW Agent").

4.5.5.2.1 How to set up the INTERFACE Agent

1. Start Topology Manager, display the Physical Architecture tab, and expand the Agents node.



2. Double-click the INTERFACE node to display the Agent: <Name> dialog.
3. Display the Definition tab, and enter the appropriate information, as described in the table below.

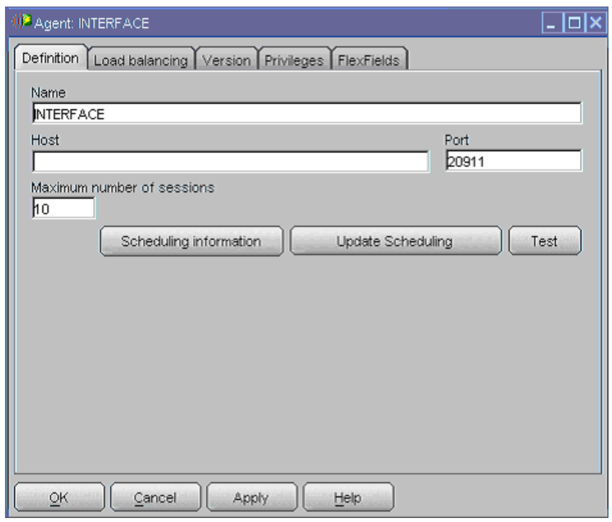


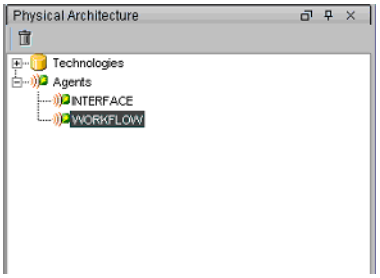
Table 4–17 Agent: <Name> dialog, Definition tab fields

Field	Description
Name	Do not change the default value INTERFACE.
Host	The host name of the machine that is running the Agent, typically the ODI installation machine. For example, localhost, or US12345.us.company.com.
Port	Do not change the default value. If you encounter a port conflict when you run the ODI Agents, you might need to use a different port number (for more information, see (Section 4.6.13, "How to Resolve Conflicts in ODI Agent Port Numbers").
Maximum number of sessions	The maximum number of interfaces that can run in parallel (default is 10). This value is used when load balancing.

4. Click Test to verify the details.
5. Click OK to close the Test window.
6. Click Apply, then click OK to save the details.

4.5.5.2.2 How to set up the WORKFLOW Agent How to set up the WORKFLOW Agent

1. Start Topology Manager, display the Physical Architecture tab, and expand the Agents node.



2. Double-click the WORKFLOW node to display the Agent: <Name> dialog.
3. Display the Definition tab, and enter the appropriate information, as described in the table below.

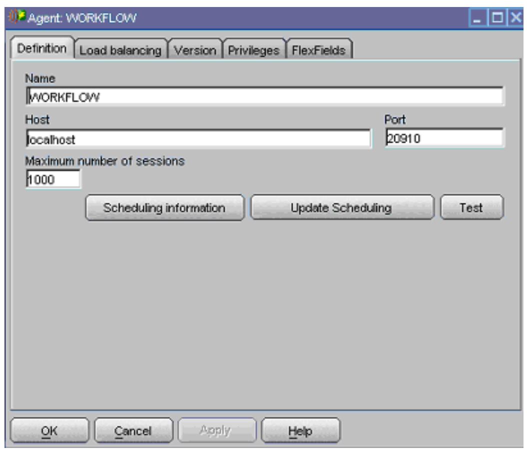


Table 4–18 Agent: <Name> dialog, Definition tab fields

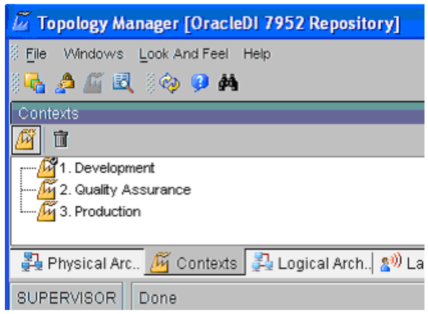
Field	Description
Name	Do not change the default value WORKFLOW.
Host	The host name of the machine that is running the Agent, typically the ODI installation machine. For example, localhost, or US12345.us.company.com.
Port	Do not change the default value. If you encounter a port conflict when you run the ODI Agents, you might need to use a different port number (for more information, see (Section 4.6.13, "How to Resolve Conflicts in ODI Agent Port Numbers").
Maximum number of sessions	The maximum number of interfaces that can run in parallel (default is 1000). This value is used when load balancing.

4. Click Test to verify the details.
5. Click OK to close the Test window.
6. Click Apply, then click OK to save the details.

4.5.5.3 How to set up the Data Source Number

The section describes how to uniquely identify the data source for each context using a Data Source Number.

1. Start Topology Manager, and display the Contexts tab.



   The Contexts tab displays three contexts: Development, Quality Assurance, and Production.

2. Edit each Context, and do the following:

   a. Display the Flexfields tab, and enter the appropriate information, as described in the table below.

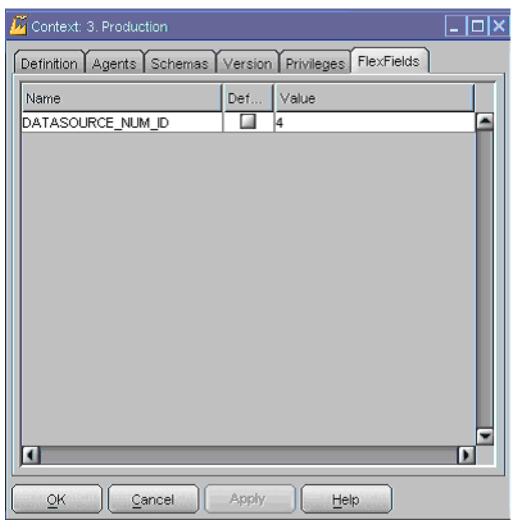


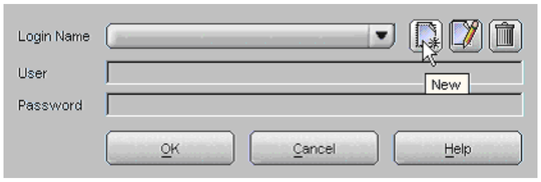
Table 4–19 Context: <Name> dialog, FlexFields tab fields

Field	Description
DATASOURCE_NUM_ID	Specify the value according to your data source type. The default is 4, which is the ID for Oracle EBS source systems.
Def...	Clear this check box.

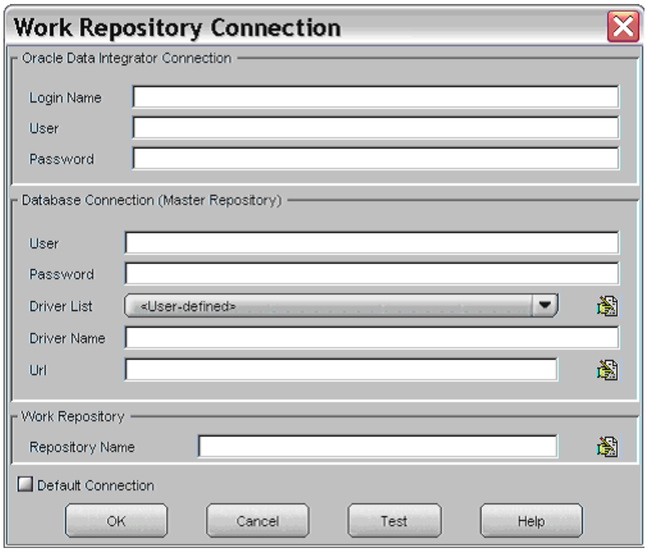
4.5.5.4 How to set up the ODI Designer connection to the ODI Master Repository

To set up the ODI Designer connection to the ODI Master Repository:

1. On the ODI machine, choose Start, then All Programs, then Oracle Data Integrator, then Designer to display the Designer login dialog.



2. Click New to display the Work Repository Connection dialog.



3. In the Work Repository Connection dialog, enter the appropriate information, as described in the table below.

Table 4–20 Work Repository Connection dialog fields

Field	Description
Login Name	Specify 'OracleDI 7952 Repository'.
User	Specify SUPERVISOR. This is the ODI Administrator database user.
Password	Specify SUPERVISOR. This is the default password for the ODI Administrator database user.
User	Specify ODI_REP_7952. This is the ODI Repository database user name.
Password	Specify SUPERVISOR. This is the default password for the ODI Repository database user name.
Driver List	Select 'Oracle JDBC Driver'.
Driver Name	Specify 'oracle.jdbc.driver.OracleDriver'.
URL	Specify the JDBC URL to the Oracle Business Analytics Warehouse in the format jdbc:oracle:thin:@<host>:<port>:<sid>. Replace <host>, <port> and <sid> with your database installation specific values. For example, 'jdbc:oracle:thin:@US12345.us.company.com:1521:US12345'.

Table 4–20 (Cont.) Work Repository Connection dialog fields

Field	Description
Repository Name	Specify WORKREP_7952.
Default Connection	Select this check box.

4. Click Test to verify the details.
5. Click OK on the Information dialog.
6. Click OK on Repository Connections dialog.
7. Click OK on the Oracle Data Integrator Login dialog to log in to ODI Designer.

4.5.5.5 How to import the Oracle BI Applications Work Repository

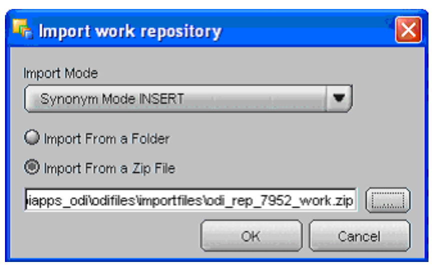
This section explains how to create the Oracle Business Intelligence Applications Work Repository by importing a zipped repository file into ODI. Importing this repository creates a project in 'ODI Designer called Oracle BI Applications 7.9.5.2'.

To import the Oracle BI Applications Work Repository:

1. In ODI Designer, select an appropriate Context from the Context drop down list.

   For example, if you are in a production environment, select Production.

2. Choose File, then Import, then Work Repository to display the Import work repository dialog.



3. In the 'Import work repository' dialog, enter the appropriate information, as described in the table below.

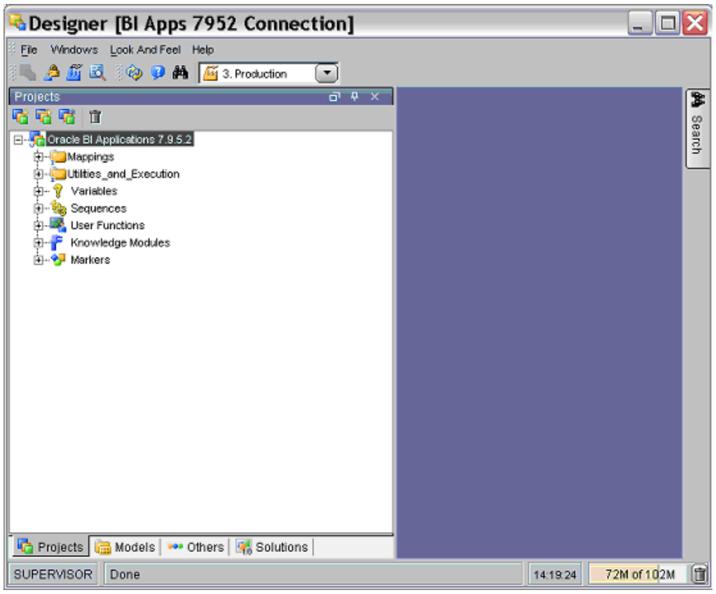
Table 4–21 Import work repository dialog fields

Field	Description
Import Mode	Select Synonym Mode INSERT from the drop down list.
Import From a Zip File	Select the **Import from a Zip File** radio button.
<Zip file path>	Use this field to specify the following path and ZIP file name: `$ODI_HOME/oracledi/biapps_odi/odifiles/importfiles/odi_rep_7952_work.zip`

4. Click OK to import the Work Repository.

   Tip: Depending on the speed of your ODI machine, ODI Designer can take up to between one and three hours to import the repository. Do not interrupt the import process during this time, even though ODI Designer does not provide a progress report during the import process.

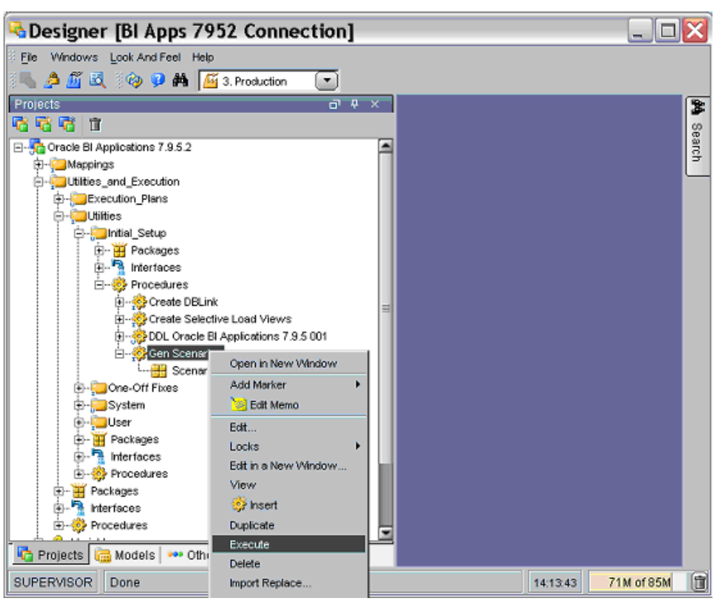
5. Display the Projects view to display the Oracle BI Applications 7.9.5.2 project that contains the work repository objects.



4.5.6 How to generate the required ODI Scenarios

This section explains how to generate the required scenarios in ODI, which are used to control E-LT processes.

1. In ODI Designer, log in as SUPERVISOR, and display the Projects view.
2. Expand the Oracle BI Applications 7.9.5.2 project.
3. Select Utilities_and_Execution, then Utilities, then Initial_Setup, then Procedures.
4. Right click on Gen Scenarios and select Execute to start creating the scenarios required for Oracle Business Intelligence Applications.

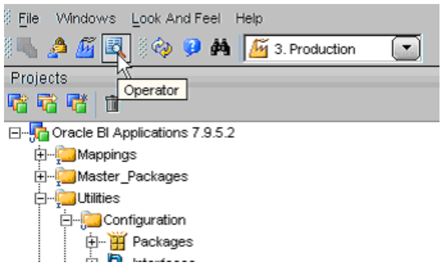


ODI Designer displays the Execution dialog.

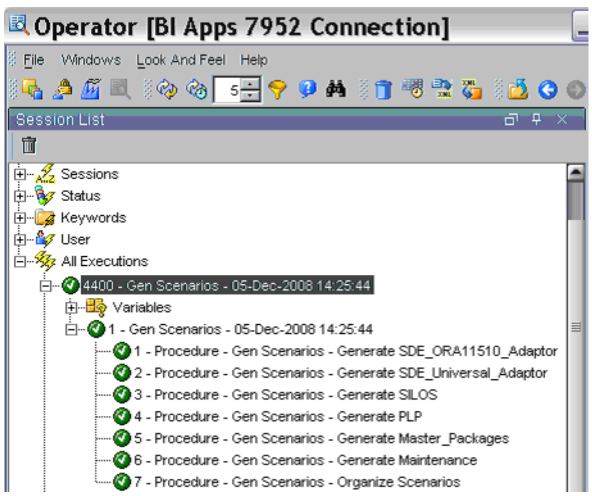
5. Use the Execution dialog to specify the appropriate context in the **Context** field, and specify Local (No Agent) as the **Agent** value.

   You use ODI Operator to monitor executed processes.

6. To display ODI Operator, click on the Operator icon on the ODI Designer tool bar.



7. In ODI Operator, display the Session List tab, expand the Sessions node, and verify the session running the Gen Scenario procedure.



   Make sure that all of the procedures on the Gen Scenarios node execute successfully.

4.5.7 How to install the Schema Objects for the Oracle Business Analytics Warehouse

This section explains how to install the Schema Objects required by the Oracle Business Analytics Warehouse.

To install the Schema Objects, do the following:

1. Generate the required procedures (for more information, see Section 4.5.7.1, "How to Generate the Data Warehouse DDL Procedure").
2. Execute the required procedures (for more information, see Section 4.5.7.2, "Execute the Data Warehouse DDL Procedure").
3. Execute the Load Control Seed Data Package (for more information, see Section 4.5.7.3, "Running Load Control Seed Data Package").
4. Execute the Create DBLink procedure (for more information, see Section 4.5.7.4, "Running Create DBLink Procedure").
5. Verify the INDEX_TABLESPACE setting in ODI (for more information, see Section 4.5.7.5, "Verifying the INDEX_TABLESPACE Setting").

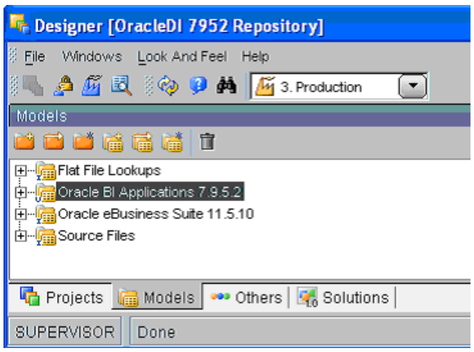
Notes

- Before performing the following tasks, make sure that the Oracle Business Analytics Warehouse schema is empty.
- To maximize performance, Oracle recommends that you implement range partitioning for fact tables before you create the Oracle Business Analytics Warehouse Target Schema Objects. For more information about implementing partitions, see Section 4.6.9, "How to add partitions to Data Warehouse tables".

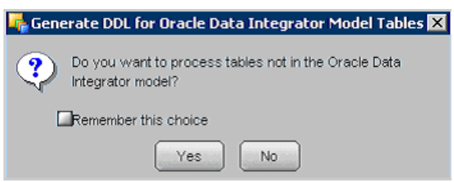
4.5.7.1 How to Generate the Data Warehouse DDL Procedure

To generate the required DDL Procedure:

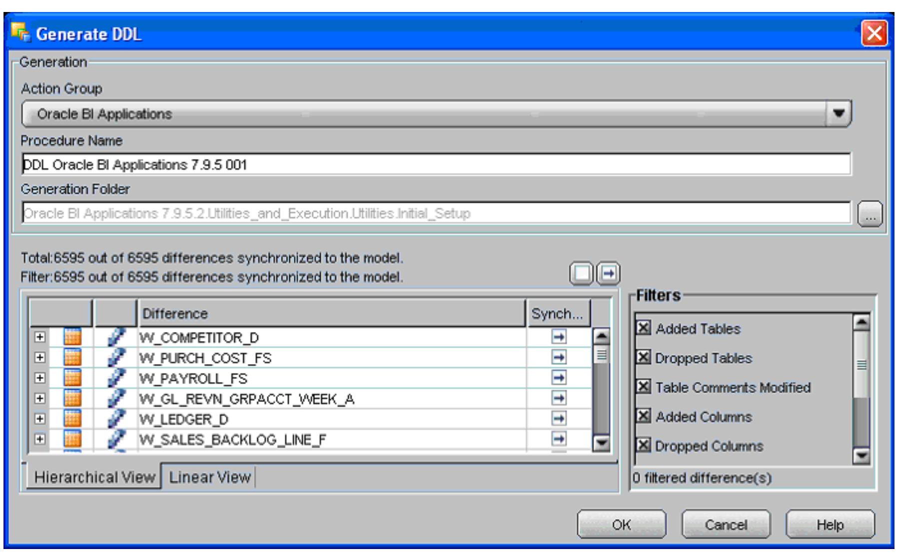
1. In ODI Designer, log in as SUPERVISOR, and display the Models view.



2. Right-click on the Oracle BI Applications 7.9.5.2 node, then select Generate DDL from the right-click menu to display the Generate DDL for Oracle Data Integrator Model Tables dialog.



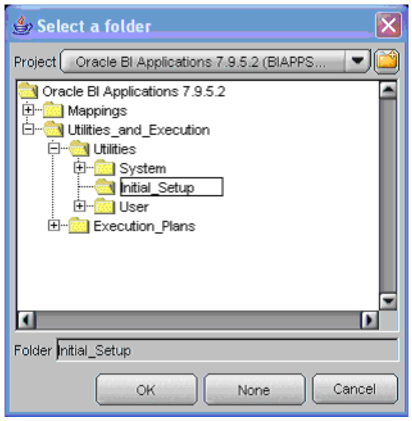
3. Click No to display the Generate DDL dialog.



4. To select all objects for generation, click the right arrow (-->) in the top right hand corner of the list of objects.

   A right arrow (-->) is displayed in the **Synchronization** column for each row in the objects table, to indicate that all objects are selected.

5. Click the (...) button to the right of the **Generation Folder** field to display the **Select a folder** dialog, and select the \Initial_Setup folder.

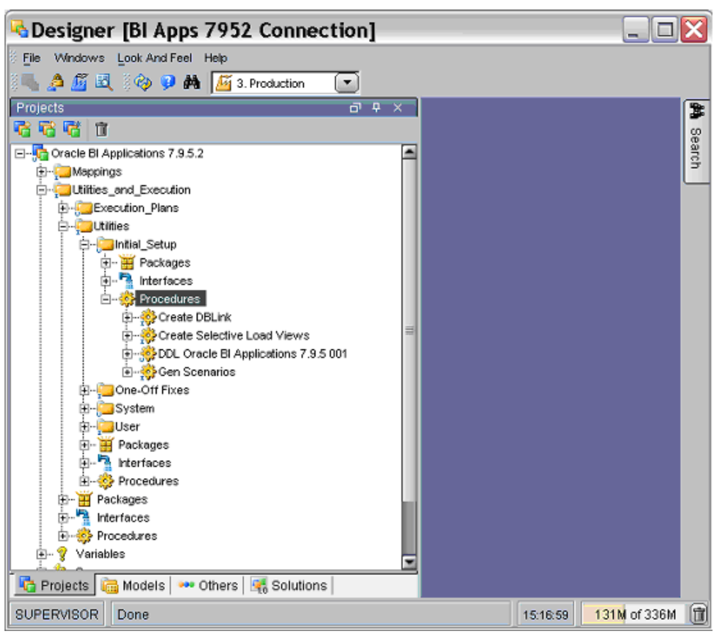


6. Click OK to start the generation of DDL procedures.

4.5.7.2 Execute the Data Warehouse DDL Procedure

To execute the required procedures:

1. In ODI Designer, log in as SUPERVISOR, and display the Projects view.
2. Expand the Oracle BI Applications 7.9.5.2 project.
3. Select Utilities_and_Execution, then Utilities, then Initial_Setup, then Procedures.

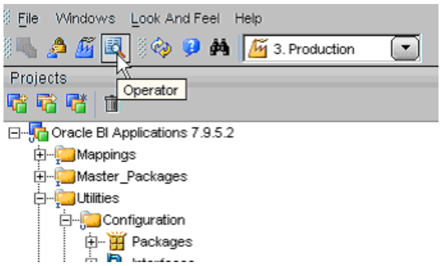


4. Right click on 'DDL Oracle BI Applications 7.9.5 001' and select Execute.
5. Right click on 'Create Selective Load Views' and select Execute to display the Execute dialog.

6. Use the Execution dialog to specify the appropriate context in the **Context** field, and specify INTERFACE as the **Agent** value.

   You use ODI Operator to monitor executed processes.

7. To display ODI Operator, click on the Operator icon on the ODI Designer tool bar.



4.5.7.3 Running Load Control Seed Data Package

To run the Load Control Seed Data Package:

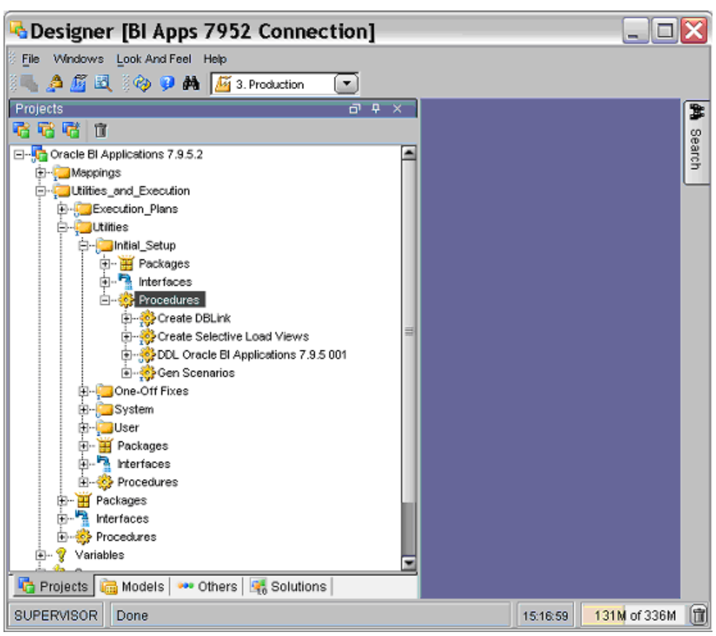
1. In ODI Designer, log in as SUPERVISOR, and display the Projects view.
2. Expand the Oracle BI Applications 7.9.5.2 project.
3. Select Utilities_and_Execution, then Utilities, then Initial_Setup, then Packages.
4. Right click on 'Load Control Seed Data' and select Execute.

   You use ODI Operator to monitor executed processes.

4.5.7.4 Running Create DBLink Procedure

To run the Create DBLink Procedure:

1. In ODI Designer, log in as SUPERVISOR, and display the Projects view.
2. Expand the Oracle BI Applications 7.9.5.2 project.
3. Select Utilities_and_Execution, then Utilities, then Initial_Setup, then Procedures.



4. Right click on Create DBLink and select Execute.

You use ODI Operator to monitor executed processes.

4.5.7.5 Verifying the INDEX_TABLESPACE Setting

You need to make sure that the INDEX_TABLESPACE setting in ODI is configured to use the BIAPPS_INDEX index that you created in Section 4.3.3.1, "Creating the Required Databases and Tablespaces".

To verify the INDEX_TABLESPACE setting:

1. In ODI Designer, log in as SUPERVISOR, and display the Models view.
2. Edit the Oracle BI Applications 7.9.5.2 project, to display the Model:<*Name*> dialog.
3. Display the FlexFields tab.
4. Make sure that the INDEX_TABLESPACE FlexField is set with a value BIAPPS_ INDEX.

   This FlexField is used by Index Management procedures in related Knowledge Modules (KM).

4.5.8 Setting Up Oracle BI Applications Configuration Manager

Oracle BI Applications Configuration Manager is a web-application that enables you to:

- Create and manage Execution Plans for E-LT executions.
- Set E-LT parameters.
- View Master Package structures.
- Run and monitor Execution Plans.

You can also set up connections to different environments, and use a single deployment of Oracle BI Applications Configuration Manager to manage them.

The source files for Oracle BI Applications Configuration Manager are installed when you install Oracle Business Intelligence Applications on a machine. This section explains how to use those files to set up and run Oracle BI Applications Configuration Manager, and contains the following topics:

- Section 4.5.8.1, "Overview to Setting Up Oracle BI Applications Configuration Manager"
- Section 4.5.8.2, "How to Execute the Oracle BI Applications Configuration Manager Setup Script"
- Section 4.5.8.3, "How to Set Up Oracle BI Applications Configuration Manager on Windows"
- Section 4.5.8.4, "How to Set Up Oracle BI Applications Configuration Manager on UNIX/Linux"
- Section 4.5.8.5, "About accessing the WebLogic Administration Console"
- Section 4.5.8.6, "How to Launch Oracle BI Applications Configuration Manager"
- Section 4.5.8.7, "How to Launch Oracle BI Applications Configuration Manager in Debug Mode"
- Section 4.5.8.8, "How to create a Connection For Your Development Environment"

- Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection"

For information about using Oracle BI Applications Configuration Manager, see Section 4.6.3, "Using Oracle BI Applications Configuration Manager".

4.5.8.1 Overview to Setting Up Oracle BI Applications Configuration Manager

Oracle BI Applications Configuration Manager uses a Web Logic application server with Oracle Application Development Framework (ADF) runtime that you must set up after you have installed Oracle Business Intelligence Applications on a machine.

After you have run the Oracle Business Intelligence Applications installer on a machine, the Oracle BI Applications Configuration Manager files are installed in the \OracleBI\dwrep\biapps_configmgr\ directory. To install Oracle BI Applications Configuration Manager on a different machine (for example, a UNIX or Linux machine), you copy the appropriate files from the \OracleBI\dwrep\biapps_configmgr\ directory to the target machine. On the target machine, you then set up Oracle BI Applications Configuration Manager as described in this section.

When deciding where to install WebLogic Server and deploy Oracle BI Applications Configuration Manager, note the following:

- You can install WebLogic Server and deploy Oracle BI Applications Configuration Manager on any Windows, UNIX, or Linux machine. For example, you can install WebLogic Server and deploy Oracle BI Applications Configuration Manager on the machine that hosts the ODI Agents and Tools.
- For performance reasons, the machine on which you install Oracle WebLogic Server and deploy Oracle BI Applications Configuration Manager must be the same Local Area Network as the Oracle Business Analytics Warehouse machine.
- The machine on which you run the WebLogic server for Oracle BI Applications Configuration Manager must have JDK 1.6.0.05 or a higher 1.6.x version.

To set up Oracle BI Applications Configuration Manager:

1. Execute the setup script for Oracle BI Applications Configuration Manager, as described in Section 4.5.8.2, "How to Execute the Oracle BI Applications Configuration Manager Setup Script".
2. If you are deploying Oracle BI Applications Configuration Manager from a Windows machine, follow the steps in Section 4.5.8.3, "How to Set Up Oracle BI Applications Configuration Manager on Windows"
3. If you are deploying Oracle BI Applications Configuration Manager from a UNIX or Linux machine, follow the steps in Section 4.5.8.4, "How to Set Up Oracle BI Applications Configuration Manager on UNIX/Linux"

4.5.8.2 How to Execute the Oracle BI Applications Configuration Manager Setup Script

The biappstx_setup.sql file sets up the required GRANTs on the repositories used by Oracle BI Applications Configuration Manager.

To execute the Oracle BI Applications Configuration Manager setup script:

1. On the ODI machine, open the biappstx_setup.sql in a text editor or SQL tool, and make sure that the schema names that are defined by the DEFINE statements at the top of the file match the schema names that you have used in your deployment.

The biappstx_setup.sql file is located in the $ODI_HOME\oracledi\biapps_odi\dbfiles\ directory. For example, D:\OraHome_1\oracledi\biapps_odi\dbfiles\.

The default values are defined as follows:

```
/*  DEFINEs for default schema names  */
--ODI Repository default schema
DEFINE L_ODI_SCHEMA   = ODI_REP_7952
--OBIA Data warehouse default schema
DEFINE L_OBIA_SCHEMA  = DATA_BIAPPS
--Configuration Manager default schema
DEFINE L_TX_SCHEMA    = DATA_BIAPPSTX
```

2. Using Oracle SQL Developer or Oracle SQL*Plus, connect to your database machine as a user with DBA privileges (for example, SYS or SYSTEM), and execute the biappstx_setup.sql file.

   Alternatively, you can execute the individual SQL commands in the biappstx_setup.sql file.

4.5.8.3 How to Set Up Oracle BI Applications Configuration Manager on Windows

This section provides instructions on how to install Oracle WebLogic Server 10.3 + Application Development Framework (ADF) 11g Runtime on Windows, create a WebLogic domain, and deploy the Oracle Business Intelligence Applications Configuration Manager to the created domain in the installed Oracle WebLogic Server instance.

To set up Oracle BI Applications Configuration Manager on Windows:

1. On the target machine, install JDK 1.6.0_05 or a higher 1.6.x version on the machine where Oracle WebLogic Server will be installed.

   Do not use spaces in the directory path name.

   If you are installing Oracle BI Applications Configuration Manager on the machine on which the Oracle BI Applications installer was run, skip Step 2 and go straight to Step 3.

2. If you are installing Oracle BI Applications Configuration Manager on a machine different from the one on which the Oracle BI Applications installer was run, copy the following files from the \OracleBI\dwrep\biapps_configmgr\ directory on the Oracle Business Intelligence Applications installation machine to a directory on the target machine, as follows:

   a. On the Oracle Business Intelligence Applications installation machine, navigate to the OracleBI\dwrep\biapps_configmgr\ directory.

   b. Copy the following files from the \OracleBI\dwrep\biapps_configmgr\ directory to an equivalent directory on the target machine:

      ```
      biapps_configmgr.ear
      jdevstudio11110install.jar
      silent.xml
      WIN_1_HowToInstallBIAppsConfigMgr.txt
      WIN_2_RunJDevInstallerSilent.cmd
      WIN_3_CreateDomain.txt
      WIN_4_DeployBIAppsConfigMgr.cmd
      ```

3. Install WebLogic Server10.3 and ADF 11g Runtime, as follows:

a. Edit silent.xml and specify the full path name to the WebLogic (BEA) Home directory by setting the value for the data-value name 'BEAHOME'.

   This is the location where Oracle WebLogic Server will be installed. For example, D:\Weblogic\.

b. Edit WIN_2_RunJDevInstallerSilent.cmd, and set the JAVA_LOCATION variable to the path to the java.exe (JDK 1.6.0_05 or higher 1.6 version) on your machine, then save the changes to the file.

   For example, D:\jre1.6.0_02\bin.

c. Execute the WIN_2_RunJDevInstallerSilent.cmd file.

   This script installs WebLogic Server + Application Developer Framework and ADF Runtime Libraries. WebLogic is installed in the directory specified by 'BEAHOME' in silent.xml.

4. Set up the required WebLogic domain, as follows:

   a. Launch the Oracle WebLogic Configuration Wizard by selecting Start > Programs > Oracle Fusion Middleware BEAHOME > Weblogic 10.3 > Tools > Configuration Wizard.

   b. In the Welcome screen, select the **Create a New WebLogic domain** radio button, then click Next.

   c. In the Select Domain Source screen, select the **Generate a domain configured automatically to support the following products:** radio button.

   d. Select the **Application Development Framework** check box, then click Next.

      Note: This option is mandatory for Oracle BI Applications Configuration Manager deployment.

   e. In the Configure Administrator Username and Password screen, enter User name and User Password, confirm the password, then click Next.

   f. In the Configure Server Start Mode and JDK screen, retain the default selection of **Development Mode** for the WebLogic Domain Startup Mode.

   g. In the JDK Selection box, select the location of the JDK 1.6 installation on the machine, then click Next.

   h. In the Customize Environment and Services settings screen, select the **No** radio button, then click Next.

   i. In the Create WebLogic Domain screen, specify the Domain name.

      Note: Do not use spaces. Do not change the values for Domain location or Application location.

   j. At the summary screen, do not select the **Start Admin Server** check box.

      Note: Oracle WebLogic Server should not be running for the next step.

5. Deploy the Oracle BI Applications Configuration Manager application, as follows:

   Note: Before executing this step, make sure the Oracle WebLogic Server is not running.

   a. Edit the WIN_4_DeployBIAppsConfigMgr.cmd file.

   b. Set BEAHOME to the install location of Oracle WebLogic Server as specified in the silent.xml file in Step 3.

c. Set USER_DOMAIN to the name of the WebLogic domain that you created in step 4.

d. Set EAR_FILE_PATH to the location of the biapps_configmgr.ear file.

e. Save changes to the file.

f. Execute the WIN_4_DeployBIAppsConfigMgr.cmd file by double-clicking the file or by executing from a command prompt window.

   This cmd file will deploy the Oracle BI Applications Configuration Manager application using the biapps_configmgr.ear file to the Oracle WebLogic Server installed in Step 3 and in the domain you have created in Step 4.

   The script starts the Oracle WebLogic Server after the BI Applications Configuration Manager is deployed. A notice similar to "<Notice> <WebLogicServer> <BEA-000360> <Server started in RUNNING mode>" is displayed in the command line window to indicate that the WebLogic Server is running and listening at the default port 7001.

   Launch a supported browser with the URL for Oracle BI Applications Configuration Manager - http://<WebLogic Server Host>:7001/biapps.

   For information on how to log in and use Oracle BI Applications Configuration Manager, refer to Section 4.5.8.6, "How to Launch Oracle BI Applications Configuration Manager".

   The URL for the WebLogic Console is:

   ```
   http://<Weblogic Server Host>:7001/console
   ```

   The credentials required to log in to the WebLogic Console are the username and password you entered in Step 4.

4.5.8.4 How to Set Up Oracle BI Applications Configuration Manager on UNIX/Linux

This section provides instructions on how to install Oracle WebLogic Server 10.3 + Application Development Framework (ADF) 11g Runtime on UNIX or Linux, create a WebLogic domain, and deploy the Oracle Business Intelligence Applications Configuration Manager to the created domain in the installed Oracle WebLogic Server instance.

To deploy Oracle Business Intelligence Applications Configuration Manager, you must copy the setup files from the Oracle Business Intelligence Applications installation machine to a UNIX or Linux machine, as described in this section.

To set up Oracle BI Applications Configuration Manager on UNIX or Linux:

1. On the target machine, install JDK 1.6.0_05 or a higher 1.6.x version on the machine where Oracle WebLogic Server will be installed.

2. Copy the following files from the \OracleBI\dwrep\biapps_configmgr\ directory on the Oracle Business Intelligence Applications installation machine to a directory on the target machine, as follows:

   a. On the Oracle Business Intelligence Applications installation machine, navigate to the OracleBI\dwrep\biapps_configmgr\ directory.

   b. Copy the following files from the \OracleBI\dwrep\biapps_configmgr\ directory to an equivalent directory on the target machine:

      ```
      biapps_configmgr.ear
      jdevstudio11110install.jar
      silent.xml
      ```

```
UNIX_1_HowToInstallBIAppsConfigMgr.txt
UNIX_2_RunJDevInstallerSilent.sh
UNIX_3_CreateDomain.txt
UNIX_4_DeployBIAppsConfigMgr.sh
```

Note: The UNIX_2_RunJDevInstallerSilent.sh and UNIX_4_DeployBIAppsConfigMgr.sh scripts must have read, write, and execute permissions. If the files do not have these permissions, modify them using the chmod command.

When you copy across these files, use a MS-DOS to UNIX conversion tool, convert the script files to UNIX format (that is, remove the carriage return and line feed characters). There are many MS-DOS to UNIX conversion tools that are freely available for download on the Internet. Alternatively, you can manually remove the carriage return and line feed character (/r) from the script files by running the following commands:

```
tr -d '\r'<UNIX_2_RunJDevInstallerSilent.sh>tmp.sh
mv tmp.sh UNIX_2_RunJDevInstallerSilent.sh
```

Repeat for the UNIX_4_DeployBIAppsConfigMgr.sh file.

3. Install WebLogic Server10.3 and ADF 11g Runtime, as follows:

   a. Edit silent.xml and specify the full path name to the WebLogic (BEA) Home directory by setting the value for the data-value name 'BEAHOME'.

      This is the location where Oracle WebLogic Server will be installed.

   b. Edit UNIX_2_RunJDevInstallerSilent.sh, and set the JAVA_LOCATION variable to the path to the java binary (JDK 1.6.0_05 or higher 1.6 version) on your machine, then save the changes to the file.

      For example, /jre1.6.0_02/bin.

   c. Execute the command using:

      ```
      ./UNIX_2_RunJDevInstallerSilent.sh
      ```

      Note: The script should be run under the bash shell

      This script installs WebLogic Server + Application Developer Framework and ADF Runtime Libraries. WebLogic is installed in the directory specified by 'BEAHOME' in silent.xml.

4. Set up the required WebLogic domain, as follows:

   a. Launch the Oracle WebLogic Configuration Wizard by opening a command shell. Go to the \common\bin subdirectory of the WebLogic product installation directory, and execute the following command:

      ```
      sh config.sh -mode=console
      ```

      Note: The command and arguments must be entered in lower case.

   b. At the Welcome prompt, choose "1 Create a new WebLogic domain" by typing 1, then press Enter.

   c. At the Select Domain Source prompt, select "1 Choose Weblogic Platform components" by typing 1, then press Enter.

   d. At the Application Template Selection prompt, select "Application Development Framework" by typing 1, then press Enter.

Note: This option is mandatory for Oracle BI Applications Configuration Manager deployment.

e. At the Application Template Selection prompt, type next and press Enter.

f. At the Configure Administrator Username and Password prompt, create a user to be assigned to the Administrator role. This user is the default administrator used to start development mode servers.

g. Follow the on screen instructions to set the user password and confirm user password.

   Optionally, change the default user name.

h. At the Domain Mode Configuration prompt, select "1 Development Mode" by typing 1, then press Enter.

i. At the Java SDK Selection prompt: specify Java SDK 1.6.0_05 or higher JDK 1.6 location.

j. At the Choose Configuration Option prompt: do not change the default settings by typing 1, then press Enter.

k. At the Select the target domain directory for this domain prompt, accept the default target location for the domain directory.

   Note: Do not change the default value.

l. At the Select the applications directory for this domain prompt, accept the default target location for the application directory.

   Note: Do not change the default value.

m. At the Edit Domain Information prompt, select the default name 'base_domain'.

   Alternatively, specify a different name if required.

n. Complete the domain creation.

5. Deploy the Oracle BI Applications Configuration Manager application, as follows:

   Note: Before executing this step, make sure the Oracle WebLogic Server is not running.

   a. Edit the UNIX_4_DeployBIAppsConfigMgr.sh file.

   b. Set BEAHOME to the install location of Oracle WebLogic Server as specified in the silent.xml file in Step 3.

   c. Set USER_DOMAIN to the name of the WebLogic domain that you created in step 4.

   d. Set EAR_FILE_PATH to the location of the biapps_configmgr.ear file.

   e. Save changes to the file.

   f. Execute the UNIX_4_DeployBIAppsConfigMgr.sh file using:

```
./UNIX_4_DeployBIAppsConfigMgr.sh
```

   Note: The script should be run under the bash shell

   This cmd file will deploy the Oracle BI Applications Configuration Manager application using the biapps_configmgr.ear file to the Oracle WebLogic Server installed in Step 3 and in the domain you have created in Step 4.

The script starts the Oracle WebLogic Server after the BI Applications Configuration Manager is deployed. A notice similar to "<Notice> <WebLogicServer> <BEA-000360> <Server started in RUNNING mode>" is displayed in the command line window to indicate that the WebLogic Server is running and listening at the default port 7001.

For information on how to log in and use Oracle BI Applications Configuration Manager, refer to Section 4.5.8.6, "How to Launch Oracle BI Applications Configuration Manager".

The URL for the WebLogic Console is:

```
http://<Weblogic Server Host>:7001/console
```

The credentials required to log in to the WebLogic Console are the username and password you entered in Step 4.

4.5.8.5 About accessing the WebLogic Administration Console

The WebLogic Admin Console is accessed from the URL:

http://<IP or Name of machine where WebLogic is installed>:7001/biapps. For example,

The default user name and password is weblogic/weblogic.

Log into the BI Applications Configuration Manager by accessing the URL:

http://<IP or Name of machine where WebLogic is installed>:7001/biapps

The default username and password is data_biappstx/ data_biappstx.

4.5.8.6 How to Launch Oracle BI Applications Configuration Manager

You launch Oracle BI Applications Configuration Manager to enable you to create and manage E-LT execution plans, set E-LT parameters, and monitor E-LT processes.

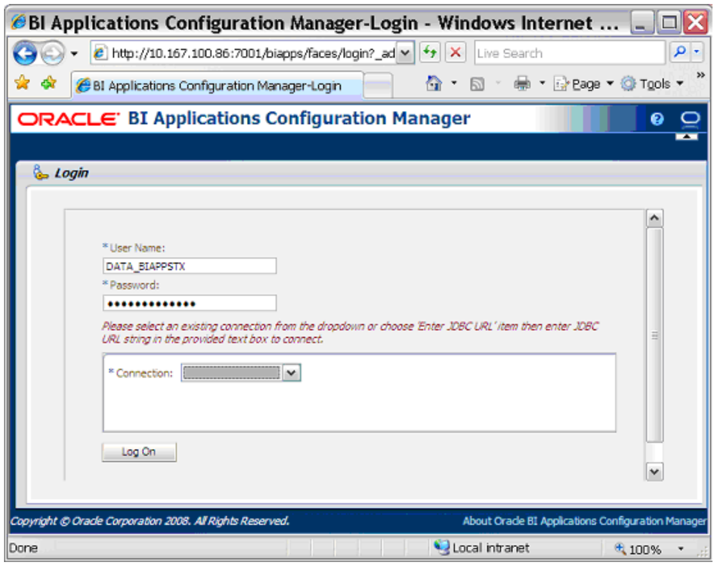
For a list of supported browsers and requirements, see *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*.

To start Oracle BI Applications Configuration Manager for the first time:

1. Open a supported Web browser.
2. Enter or select the Oracle BI Applications Configuration Manager URL using the fully qualified host name used by your installation to display the Login page.

   Use the URL format: http://<*hostname*>:7001/biapps. For example:

   ```
   http://mymachine.us.company.com:7001/biapps
   ```

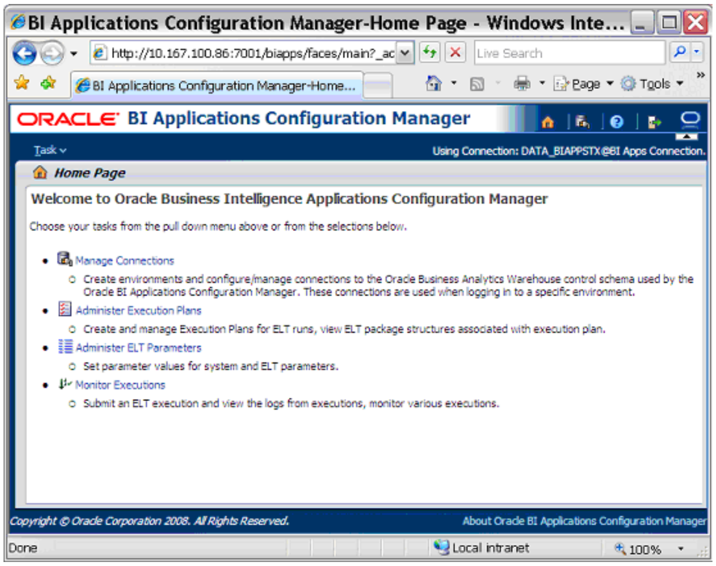


3. Enter appropriate login information, as described in the table below, then click Log On.

Table 4–22 Oracle BI Applications Configuration Manager Login page fields

Field Name	**Description**
User Name	Enter 'DATA_BIAPPSTX'.
Password	Enter 'DATA_BIAPPSTX' (default password) or a different password if you have changed it.
Connection	Select 'Enter JDBC URL'.
JDBC URL	Specify the JDBC URL to the Oracle Business Analytics Warehouse in the format jdbc:oracle:thin:@<host>:<port>:<sid>. Replace <host>, <port> and <sid> with the values for the database hosting the ODI Repositories. For example, 'jdbc:oracle:thin:@US12345.us.company.com:1521:US12345'.

The main Oracle BI Applications Configuration Manager page is displayed.



Use the navigation links at the left-hand side (or use the Task menu) to select the area that you want to manage.

4.5.8.7 How to Launch Oracle BI Applications Configuration Manager in Debug Mode

You launch Oracle BI Applications Configuration Manager when you want to debug an Execution Plan by creating additional debug information that you analyze or send to Oracle Support if required.

To run Oracle BI Applications Configuration Manager in debug mode:

1. Log out of any Oracle BI Applications Configuration Manager sessions.
2. Launch Oracle BI Applications Configuration Manager by appending the loglevel parameter to the URL. For example:

   ```
   <HOST-NAME>:7001/biapps/faces/login.jspx?loglevel=debug
   ```

3. Log in, and run the execution plan.
4. Log out of Oracle BI Applications Configuration Manager.
5. Access the log file in the following location:

   <*Stand-alone WebLogic installation folder*>/user_projects/domains/base_domain

   Note: The default domain name is 'base_domain'. Specify the appropriate domain name. The format of the file name is <WebLogic Session-Id>_<date string>_<time string>_debug.log. The WebLogic session Id is stripped of any special characters to ensure a valid Unix/Windows file name.

4.5.8.8 How to create a Connection For Your Development Environment

This section explains how to create a connection for your development environment in Oracle BI Applications Configuration Manager.

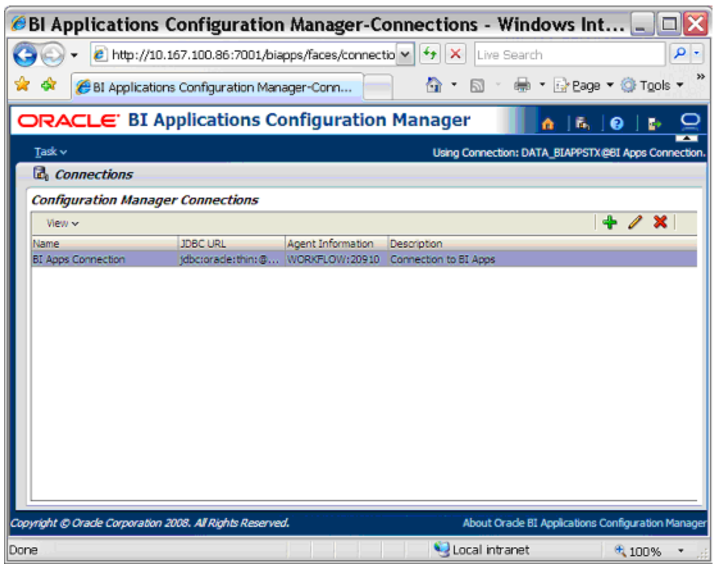
A connection is a stored set of environment details that includes a JDBC connect string to the Oracle Business Analytics Warehouse, an ODI agent name and port, and an ODI context. For example, you might create separate connections for Development, Test, and Production environments.

After you have created a connection, you can log into Oracle BI Applications Configuration Manager by selecting the connection from a drop down list and entering a user name and password, instead of specifying environment details every time they log in.

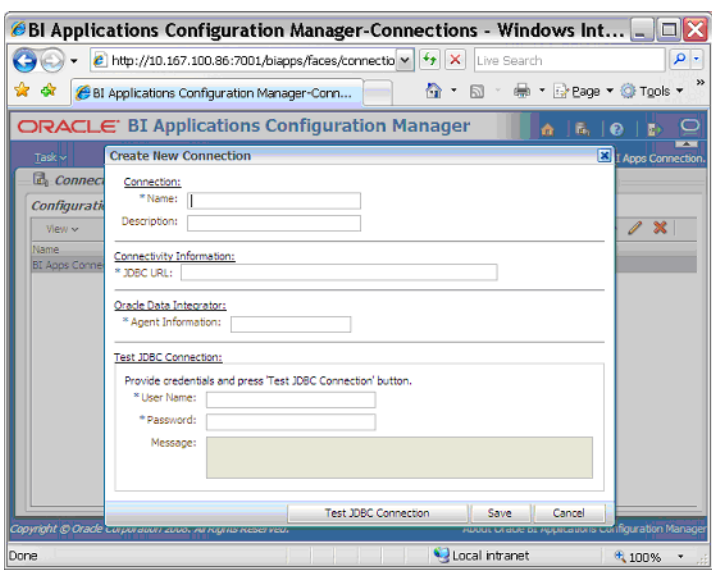
To create a connection in Oracle BI Applications Configuration Manager:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.6, "How to Launch Oracle BI Applications Configuration Manager").

2. Select the **Manage Connections** link to display the Configuration Manager Connections page.



3. Click the New button (+) to display the Create New Connection page.



4. Enter the appropriate connection information, as described in the table below.

Table 4–23   Create New Execution Plan page fields

Field	Description
Name	A user-specified short descriptive name to identify the connection. For example, 'DEV Connection APAC'.
Description	A user-specified short description that is displayed on the Connections page to assist users in selecting the correction connection. For example, 'Connects to Development environment for APAC region'.
JDBC URL	The JDBC URL connection string to the database that hosts the DATA_BIAPPSTX account. Specify in the format jdbc:oracle:thin:@<host>:<port>:<sid>. Replace <host>, <port> and <sid> with the values for the database hosting the ODI Repositories. For example, 'jdbc:oracle:thin:@US12345.us.company.com:1521:US12345'.
Agent Information	Specify the Agent name and port number of the ODI WORKFLOW Agent in the format <**Agent name**>:<**port number**>. For example, 'WORKFLOW:20910'.
User Name	Specify DATA_BIAPPSTX.
Password	Specify DATA_BIAPPSTX.
Message	A read-only field that shows the status of the JDBC Connection test.

5. Click Test JDBC Connection to test the connection details.

   When a valid set of connection details are specified, a 'Successful!' message is displayed.

6. Click Save.

   Now that you have created a Connection, you can use it to log into Oracle BI Applications Configuration Manager using the stored environment details.

   Note: To use the connection that you just created, click Logoff to log out of Oracle BI Applications Configuration Manager, and re-login using the new connection.

4.5.8.9 How to Login to Oracle BI Applications Configuration Manager Using A Connection

You start Oracle BI Applications Configuration Manager to enable you to create and manage E-LT execution plans, set E-LT parameters, and run and monitor E-LT processes.

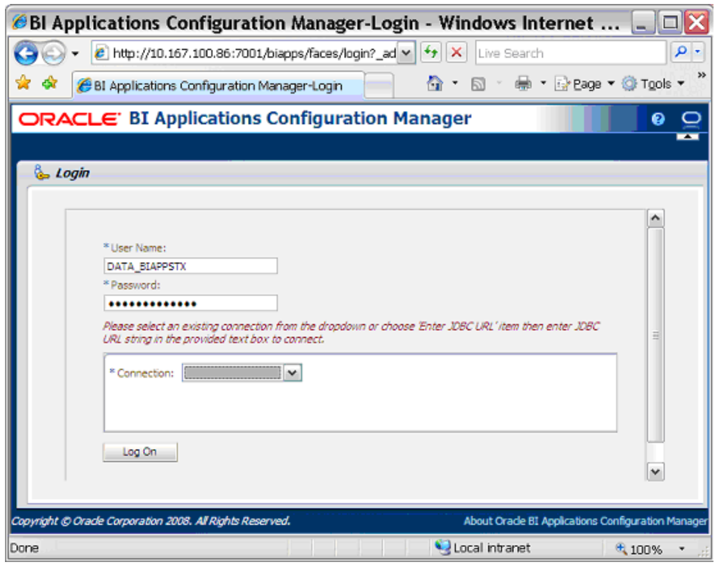
To start Oracle BI Applications Configuration Manager using a connection:

1. Open a supported Web browser.

2. Enter or select the Oracle BI Applications Configuration Manager URL using the fully qualified host name and domain used by your installation to display the Login page.

   Use the URL format: http://<host.domain>:7001/biapps. For example:

   ```
   http://mymachine.us.company.com:7001/biapps
   ```



3. Enter login details as described in the table below, then click Log On.

Table 4–24 Oracle BI Applications Configuration Manager Login page fields

Field Name	**Description**
User Name	Enter 'DATA_BIAPPSTX'.
Password	Enter 'DATA_BIAPPSTX'.
Connection	Select a connection name from the drop down list. For example, you might select 'PROD Connection APAC' for the Production environment for the APAC region.

4. Use the navigation links at the left-hand side (or use the Task menu) to select the area that you want to manage.

4.6 Miscellaneous and Supporting Tasks

This section contains the following topics:

- Section 4.6.1, "How to verify installation and Set up"
- Section 4.6.2, "Using ODI Designer to Manage Your Environment"
- Section 4.6.3, "Using Oracle BI Applications Configuration Manager"
- Section 4.6.4, "About Resetting the Default Passwords, Schema Names, and ODI Connection Details"
- Section 4.6.5, "How to configure and activate Automated Database Statistics Collection"
- Section 4.6.6, "How to configure and activate Automated Delete Handling"
- Section 4.6.7, "About Data Warehouse Loads"
- Section 4.6.8, "How to reset the Oracle Business Analytics Warehouse for Full Load"
- Section 4.6.9, "How to add partitions to Data Warehouse tables"
- Section 4.6.10, "About deploying ODI across multiple environments"

- Section 4.6.11, "How to run an ODI Agent as a Unix background process"
- Section 4.6.12, "How to uninstall ODI Agent Windows Services"
- Section 4.6.13, "How to Resolve Conflicts in ODI Agent Port Numbers"
- Section 4.6.14, "How to activate or deactivate Flow Control in ODI"
- Section 4.6.15, "List of Log Files"

4.6.1 How to verify installation and Set up

To verify and Oracle Business Intelligence Applications environment, do the following: (need to add cross-refs to appropriate tasks).

1. Log into ODI Designer or ODI Topology Manager as user SUPERVISOR.
2. Use ODI Designer to connect to the Work Repository.
3. Use ODI Topology Manager to check the Oracle Data Server Connections.
4. Use ODI Topology Manager to check the Agents.
5. Use ODI Topology Designer to check that the Model objects are available.

4.6.2 Using ODI Designer to Manage Your Environment

This section explains how to use ODI Designer to manage the Oracle Business Intelligence Applications environment.

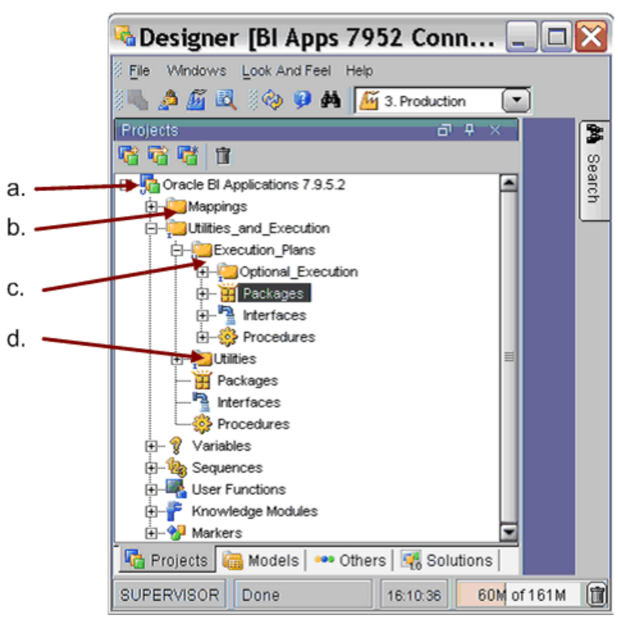
Note: To manage Oracle Business Intelligence Applications environment in ODI, you must log in as user 'SUPERVISOR'.

4.6.2.1 Using the Projects view

This section explains how to use the Projects view in ODI Designer to manage Oracle Business Intelligence Applications.

The figure below shows the main ODI Designer screen for projects:

Figure 4–2 The main Oracle BI Applications project in ODI Designer

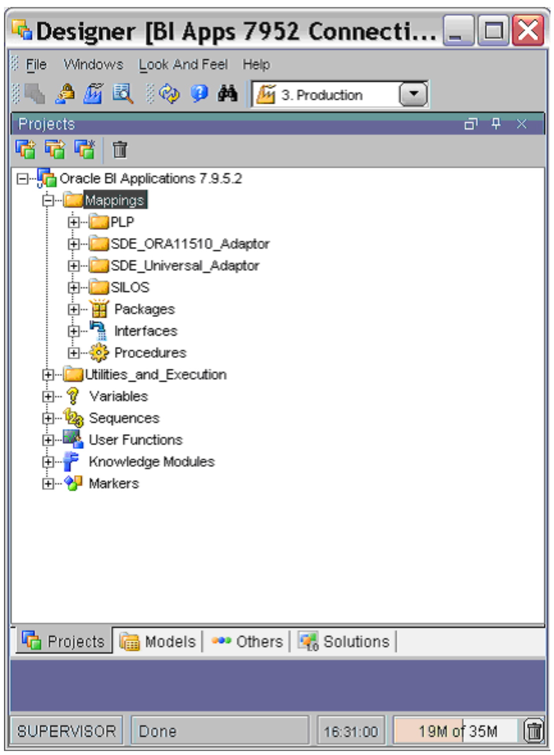


Key

a. The main Oracle BI Applications 7.9.5.2 project (for more information, see Section 4.6.2.1.1, "About the Oracle BI Applications 7.9.5.2 project").
b. The Mappings folder (for more information, see Section 4.6.2.1.2, "About the Mappings folder").
c. The Execution Plans folder (for more information, see Section 4.6.2.1.3, "About the Execution Plans folder").
d. The Utilities folder (for more information, see Section 4.6.2.1.4, "About the Utilities folder").

4.6.2.1.1 About the Oracle BI Applications 7.9.5.2 project The project contains all of the ODI project components required by Oracle Business Intelligence Applications.

4.6.2.1.2 About the Mappings folder The Mappings folder contains the Packages and Interfaces used by Oracle Business Intelligence Applications to perform E-LT. For example, the \SDE_ORA11510_Adaptor folder contains a sub-folder for every Application and Subject Area combination containing Packages and Interfaces for that Subject Area.

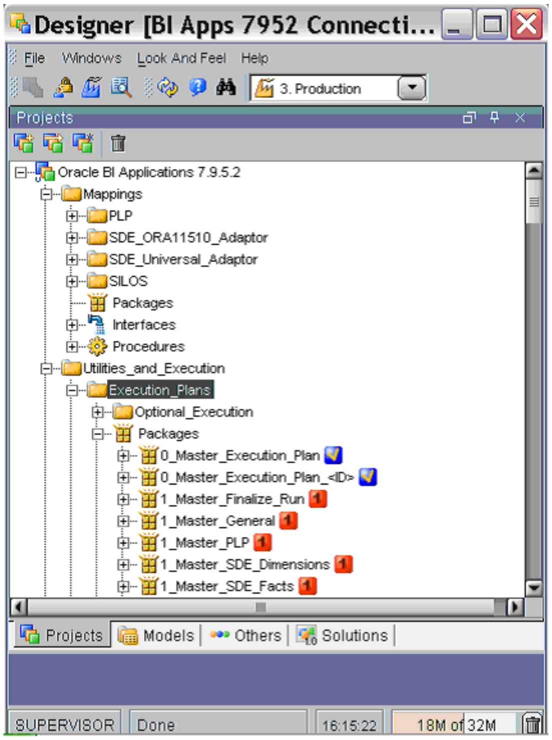


The Mappings folder contains the following main sub-folders:

- PLP - Contains the Post Load Processing (PLP) Mappings.
- SDE_ORA11510_Adaptor - Contains the Source-Dependent Extract (SDE) mappings for Oracle EBS 11.5.10.
- SDE_Universal_Adaptor - Contains the Source-Dependent Extract (SDE) mappings for Universal Adaptor.
- SILOS - Contains the Source Independent Load (SIL) mappings.

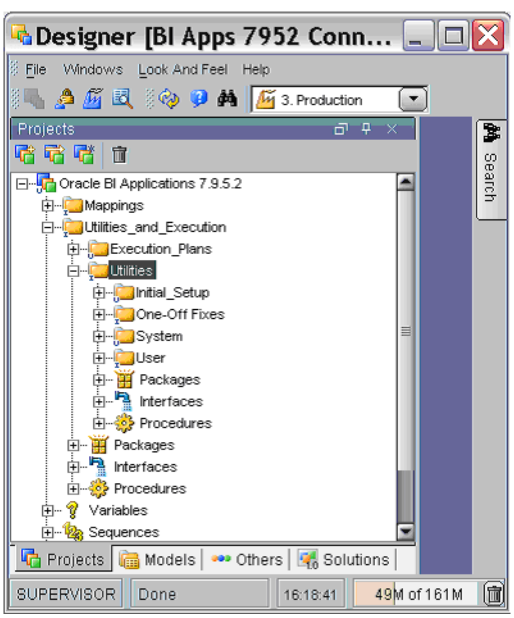
4.6.2.1.3 About the Execution Plans folder

The Execution_Plans folder contains the Master Packages used by Oracle Business Intelligence Applications to run Execution Plans.



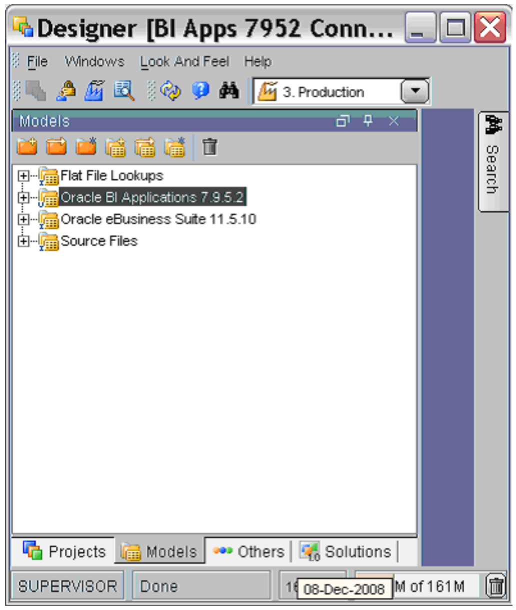
4.6.2.1.4 About the Utilities folder

The Utilities folder contains procedures and packages. that you can use to maintain your Oracle Business Intelligence Applications environment. For example, to collect database statistics, manage indexes, and truncate tables.



4.6.2.2 About the Models view

This section explains how to use the Models view in ODI Designer to manage Oracle Business Intelligence Applications.



Oracle Business Intelligence Applications installs with the following models:

- Flat File Lookups - Contains definitions (metadata) pertaining to the lookup flat file data stores. These objects are mapped to the files in the location $ODI_HOME/oracledi/biapps_odi/odifiles/odidatafiles/lkpfiles.
- Oracle BI Applications 7.9.5.2 - Contains definitions (metadata) of all the Oracle Business Analytics Warehouse objects.
- Oracle eBusiness Suite 11.5.10 - Contains definitions (metadata) of all the EBS 11.5.10 data source objects.
- Source Files - Contains definitions (metadata) of all the source flat-file data stores. These objects are mapped to the files in the location $ODI_HOME/oracledi/biapps_odi/odifiles/odidatafiles/srcfiles.

4.6.3 Using Oracle BI Applications Configuration Manager

This section explains how to use Oracle BI Applications Configuration Manager to manage your Oracle Business Intelligence Applications environment, and contains the following topics:

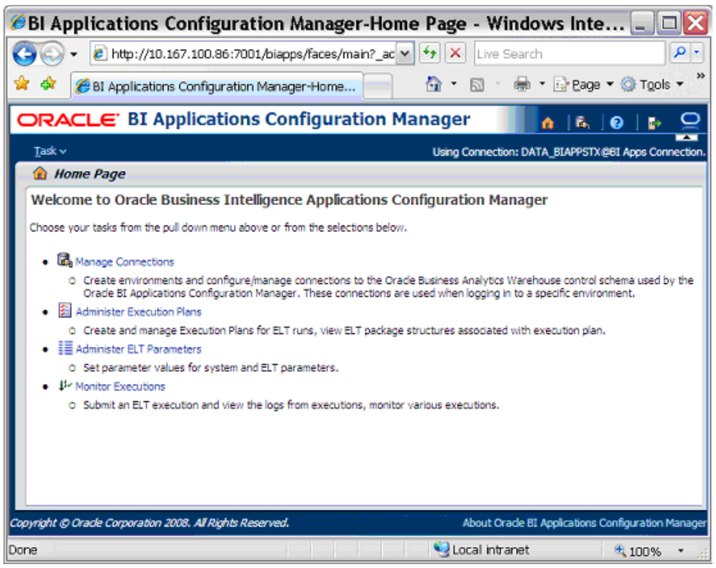
- Section 4.6.3.1, "Overview to Oracle BI Applications Configuration Manager"
- Section 4.6.3.2, "Managing Connections"
- Section 4.6.3.3, "Managing E-LT Parameters"
- Section 4.6.3.4, "Managing Execution Plans"
- Section 4.6.3.5, "Monitoring E-LT Processes"

Note: For a list of log files produced by Oracle BI Applications Configuration Manager, see Section 4.6.15, "List of Log Files".

4.6.3.1 Overview to Oracle BI Applications Configuration Manager

The Oracle BI Applications Configuration Manager is a web-application that enables you to:

- Create and manage execution plans for E-LT executions.
- Set system and E-LT parameters.
- View Master Package structures.
- Run and monitor E-LT executions.



You can also setup connections to different environments, and use a single deployment of Oracle BI Applications Configuration Manager to manage them.

4.6.3.2 Managing Connections

This section explains how to manage connections in Oracle BI Applications Configuration Manager.

A connection is a stored set of environment details that includes a JDBC connect string to the Oracle Business Analytics Warehouse, an ODI agent name and port. For example, you might create separate Production connections for APAC, EMEA, and US regions.

After you have created a connection, you can log into Oracle BI Applications Configuration Manager by selecting the connection from a drop down list and entering a user name and password, instead of specifying JDBC URL details every time you log in.

For more information about creating connections, see Section 4.5.8.8, "How to create a Connection For Your Development Environment".

4.6.3.2.1 Editing and Deleting Connections Use the Configuration Manager Connections page to edit and delete connections. To display the Configuration Manager Connections page, select the Manage Connections link on the main page. Select a connection in the Connections list, then click either Edit or Delete the connection details.

4.6.3.3 Managing E-LT Parameters

This section explains how to manage E-LT Parameters in Oracle BI Applications Configuration Manager.

The Parameters Administration page contains a tab for each category of parameters, as follows:

- Global - these parameters are independent of ELT tasks or apply to all ELT tasks.
- Common - these parameters are for ELT tasks that are common to more than one BI Application module (for example, Human Resources Analytics).
- Application Specific - these parameters are specific to a given BI Application module (for example, Human Resources Analytics).

To navigate to the Parameters Administration page, either click on the Administer Parameters link from the left Navigation pane or click on the link from the Home page or choose the Administer Parameters menu item from the application menu. The Parameters Administration page consists of three tabs that display Global, Common and Application Specific parameters. To set a parameter value, double-click on the parameter record to make the record editable, then enter a value in the **Parameter Value** field.

Note: Part III Configuring Your Analytical Applications describes how to use Oracle BI Applications Configuration Manager to configure specific applications. For more information, refer to the appropriate chapters below for the applications that you have installed:

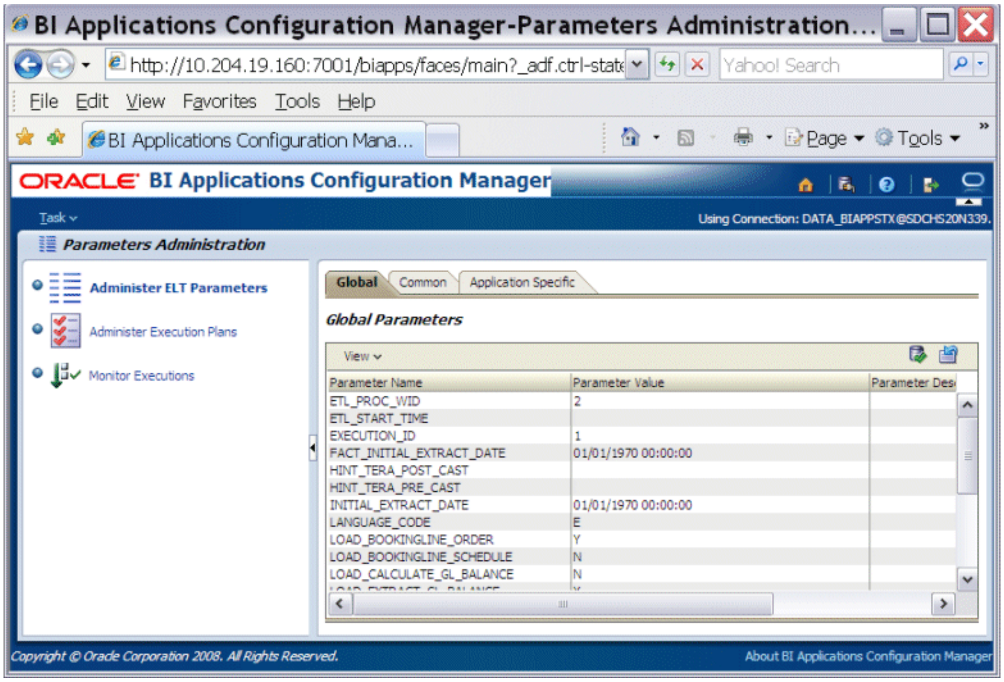
- Chapter 5, "Configuring Common Areas and Dimensions"
- Chapter 6, "Configuring Oracle Procurement and Spend Analytics"
- Chapter 7, "Configuring Oracle Financial Analytics"
- Chapter 8, "Configuring Oracle Supply Chain and Order Management Analytics"
- Chapter 9, "Configuring Oracle Human Resources Analytics"

4.6.3.3.1 How to Set E-LT Parameters In Oracle BI Applications Configuration Manager This section provides generic instructions for modifying E-LT parameters. Part III Configuring Your Analytical Applications describes how to use Oracle BI Applications Configuration Manager to configure specific applications.

Note: Oracle BI Applications Configuration Manager does not perform validation on the parameter values that are entered. For parameters that must be entered in a specific format, follow the formatting instructions noted in the Description column for the parameter. For example, date parameter values must be entered in either YYYYMMDD or MM/DD/YYYY format. Refer to the parameter **Description** information for the parameter to find out the format to use.

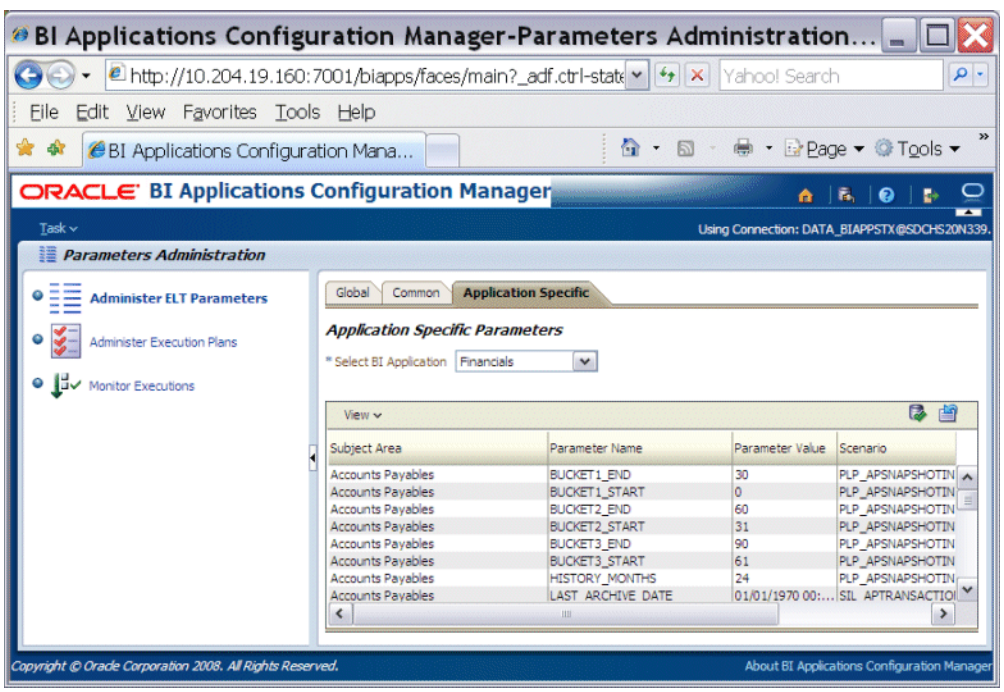
You use Oracle BI Applications Configuration Manager to specify E-LT parameter values that are used to control the E-LT processes for your Applications and Subject Areas.

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link to display the Parameters Administration page.



3. Use the Global, Common, and Application Specific tabs to change the default values of the parameters.

   At the Application Specific tab, the parameters are grouped into Subject Area within Application, to help you locate the parameters that are specific to your Application.



4.6.3.4 Managing Execution Plans

The Execution Plans Administration page enables you to create and maintain your Execution Plans for E-LT runs.

To navigate to the Execution Plans Administration page, either click on the Administer Execution Plans link from the left Navigation pane or click on the link from the Home page or choose the Administer Execution Plans menu item from the application menu.

An execution plan consists of a name, description, an Execution Plan ID, and associated Subject Areas. The Execution Plans Administration page displays a top pane with the list of execution plans. The bottom pane contains two tabs. For the selected execution plan record in the top pane, the Subject Areas tab displays the Subject Areas which are included in the execution plan.

For information about how to create an Execution Plan, see Section 4.7.2.2, "How to Create an Execution Plan In Oracle BI Applications Configuration Manager".

For information about how to run an Execution Plan, see Section 4.7.2.3, "How to Run an Execution Plan in Oracle BI Applications Configuration Manager".

For an example of creating and running an Execution Plan, see Section 4.7.4, "Example of Running A Full Load E-LT in Oracle BI Applications Configuration Manager".

For more information about re-starting an Execution Plan (for example, after diagnosing and fixing an E-LT error), see Section 4.6.3.5.2, "How to Restart Execution Plans After Error Correction".

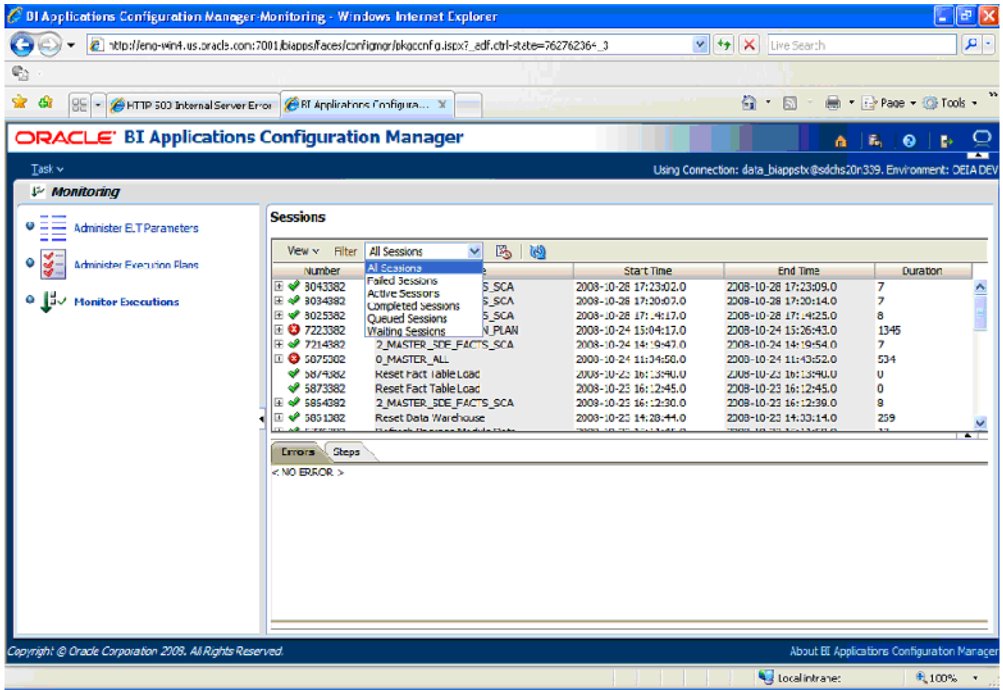
4.6.3.4.1 Editing and Deleting Execution Plans Use the Execution Plans page to edit and delete Execution Plans. To display the Execution Plans page, select the Administer Execution Plans link on the main page.

4.6.3.5 Monitoring E-LT Processes

This section explains how to monitor E-LT processes in Oracle BI Applications Configuration Manager, and contains the following topics:

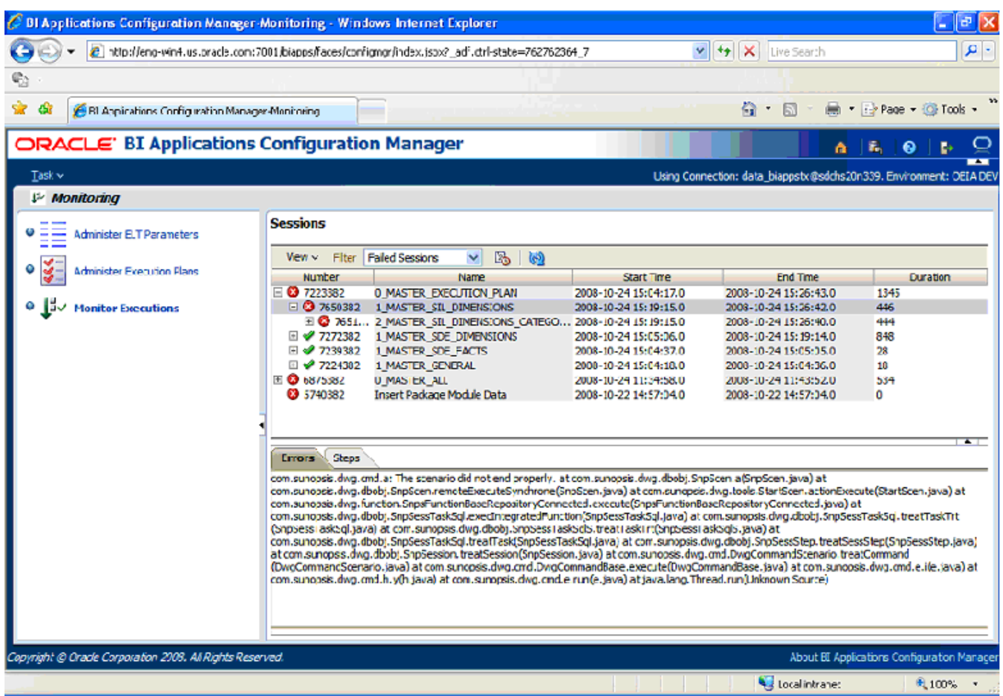
- Section 4.6.3.5.1, "How to Monitor E-LT Processes"
- Section 4.6.3.5.2, "How to Restart Execution Plans After Error Correction"

4.6.3.5.1 How to Monitor E-LT Processes You can use Oracle BI Applications Configuration Manager to monitor E-LT processes. You can generate logs that you can use to diagnose issues that you might have. You monitor E-LT processes using the Sessions page, which you display by selecting the Monitor Executions link on the main page.

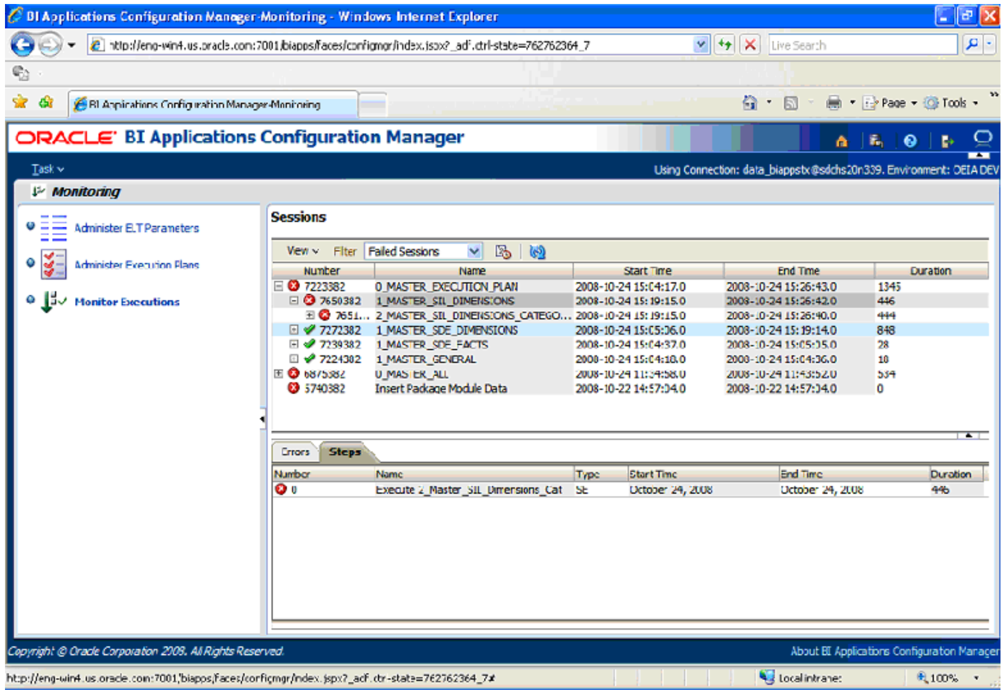


The Sessions displayed are identified by Execution Plan ID, and relate to the 0_ Master_Execution_Plan package (or copy of) that you set up as the Master Package (for more information, see Section 4.7.1, "How to set up Master Packages to run an Execution Plan"). The Sessions page lists sessions initiated in both ODI Designer and BI Applications.

The Errors tab in the bottom pane displays errors for the session that is highlighted in the top pane. The figure below shows the Errors tab for a failed session.



The Steps tab in the bottom pane displays more information about the session and task that is highlighted in the top pane.



4.6.3.5.2 How to Restart Execution Plans After Error Correction

If an Execution Plan fails, after you diagnose and fix the underlying error you need to restart the Execution Plan, as described in this section.

To restart an Execution Plan after error correction:

1. Start Oracle BI Applications Configuration Manager and connect to the required environment (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer Execution Plans** link.
3. In the upper pane, select the Execution Plan that you want to restart.
4. Display the 'Package Structure' tab in the lower pane.
5. Set the Filter to 'Execution Structure'.
6. Click the **Execute** button on the right hand side of the **Filter** drop down list.

4.6.4 About Resetting the Default Passwords, Schema Names, and ODI Connection Details

Oracle recommends that you use the out-of-the-box schema names, tablespace names, object names, and passwords. If you do use different names, follow the procedures in this section to keep the Oracle Business Intelligence Applications components synchronized.

This section explains how to reset database passwords, schema names, and connection details for the components in Oracle Business Intelligence Applications, and what related components you need to change to keep your components synchronized. For example, if you want to use a different database schema name to the recommended 'ODI_REP_7952', you also need to modify ORACLE_WORK_REP schema in ODI Topology Manager.

This section contains the following topics:

- Section 4.6.4.1, "How to Change the Default Database Passwords"
- Section 4.6.4.2, "How to Change the ODI Topology Connection Details"
- Section 4.6.4.3, "How to Change the Encoded Passwords in odiparams.bat"
- Section 4.6.4.4, "How to Change the Password for the ODI SUPERVISOR User"
- Section 4.6.4.5, "Using Different Schema and Tablespace Names"
- Section 4.6.4.6, "Using A Different ODI Work Repository Name"

4.6.4.1 How to Change the Default Database Passwords

This section explains how to change the default database passwords for the databases that you set up in Section 4.3.3.1, "Creating the Required Databases and Tablespaces".

Use an appropriate SQL client to change the database password.

For example, you might connect to Oracle SQL Developer or SQL*Plus and execute the following command:

```
ALTER USER ODI_REP_7952 IDENTIFIED BY <New Password> REPLACE <Old Password>
```

Notes

- If you change the ODI_REP_7952 database password, you also need to modify the connection details for ORACLE_WORK_REP work repository as described in Section 4.6.4.2, "How to Change the ODI Topology Connection Details".
- If you change the DATA_BIAPPS database password, you also need to modify the connection details for ORACLE_BI_APPLICATIONS as described in Section 4.6.4.3, "How to Change the Encoded Passwords in odiparams.bat".

4.6.4.2 How to Change the ODI Topology Connection Details

This section explains how to change the default ODI Topology Connection details. The table below shows the default schemas and what you need to do if you change the connection details.

Table 4–25 Database Schemas and Related Physical Data Servers

Database Schema Name	Related Physical Data Server	Action Required to Synchronize
ODI_REP_7952	ORACLE_WORK_REP	If you change the ODI_REP_7952 schema password, you need to modify the Oracle BI Applications Configuration Manager setup.
TEMP_BIAPPS	Not applicable.	Not applicable.
DATA_BIAPPS	ORACLE_BI_ APPLICATIONS	If you change the DATA_BIAPPS schema password, you need to modify the Oracle BI Applications Configuration Manager setup.
DATA_BICONSOLE	Not applicable.	If you change the DATA_BICONSOLE schema password, you need to modify the Oracle BI Applications Configuration Manager setup.

To change the Work Repository connection details:

1. In ODI Topology Manager, display the Repositories tab.

2. Expand the Work Repositories node.
3. Edit the repository WORKREP_7952, to display the Work Repository: <*Repository Name*> dialog.
4. Click Connection to display the Data Server: <*Repository Name*> dialog.
5. Use the **Password** field to modify the password, then click Test to verify the new password.
6. Save the details.

4.6.4.3 How to Change the Encoded Passwords in odiparams.bat

This section explains how to change the encoded passwords that are set in the ODI configuration file odiparams.bat. For example, the odiparams.bat file located in the $ODI_HOME\oracledi\bin directory might contain the following parameters:

```
set ODI_SECU_USER=ODI_REP_7952
set ODI_SECU_ENCODED_PASS=dpfHHIqHwW.v9VGofVVd6eG0x
set ODI_USER=SUPERVISOR
set ODI_ENCODED_PASS=d,ypFC5Tzt5plXN82JwXASAUp
```

You might want to change the password for ODI_REP_7952 user or the SUPERVISOR user.

To encode a password:

1. On the ODI machine, open a command prompt, and use the CD command to change to the $ODI_HOME\oracledi\bin directory.
2. Enter the following command:

   ```
   agent ENCODE <password>
   ```

   Where <password> is either ODI_REP_7952 or SUPERVISOR.

   The 'agent ENCODE' command returns an encoded password for ODI_REP_7952 or SUPERVISOR and displays it on screen.
3. Note down or copy into memory the encoded password returned by the 'agent ENCODE' command.
4. Open the file $ODI_HOME\oracledi\bin\odiparams.bat in a text editor.
5. If you obtained a new encoded password for the ODI_REP_7952 user, type in or paste in the encoded password that you obtained in step 3 as the value of the set 'ODI_SECU_ENCODED_PASS=' parameter.
6. If you obtained a new encoded password for the SUPERVISOR user, type in or paste in the encoded password that you obtained in step 3 as the value of the set 'set ODI_ENCODED_PASS=' parameter.
7. Save the odiparams.bat file.

4.6.4.4 How to Change the Password for the ODI SUPERVISOR User

This section explains how to change the ODI SUPERVISOR password.

To change the passwords for the ODI SUPERVISOR user:

1. Log into ODI Designer or ODI Topology Manager.
2. Choose File, then Change Password to display the 'Password change' dialog.

3. Use the 'Password change' dialog to specify the current password, and the new password.
4. Save the details.

   You now need to modify the ODI Connections to use the new SUPERVISOR password.

4.6.4.5 Using Different Schema and Tablespace Names

This section explains what you have to change if you use different schema and tablespace names to the default names.

4.6.4.5.1 Changing the Oracle Business Analytics Warehouse Schema Names To use different Oracle Business Analytics Warehouse Schema names for DATA_BIAPPS or TEMP_BIAPPS, do the following:

1. Create new database account names for DATA_BIAPPS or TEMP_BIAPPS.

   For more information about creating database accounts, see Section 4.3.3.1, "Creating the Required Databases and Tablespaces".
2. In ODI Topology, replace the database account names for DATA_BIAPPS or TEMP_BIAPPS to the new database account names that you created in step 1.
3. Modify the biappstx_setup.sql file as follows:
   a. On the ODI machine, open the biappstx_setup.sql in a text editor or SQL tool.

      The biappstx_setup.sql file is located in the $ODI_HOME\oracledi\biapps_odi\dbfiles\ directory. For example, D:\OraHome_1\oracledi\biapps_odi\dbfiles\.
   b. Modify the value of the 'DEFINE L_OBIA_SCHEMA = 'statement.

      For example, you might change this statement to `DEFINE L_OBIA_SCHEMA = DATA_TESTSCHEMA`.
   c. Save the file.

4.6.4.5.2 Changing the ODI Repository Schema Name To use a different ODI Repository schema name, do the following:

1. Create new database account names for ODI_REP_7952.

   For more information about creating database accounts, see Section 4.3.3.1, "Creating the Required Databases and Tablespaces".
2. Modify the odiparams.bat or odiparams.sh file and change the following parameters:

   ```
   ODI_SECU_USER
   ODI_SECU_ENCODED_PASS
   ```

   For information about obtaining encoded passwords, see Section 4.6.4.3, "How to Change the Encoded Passwords in odiparams.bat".
3. Edit the ODI Master Repository to use the new database account name that you created in step 1.
4. In ODI Topology, replace the database account name for ODI_REP_7952 to the new database account name that you created in step 1.
5. Modify the biappstx_setup.sql file as follows:

a. On the ODI machine, open the biappstx_setup.sql in a text editor or SQL tool.

   The biappstx_setup.sql file is located in the $ODI_HOME\oracledi\biapps_odi\dbfiles\ directory. For example, D:\OraHome_1\oracledi\biapps_odi\dbfiles\.

b. Modify the value of the 'DEFINE L_ODI_SCHEMA = 'statement'.

   For example, you might change this statement to `DEFINE L_ODI_SCHEMA = ODI_REP_1`.

c. Save the file.

4.6.4.5.3 Changing the Schema Name for Oracle BI Applications Configuration Manager To use a different Schema name, do the following:

1. Create a new database account name for DATA_BIAPPSTX.

   For example, you might create a new database account name DATA_CONFTX that has the same privileges as DATA_BIAPPSTX.

   For more information about creating database accounts, see Section 4.3.3.1, "Creating the Required Databases and Tablespaces".

2. In Oracle BI Applications Configuration Manager, replace the database account name for DATA_BIAPPSTX to the new database account name that you created in step 1.

3. Modify the biappstx_setup.sql file as follows:

   a. On the ODI machine, open the biappstx_setup.sql in a text editor or SQL tool.

      The biappstx_setup.sql file is located in the $ODI_HOME\oracledi\biapps_odi\dbfiles\ directory. For example, D:\OraHome_1\oracledi\biapps_odi\dbfiles\.

   b. Modify the value of the 'DEFINE L_TX_SCHEMA = 'statement.

      For example, you might change this statement to `DEFINE L_TX_SCHEMA = DATA_CONFTX`.

   c. Save the file.

4.6.4.5.4 Changing the Schema Name for the Index Tablespace To use a different index tablespace schema name, do the following:

1. Create a new database account name for BIAPPS_INDEX.

   For example, you might create a new database account name BIAPPS_INDEX_1 that has the same privileges as BIAPPS_INDEX.

   For more information about creating database accounts, see Section 4.3.3.1, "Creating the Required Databases and Tablespaces".

2. In ODI Designer, modify the Oracle BI Applications 7.9.5.2 model as follows:

   a. Log into ODI Designer and display the Models view.

   b. Double click on the Oracle BI Applications 7.9.5.2 model to display the Model: <*Name*> dialog.

   c. Display the FlexFields tab.

   d. Replace the value of the INDEX_TABLESPACE field with the new database account name that you created in step 1.

3. Save the details, and re-start the ODI Agents if necessary.

4.6.4.6 Using A Different ODI Work Repository Name

Note: Before you modify the ODI Work Repository Name, make sure that all ODI processes have stopped.

To use a different ODI Work Repository name, do the following:

1. Stop the ODI Agents INTERFACE and WORKFLOW.
2. In ODI Topology Manager, display the Repositories tab.
3. Expand the Work Repositories node and double-click on WORKREP_7952 to display the Work Repository: <*Name*> dialog.
4. Use the **Name** field to change the name of the ODI Work Repository, and save the details.
5. Log out of ODI Topology Manager.
6. Start ODI Designer to display the Oracle Data Integrator Login dialog.
7. Use the **Login Name** list to select the connection details for the environment that you are working in.
8. Click the Edit button to display the Work Repository Connection dialog.
9. Use the **Repository Name** field to change the name of the ODI Work Repository, and save the details.
10. In a text editor, edit the odiparams.bat or odiparams.sh file and modify the value of the 'set ODI_SECU_WORK_REP=' command.

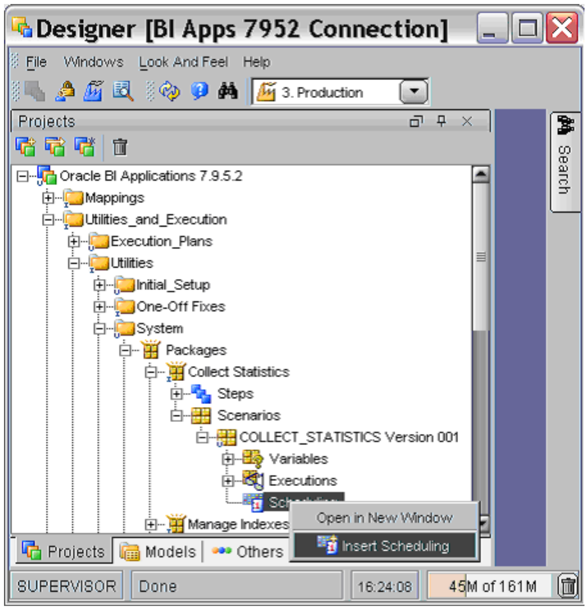
    For example, you might change the command to set `ODI_SECU_WORK_REP=WORKREP_EMEA`.
11. Re-start the ODI Agents WORKFLOW and INTERFACE.

4.6.5 How to configure and activate Automated Database Statistics Collection

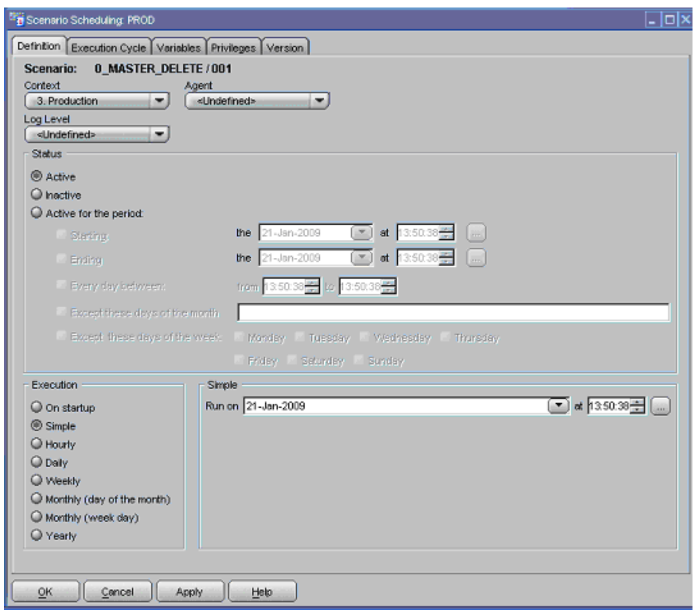
Oracle Business Intelligence Applications is installed with a schedule that you can use in ODI Designer to automatically collect database statistics for analysis. This section explains how to activate this package (named 'Collect Statistics') and optionally change its schedule (by default, it is executed once per week at 12.00 AM on Saturday).

To activate and configure the 'Collect Statistics' Package:

1. In ODI Designer, display the Projects view.
2. Open the project Oracle BI Applications 7.9.5.2.
3. Navigate to \Utilities_and_Execution\Utilities\System\Packages\Collect Statistics\Scenarios\COLLECT_STATISTICS Version 001\Scheduling.



4. Right-click on Scheduling and select Insert Scheduling to display the Scenario Scheduling: <*Name*> dialog.



5. Use the fields on the Scenario Scheduling: <*Name*> dialog to modify the schedule, as follows:

   - Use the Status area to activate or deactivate the package.

     For example, select the **Active** radio button to activate the package.

   - Use the **Context** field to select the 'Production' context.

- Use the **Agent** field to select the 'INTERFACE' agent.
- Use the other fields to specify the time and frequency of the refresh.

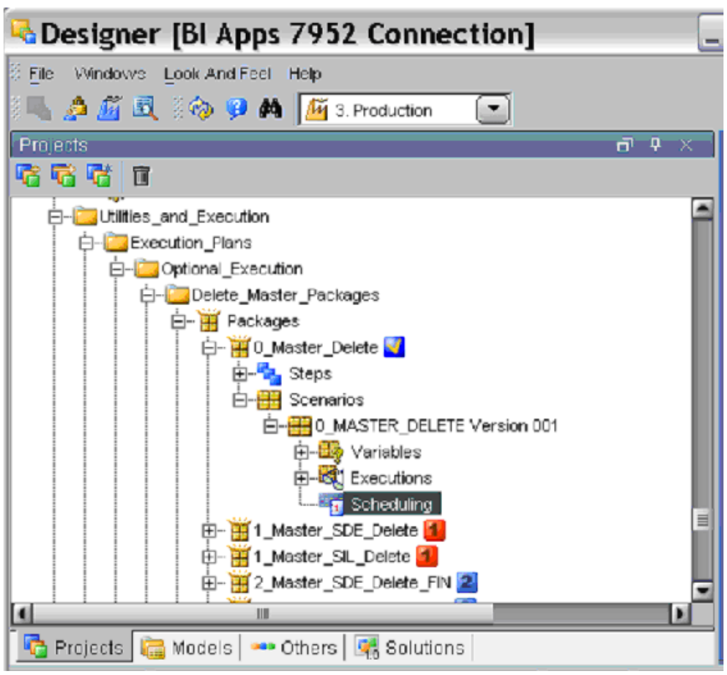
6. Save the details.

4.6.6 How to configure and activate Automated Delete Handling

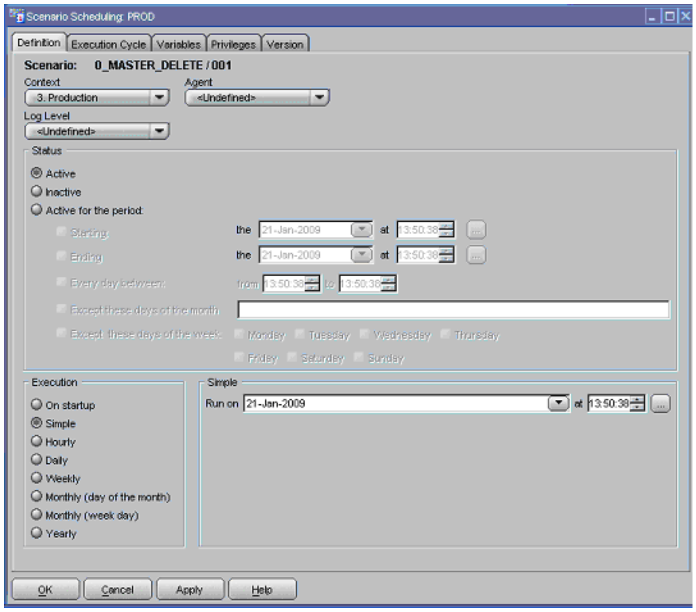
Oracle Business Intelligence Applications is installed with a schedule that you can use in ODI Designer to automatically handle deletes made in your source system and propagate them to the Oracle Business Analytics Warehouse. This section explains how to activate this scenario and specify its schedule.

To activate and configure the 'Collect Statistics' Package:

1. In ODI Designer, display the Projects view.
2. Open the project Oracle BI Applications 7.9.5.2.
3. Navigate to \Utilities_and_Execution\Execution_Plans\Optional_Execution\Delete_Master_Packages\Packages\0_Master_Delete\Scenarios\0_MASTER_DELETE Version 001\Scheduling.



4. Right click on the Scheduling node and choose Insert Scheduling to display the Scenario Scheduling: <*Name*> dialog.



5. Use the fields on the Scenario Scheduling: <*Name*> dialog to modify the schedule, as follows:

   - Use the Status area to activate or deactivate the package.

     For example, select the **Active** radio button to activate the package.

   - Use the **Context** field to select the 'Production' context.
   - Use the **Agent** field to select the 'INTERFACE' agent.
   - Use the other fields to specify the time and frequency of the refresh.

6. Save the details.

4.6.7 About Data Warehouse Loads

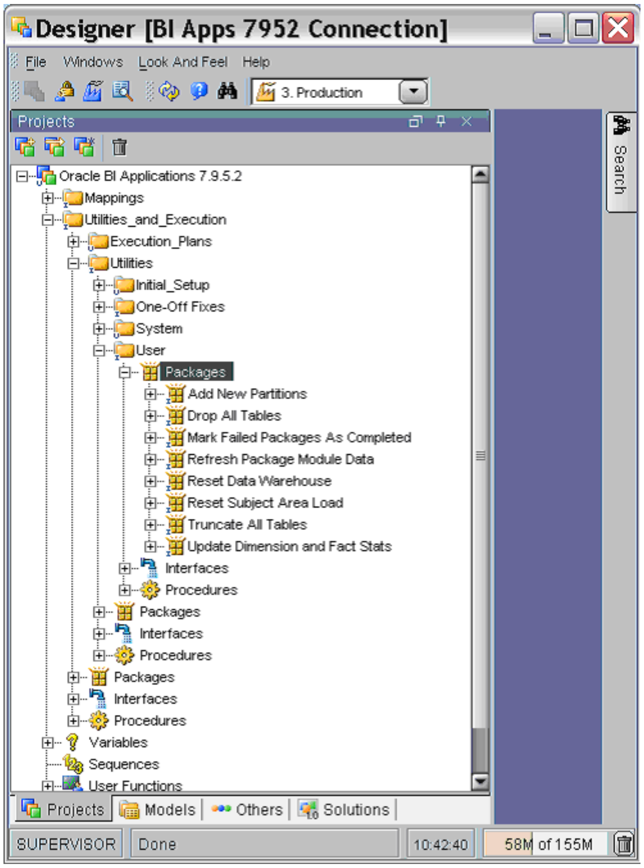
This section explains how to use loads in ODI to manage your Oracle Business Intelligence Applications environment.

The Master Packages are pre-programmed to handle incremental load logic implicitly. All the runs initiated after the initial full load is successfully done are incremental by default.

4.6.8 How to reset the Oracle Business Analytics Warehouse for Full Load

To reset the Oracle Business Analytics Warehouse for full load:

1. In ODI Designer, display the Projects view.
2. Open the project Oracle BI Applications 7.9.5.2.
3. Navigate to Utilities_and_Execution\Utilities\User\Packages\.



4. Execute the 'Reset Data Warehouse' package.

4.6.9 How to add partitions to Data Warehouse tables

This section explains how to use Oracle Data Integrator to manage partitioning and indexing on data warehouse tables, and contains the following topics:

- Section 4.6.9.1, "About supported partitioning in this release"
- Section 4.6.9.2, "An example of partitioning a table"
- Section 4.6.9.3, "How to generate a DDL for a modified table"
- Section 4.6.9.6, "How to add a partition to a table that is already partitioned"

Note: For more information about partitioning in Oracle Business Intelligence Applications, see *Oracle Database Data Warehousing Guide*.

4.6.9.1 About supported partitioning in this release

When you set up partitioning, note the following:

- This release supports only partitions based on DATE based key columns.
- This release supports only range partitioning.
- Each partitions must represent a month of data.
- Partition Index management and Statistics collection can be fine-tuned to optimally handle incremental date range windows.

4.6.9.1.1 About using NUM_PARTITIONS_INCR FlexFields The FlexField NUM_PARTITIONS_INCR is relevant only for the tables with partitioning enabled. This FlexField carries a numeric value representing the number of months affected by the Index Management and Statistics Collection routines.

The following actions are automatically performed on most current partitions determined by NUM_PARTITIONS_INCR:

- Disable Partition Index before Incremental loads.
- Statistics Collection after Incremental Loads
- Enable Partition Indexes after Incremental Loads.
- The default value for the NUM_PARTITIONS_INCR FlexField is 3.

4.6.9.2 An example of partitioning a table

This section uses an example to show how to partition a data warehouse table.

Before starting this procedure, the following information is required:

- The name of the Data Warehouse Table being partitioned.
- The Partitioning Key Column (only date key columns are supported).
- The Start Key Value.
- The Maximum Key Value.
- The number of Partitions optimally affected by Index Management/Statistics Collection routines.

Example of partitioning the W_REV_F fact table:

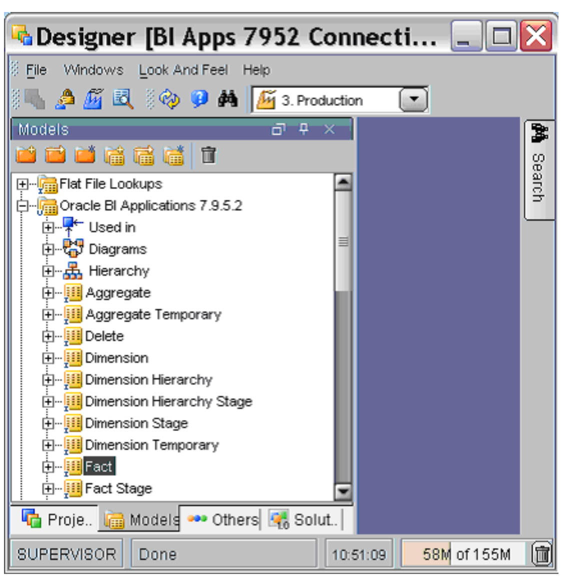
1. Use a SQL editor such as SQL*PLUS or SQL Developer to make a back-up of the fact W_REV_F fact table.

   This back-up version will enable you to repopulate the table in the later step Section 4.6.9.4, "About Repopulating A Partitioned Table").

2. In ODI Designer, display the Models view.

   Note: You must be logged in as the SUPERVISOR user.

3. Open the model Oracle BI Applications 7.9.5.2.



4. Expand the Fact node and locate the W_REV_F fact table.
5. Double-click the W_REVN_F table to display the DataStore: <*Name*> dialog.

   Note: If the Object Locking dialog is displayed, click Yes to unlock this object.
6. Display the FlexFields tab, and enter the appropriate information, as described in the table below.

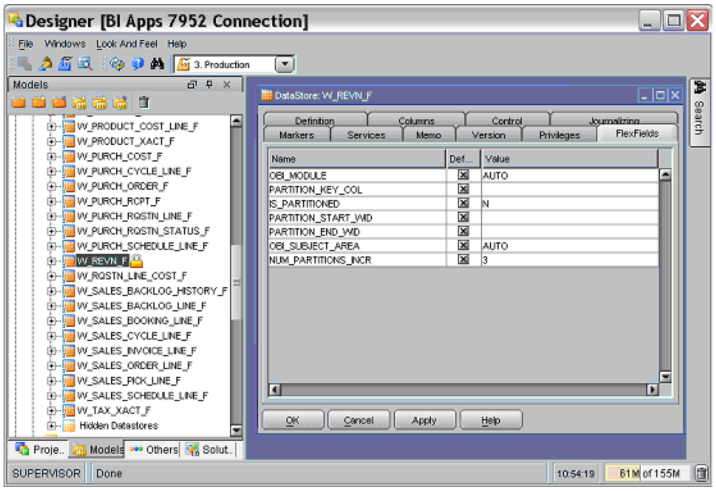


Table 4–26 DataStore: <Name> FlexField tab fields

Field	Value specified for this example
OBI_MODULE	AUTO
PARTITION_KEY_COL	CLOSE_DT_WID
IS_PARTITIONED	Y
PARTITION_START_WID	200601
PARTITION_END_WID	201012
OBI_MODULE	AUTO
NUM_PARITIONS_INCR	3

7. Save the details.
8. Re-generate a DDL for this table by following the steps in Section 4.6.9.3, "How to generate a DDL for a modified table".

4.6.9.3 How to generate a DDL for a modified table

If you have partitioned a table (for example, the W_REVN_F table (as described in Section 4.6.9.2, "An example of partitioning a table")), follow this task to generate a DDL.

To generate a DDL for a modified table:

1. Using a SQL client tool (for example, Oracle SQL*Plus or Oracle SQL Developer), connect to your data warehouse as user DATA_BIAPPS.
2. To determine whether the W_REVN_F table is populated, enter the following SQL command:

   ```
   SELECT COUNT(*) FROM W_REVN_F;
   ```

3. If the table is populated, make a copy of the table by entering the following SQL command:

```
CREATE TABLE <Temp Table Name> AS SELECT* FROM W_REVN_F;
```

4. Delete (or 'drop') the W_REVN_F table by entering the following SQL command:

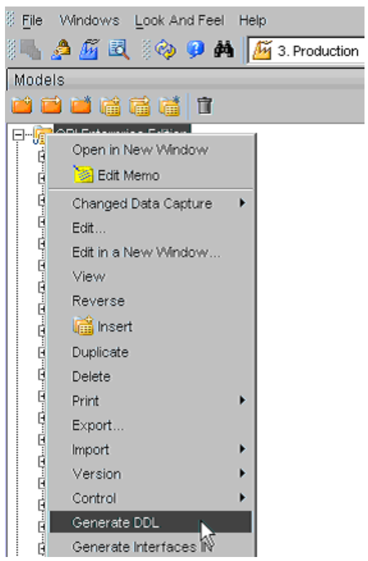
```
DROP TABLE W_REVN_F;
```

5. In ODI Designer, display the Models view.

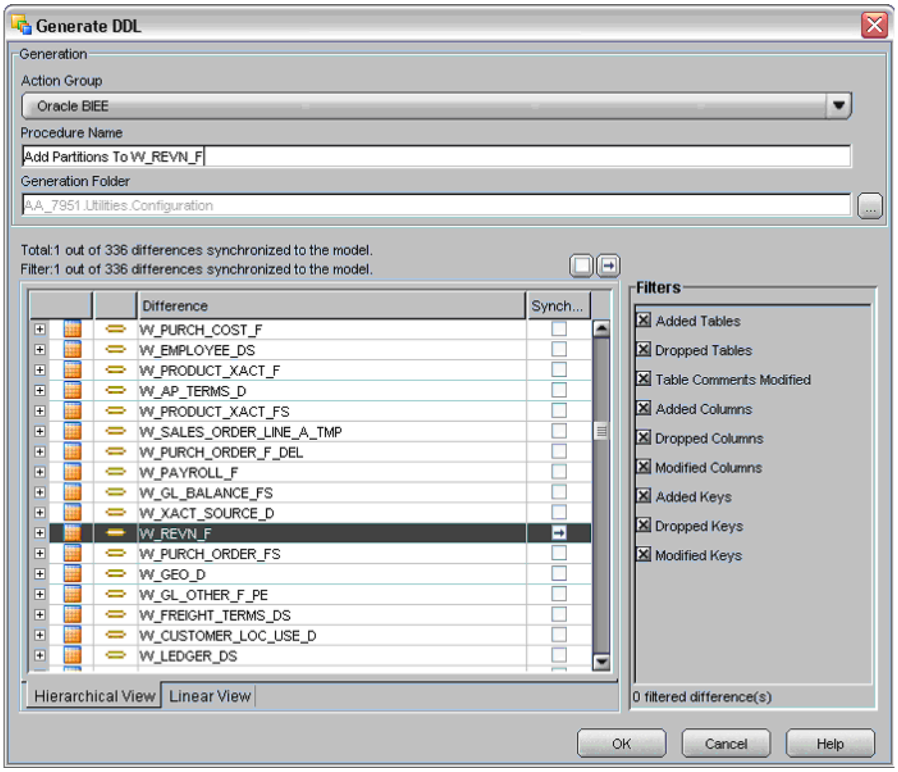
   Note: You must be logged in as the SUPERVISOR user.

6. Open the model ORACLE BI APPLICATIONS 7.9.5.2.

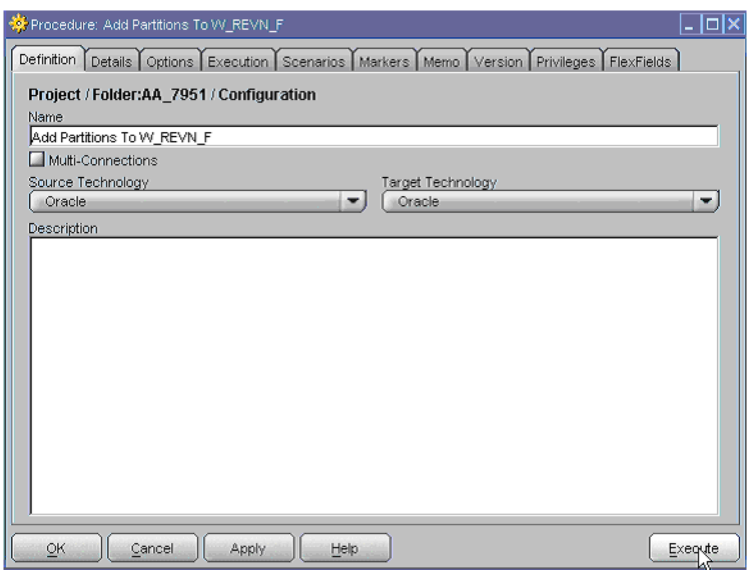
7. Right-click on Oracle BI Applications 7.9.5.2 and select Generate DDL.



8. At the Generate DDL for Oracle Data Integrator Model dialog, click No to display the Generate DDL dialog.



9. Select W_REVN_F in the list, and select the adjacent check box in the **Synchronization** column to display the Procedure: <*Name*> dialog.



10. Select Oracle from the **Source Technology** drop down list.

11. Select Oracle from the **Target Technology** drop down list.

12. Click Execute to display a Confirmation dialog, and click Yes to display the Execution dialog.

13. Click OK to start the process.

You can now use ODI Operator to monitor the process and make sure that it executes successfully (to start ODI Operator, click the ODI Operator icon on the ODI Designer toolbar).

4.6.9.4 About Repopulating A Partitioned Table

After you have partitioned a table (for an example using the fact table, see Section 4.6.9.2, "An example of partitioning a table") you need to re-populate the table.

Using a back-up copy of the table that you partitioned, repopulate the partitioned table. Use a SQL editor such as SQL*PLUS or SQL Developer to execute the 'INSERT INTO… SELECT FROM' SQL statement. Make sure that you execute a 'COMMIT' before closing the SQL session.

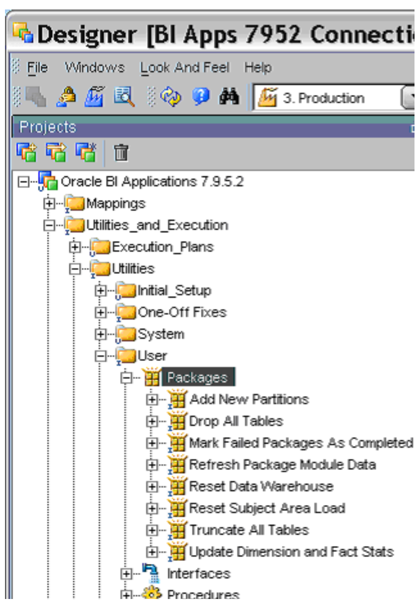
4.6.9.5 About Creating Indexes and Regenerating Statistics

Oracle recommends that you manually create Partition indexes. Refer to Oracle Database documentation for details on how to create indexes on table partitions.

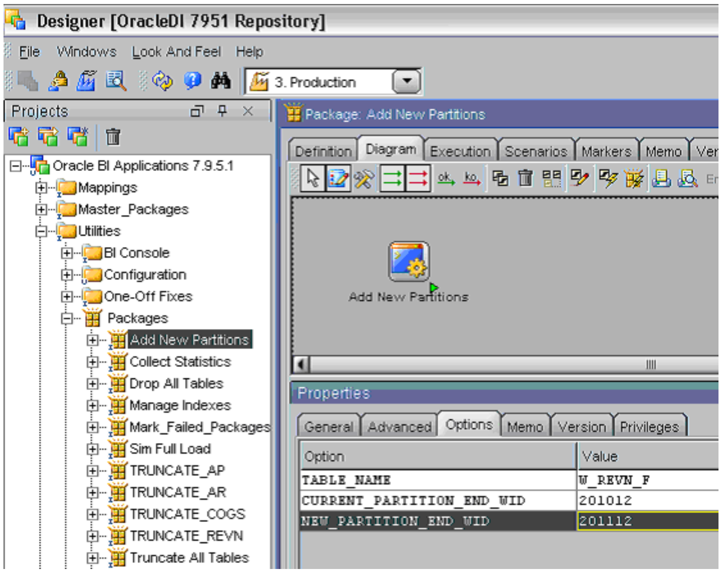
4.6.9.6 How to add a partition to a table that is already partitioned

To add a partition to a table that is already partitioned:

1. In ODI Designer, display the Projects view.
2. Open the project Oracle BI Applications 7.9.5.2.
3. Navigate to Utilities_and_Execution\Utilities\User\Packages\.



4. Edit the 'Add New Partition' package to display the Package: Add New Partitions dialog.
5. Display the Diagram tab, and click on the Add New Partition object in the Diagram pane.
6. Display the Options tab in the Properties panel below.



7. Specify the following values:
   - TABLE NAME - W_REVN_F
   - CURRENT_PARTITION_END_WID - 201012
   - NEW_PARTITION_END_WID - 201112
8. Click Apply, then Execute to display the Execution dialog.
9. At the Execution dialog, select INTERFACE from the Agent drop down list, then OK.

   You can now use ODI Operator to monitor the process and make sure that it executes successfully (to start ODI Operator, click the ODI Operator icon on the ODI Designer toolbar).

4.6.10 About deploying ODI across multiple environments

In ODI deployments, you typically maintain different environments (known as Contexts) for Production (PROD), Development (DEV) and Quality Assurance (QA) activities. Oracle recommends that you install multiple ODI Repositories and create Topologies to support multiple environments.

To avoid collision between different environments, it is also recommended that you install ODI separately on different servers for each of the environments.

Alternatively, multiple contexts can be created on a single ODI Repository corresponding to each of the additional environments, as follows:

- Create a Context for each of the additional environment i.e. DEV, QA, PROD.
- Create a database user for the ODI Staging schema.
- Create new data servers corresponding to the data source(s), Staging schema and Data Warehouse target.
- For each data server, create corresponding physical schemas.
- Create logical schemas and associate them with physical schemas in the contexts.
- Associate a DATASOURCE_NUM_ID with the new contexts.

- Create the physical agents for each agent running on a machine (as a listener, or in scheduler mode).
- Create logical agents and associate them with physical agents in the contexts.
- Create Physical and Logical servers corresponding to each of the data sources and data warehouse targets.

Note: For detailed information about ODI Contexts, refer to the Oracle Data Integrator Documentation Library.

4.6.11 How to run an ODI Agent as a Unix background process

To run an ODI Agent as a Unix background process:

1. Open a command window and change directory to the $ODI_ HOME/oracledi/bin directory.
2. Execute the following commands one at a time:

```
./agentscheduler.sh -NAME=WORKFLOW -PORT=20910 &
./agent.sh -NAME=INTERFACE -PORT=20911 &
```

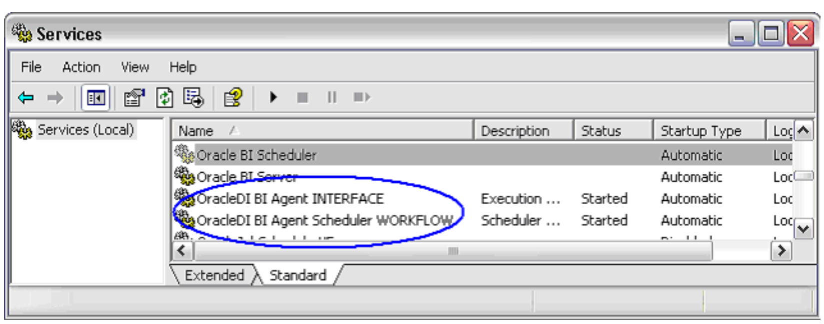
4.6.12 How to uninstall ODI Agent Windows Services

If you have a problem with the ODI Agent processes, you might need to uninstall the services so that you can reinstall them.

To uninstall the ODI Agent services:

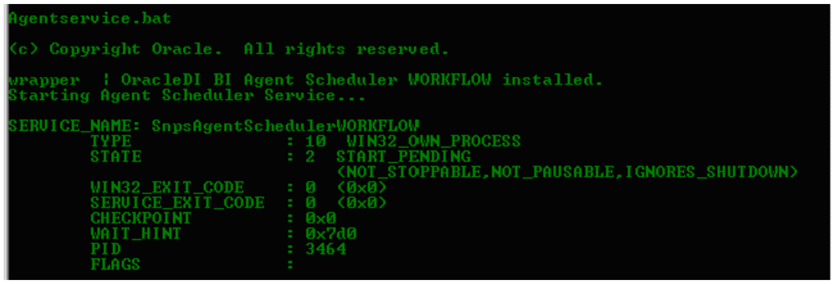
1. Display the Windows Services dialog.

   For example, run 'services.msc' from the Windows > Run dialog.
2. Stop the 'OracleDI BI Agent WORKFLOW' service and the 'OracleDI BI Agent INTERFACE' service.



3. Open a command window and change directory to the $ODI_ HOME\oracledi\bin directory.
4. Execute the following commands one at a time:

```
agentservice -r -s WORKFLOW 20910
agentservice -r -a INTERFACE 20911
```



If required, you can now reinstall the services (for more information about installing these services, see Section 4.5.3, "How to start the ODI Agents").

4.6.13 How to Resolve Conflicts in ODI Agent Port Numbers

This section explains how to modify your Oracle Business Intelligence Applications deployment to resolve conflicting port numbers for ODI Agents.

To resolve conflicts in ODI Agent port numbers:

1. Modify the port number of the INTERFACE and WORKFLOW agents, as described in sections Section 4.5.5.2.1, "How to set up the INTERFACE Agent" and Section 4.5.5.2.2, "How to set up the WORKFLOW Agent".
2. Uninstall the ODI Agents, as described in Section 4.6.12, "How to uninstall ODI Agent Windows Services".
3. Re-install the ODI Agents using the new port numbers, as described in Section 4.5.3, "How to start the ODI Agents".

4.6.14 How to activate or deactivate Flow Control in ODI

ODI supports dynamic enforcement of data integrity rules using Flow Control, which is implemented at the Interface level. By default, Flow Control in Oracle Business Intelligence Applications is turned off.

To activate or deactivate Flow Control:

1. In ODI Designer, display the Projects\Mappings tab and locate the Interface for which you want to modify the Flow Control behavior.
2. Edit the Interface to display the Interface: <*Name*> dialog, and display the Flow tab.
3. Use the **IKM Selection** field to select an appropriate knowledge module, and set the value of the FLOW_CONTROL option and the ANALYZE_FLOW_CONTROL option.

When Flow Control is turned on, ODI starts to filter out error records from the data flow. These error records are saved into error tables internally. You can enable recycling (processing) of error records by setting the RECYCLE_ERRORS property to YES.

Note: For more information about using Flow Control, refer to the ODI documentation.

4.6.15 List of Log Files

Oracle BI Configuration Manager produces the following log files in the <ODI_HOME>\biapps_odi\logs directory on the ODI machine:

Table 4–27 Oracle BI Configuration Manager Log Files

Log File Name	Log File Description
<root session number>_<date string>_<time string>_debug.log	<root_session_number> corresponds to the root of the session hierarchy as visible in ODI Operator. This log file must be submitted to Oracle Support for resolution of Oracle BI Configuration Manager Execution Plan related issues. Debug information from Oracle BI Applications Configuration Manager, which is stored in: <*Stand-alone WebLogic installation folder*>/user_projects/domains/base_domain For more information, see Section 4.5.8.7, "How to Launch Oracle BI Applications Configuration Manager in Debug Mode".
calcStatsFile_*name*.log	Database statistics collection log written by the Database Statistics collection utility.
dropTableFile_*name*.log	Drop table log written by the Utility Package 'Drop All Tables' and the Procedure 'Drop Table'.
indexFile_*name*.log	Index Management log written by Knowledge Modules during Data Warehouse loads.
truncateFile_*name*.log	Truncate table log written by Knowledge Modules during Data Warehouse loads.

Other log files:

Table 4–28 Other Log Files

Log File Name	Log File Description
agentservice.log	Command log information about ODI Agents, which is stored in the directory $ODI_HOME/oracledi/bin/.

4.7 Loading Source Data Using an Execution Plan

After you have installed and set up Oracle Business Intelligence Applications, configured your applications, (and optionally customized your applications) your Oracle Business Analytics Warehouse is empty. You need to create an Execution Plan to perform a full load E-LT to populate your Oracle Business Analytics Warehouse. You create Execution Plans in Oracle BI Applications Configuration Manager.

When you run an Execution Plan, Oracle Business Intelligence Applications automatically extracts the latest data from your source system, whether it is an initial full load or an incremental load. In other words, the first time you run an Execution Plan, a full load E-LT is performed. If you run the Execution Plan again, an incremental load is automatically performed.

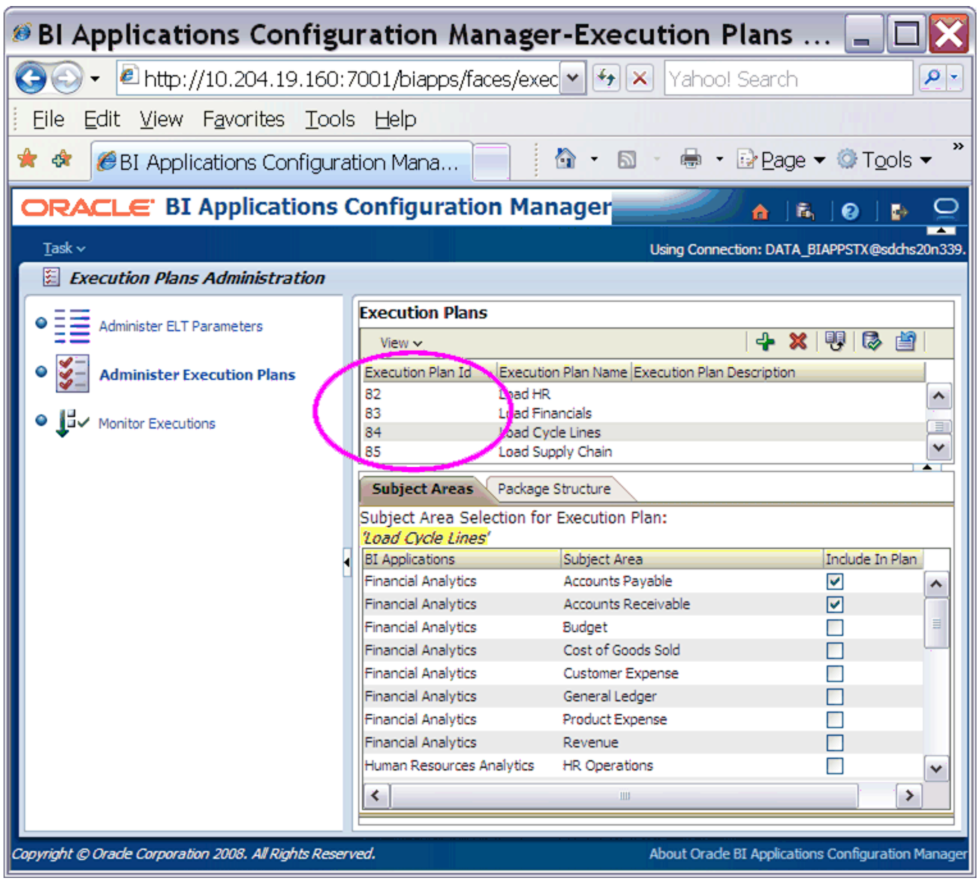
Note: Before you load your OLTP data, you typically configure your applications (for more information, see Section III, "Configuring Your Analytical Applications") and make customizations if required (for more information, see Section IV, "Customizing Oracle Business Intelligence Applications").

To load data from a source system using an Execution Plan, do the following:

1. Use Oracle BI Applications Configuration Manager to set E-LT Parameters appropriately for the applications and subjects areas that you want to load. For more information, see Section 4.7.2.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

2. Use Oracle BI Applications Configuration Manager to create an Execution Plan for the applications and subjects areas that you want to load. For more information, see Section 4.7.2.2, "How to Create an Execution Plan In Oracle BI Applications Configuration Manager".

   Oracle BI Applications Configuration Manager automatically assigns a unique Execution Plan ID to the Execution Plan. You can see the Execution Plan ID value displayed in the **Execution Plan Id** column on the Execution Plan list (see screen shot below).



3. Run the Execution Plan to load the data, in one of the following ways:

   - Use Oracle BI Applications Configuration Manager to run the Execution Plan. For more information, see Section 4.7.2.3, "How to Run an Execution Plan in Oracle BI Applications Configuration Manager".

   - Use ODI Designer to run the Execution Plan either once, or scheduled to run more than once.

     To use ODI Designer to run an Execution Plan, you must:

     a. Create a copy of the ODI master package supplied with Oracle Business Intelligence Applications, and set the value of the **OBI_EXECUTION_PLAN** setting to match the unique **Execution Plan Id** value assigned in Oracle BI

Applications Configuration Manager. For more information, see Section 4.7.1, "How to set up Master Packages to run an Execution Plan".

b. Execute the Scenario for the copy of the master package that you created. For more information about using ODI Designer to run an Execution Plan, see Section 4.7.2.3, "How to Run an Execution Plan in Oracle BI Applications Configuration Manager".

Note: The advantage of using ODI Designer to run an Execution Plan is that you can use ODI's scheduling tool to schedule the Execution Plan to run automatically.

4.7.1 How to set up Master Packages to run an Execution Plan

This section explains how to set up Master Packages in ODI Designer to run an Execution Plan. You set up Master Packages by creating a copy of the ODI master package supplied with Oracle Business Intelligence Applications, and setting the value of the **OBI_EXECUTION_PLAN** setting to match the unique **Execution Plan Id** value assigned in Oracle BI Applications Configuration Manager when you create an Execution Plan.

Note: You only need to set up Master Packages in ODI Designer to run Execution Plans if you want to use ODI Designer to execute Execution Plans, or if you want to schedule Execution Plans. If you only want to use Oracle BI Applications Configuration Manager to run Execution Plans, you can skip this task.

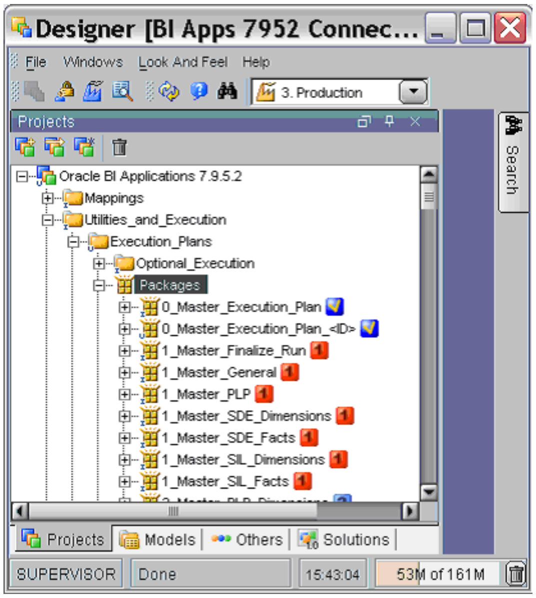
Note: Before you start this task, you need to have created an Execution Plan in Oracle BI Applications Configuration Manager and noted down the Execution Plan ID. For more information, see Section 4.7.2.2, "How to Create an Execution Plan In Oracle BI Applications Configuration Manager".

To set up Master Packages to run Execution Plans:

1. In ODI Designer, log in as SUPERVISOR, and display the Projects view.

   Note: When you log in, make sure that you specify the same JDBC URL to the Oracle Business Analytics Warehouse that you specified when you logged into Oracle BI Configuration Manager. If you specify a different JDBC URL, the Execution Plan ID that you specify in step 8 will be incorrect or not recognized.

2. Expand the Oracle BI Applications 7.9.5.2 project.
3. Select Utilities_and_Execution, then Execution Plans, then Packages.



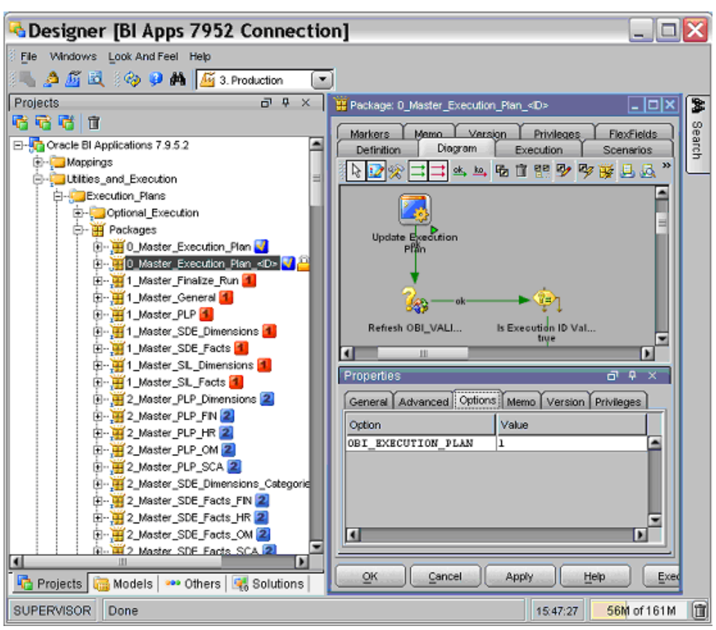
4. Right-click on 0_Master_All_Exec_Plan_<ID> and choose Duplicate.

   ODI Designer creates a copy of the package '0_Master_All_Exec_Plan_<ID>' named 'Copy of 0_Master_All_Exec_Plan_<ID>'. You can optionally rename this copy with a suitable name, for example, 'Execution Plan for HR'.

5. Double-click on the copy of the package to display the Package: <Name> dialog for this package.

6. Display the Diagram tab.

7. Select the 'Update Execution Plan' procedure at the top of the Diagram pane, then display the Options tab on the Properties pane below.



Tip: To display the Properties panel, make sure that you click the 'Show/hide Properties Panel' icon at the top of the Diagram tab.

8. On the Options tab, use the **Value** field to set the value of the OBI_EXECUTION_PLAN property.

   You set the value of the **OBI_EXECUTION_PLAN** field to the same as the unique **Execution Plan Id** value assigned to the Execution Plan in Oracle BI Applications Configuration Manager.

   For example, if an Execution Plan that you create in Oracle BI Configuration manager is assigned the ID '470', to run that Execution Plan you set the value of the **OBI_EXECUTION_PLAN** field to '470'.

   For more information about Execution Plan IDs, see Section 4.7.2.2, "How to Create an Execution Plan In Oracle BI Applications Configuration Manager").

9. Click Apply, then OK to save the details.

Notes

- To run multiple execution plans, copy the 0_Master_All_Exec_Plan_1 package, edit the package details, and set the **OBI_EXECUTION_PLAN** setting to the same as the unique **Execution Plan Id** value assigned to the Execution Plan in Oracle BI Applications Configuration Manager.

4.7.2 How to perform E-LT Using Oracle BI Applications Configuration Manager

Oracle BI Applications Configuration Manager enables you to create and run Execution Plans, and monitor their progress. To perform E-LT using Oracle BI Applications Configuration Manager, do the following:

1. Configure E-LT Parameters to enable you to control E-LT processes (for more information, see Section 4.7.2.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager").

2. Create an Execution Plan (for more information, see Section 4.7.2.2, "How to Create an Execution Plan In Oracle BI Applications Configuration Manager").

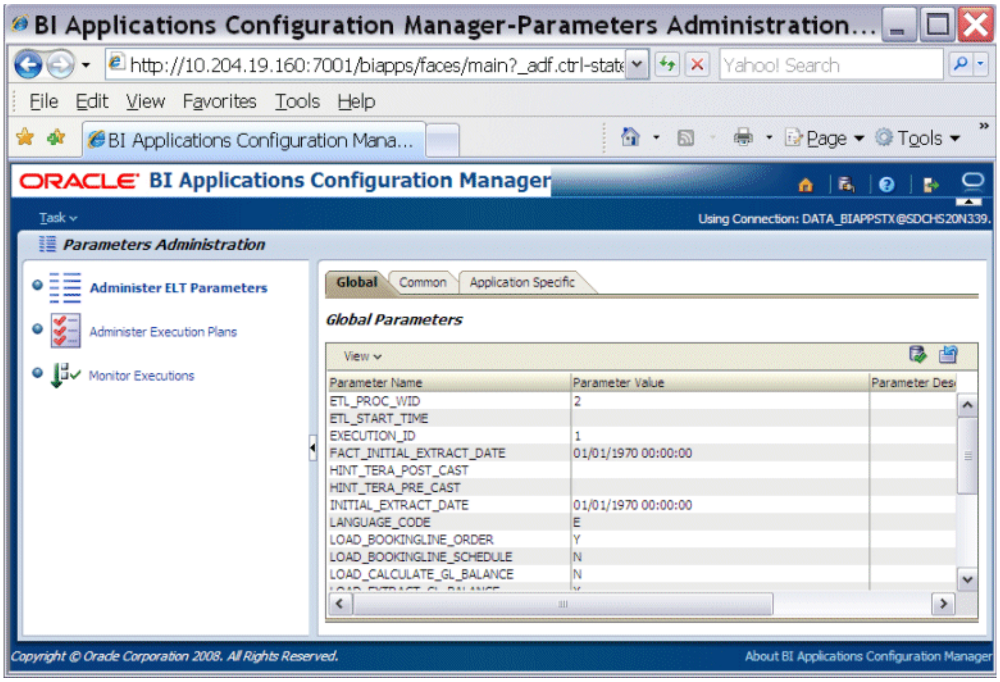
3. Run the Execution Plan (for more information, see Section 4.7.2.3, "How to Run an Execution Plan in Oracle BI Applications Configuration Manager").

4.7.2.1 How to Set E-LT Parameters In Oracle BI Applications Configuration Manager

You use Oracle BI Applications Configuration Manager to specify E-LT parameter values that are used to control the E-LT processes for your Applications and Subject Areas.

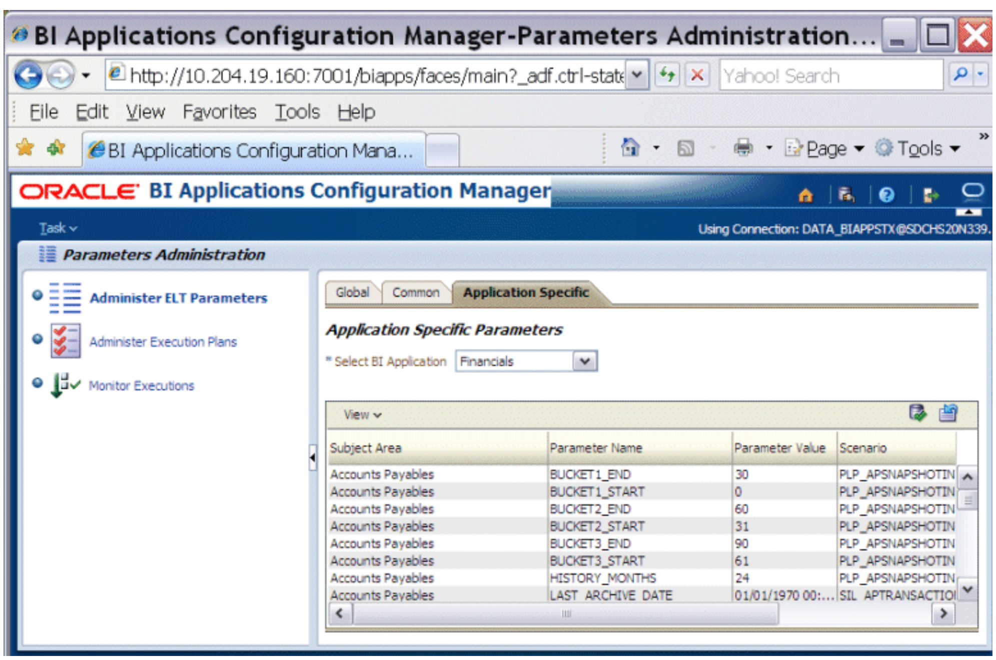
1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").

2. Select the **Administer ELT Parameters** link to display the Parameters Administration page.



3. Use the Global, Common, and Application Specific tabs to change the default values of the parameters.

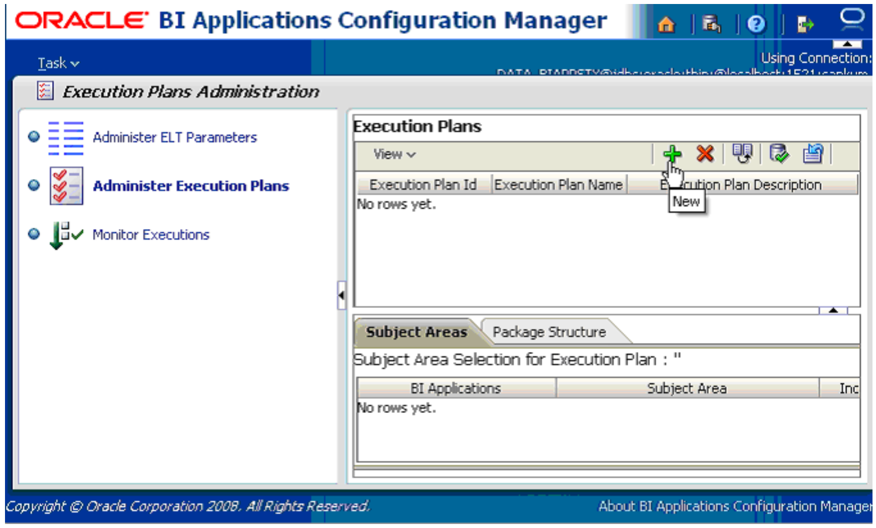
At the Application Specific tab, the parameters are grouped into Subject Area within Application, to help you locate the parameters that are specific to your Application.



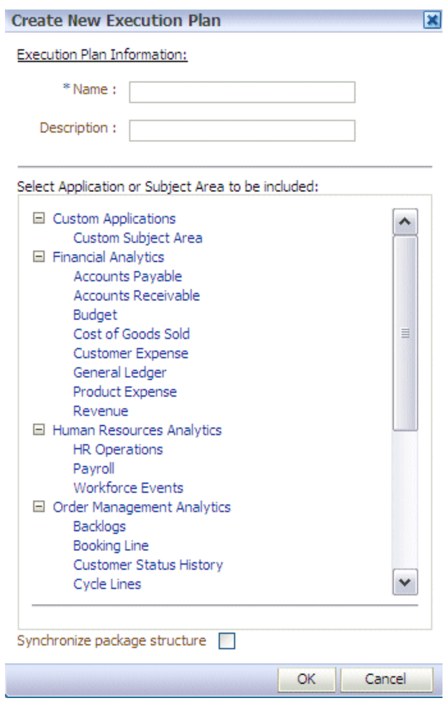
4.7.2.2 How to Create an Execution Plan In Oracle BI Applications Configuration Manager

You use Oracle BI Applications Configuration Manager to create and manage Execution Plans.

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer Execution Plans** link.



3. Click the New button (+) to display the Create New Execution Plan page.



4. Enter the appropriate execution plan information, as described in the table below.

Table 4–29 Create New Connection page fields

Field	Description
Name	Enter a short name to identify the Execution Plan. This name will be displayed in Oracle BI Applications Configuration Manager and ODI Designer. For example, 'Load HR All'.

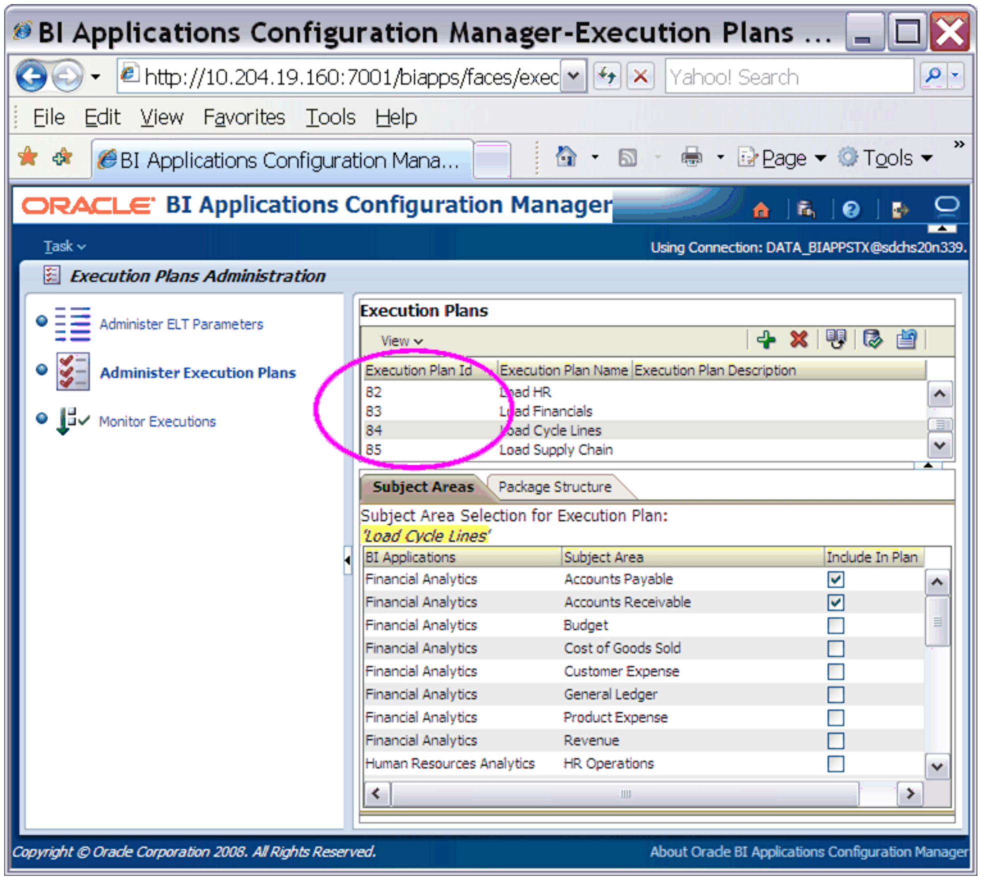
Table 4–29 (Cont.) Create New Connection page fields

Field	Description
Description	Enter a short name to description of the Execution Plan. This description will be displayed in Oracle BI Applications Configuration Manager and ODI Designer.
Select Application or Subject Area to be included	Select the analytic application from which you want to load data (for example, Financials, HR, Supply Chain). To select one item, click the item. To select multiple contiguous items, press Shift and click. To select multiple non-contiguous items, press Ctrl and click.
Synchronize Package Structure.	Select this check box. In order for the package structures to be displayed correctly on the Package Structure tab, the subject areas associated to an execution plan must be synchronized within internal Oracle BI Applications Configuration Manager tables. If you do not synchronize the package structures when creating the execution plan or when you made changes to the subject areas associated to the execution plan, you must synchronize by clicking the Synchronize package structure button that appears on the Package Structure tab in order to get correct package structure. The synchronization process may take up to several minutes depending on the network traffic between the Oracle Business Analytics Warehouse database and the Oracle BI Applications Configuration Manager installation.

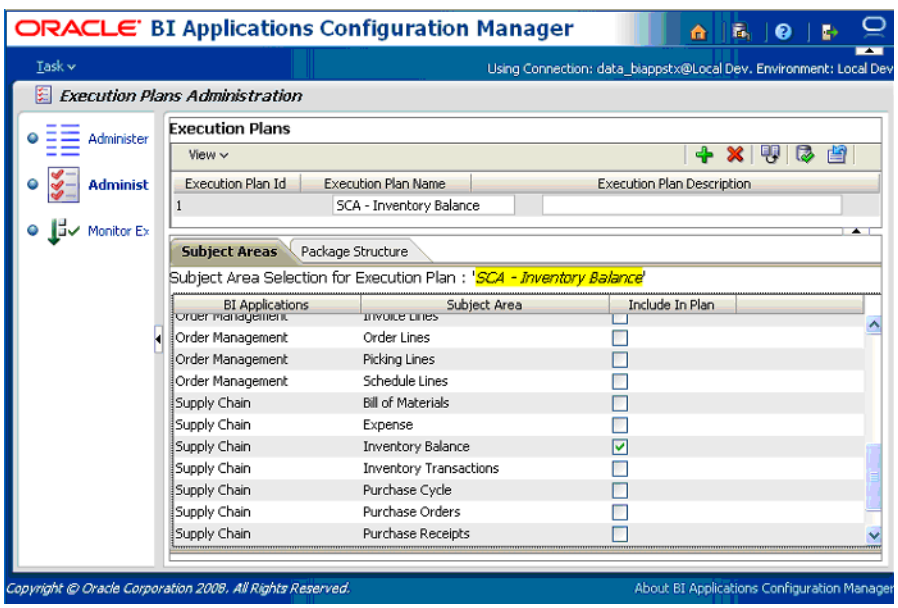
5. Save the details.

   Note: The OK button is only active if you have included at least one Subject Area in the Execution Plan.

   When you save the details, Oracle BI Applications Configuration Manager automatically assigns a unique Execution Plan ID to the Execution Plan. You can see the Execution Plan ID value displayed in the **Execution Plan Id** column on the Execution Plan list (see screen shot below).



You can use the Execution Plans Administration page to modify the Execution Plan if required (for example, to add or remove Subject Areas), as well as start and monitor it.

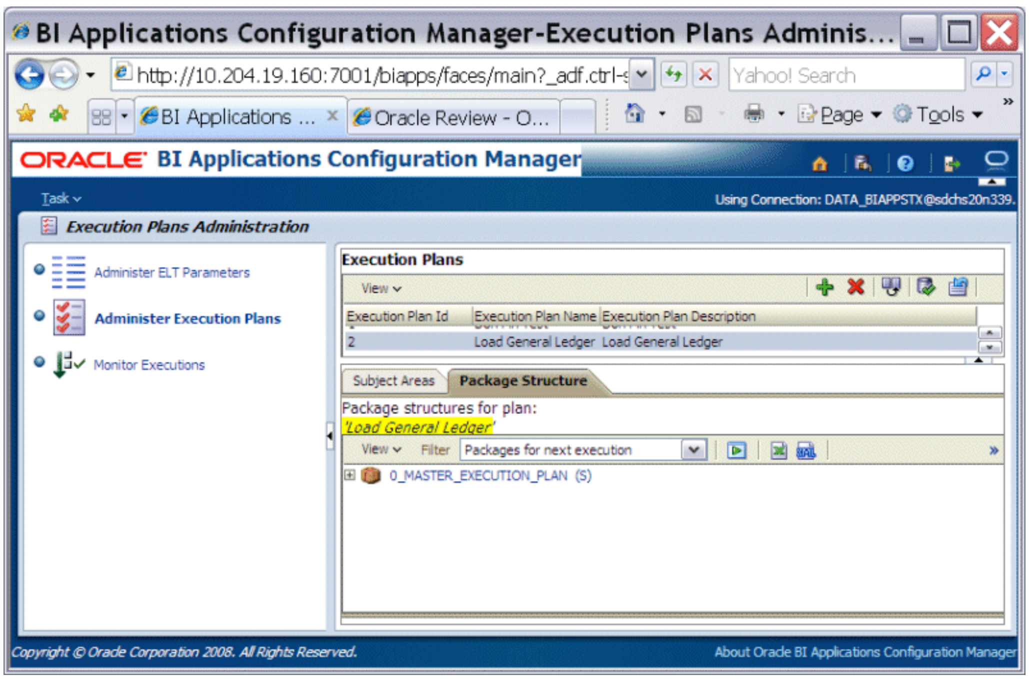


4.7.2.3 How to Run an Execution Plan in Oracle BI Applications Configuration Manager

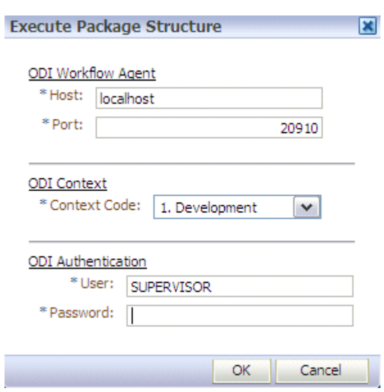
You run an Execution Plan to load data into the Oracle Business Analytics Warehouse for the Applications and Subject Areas specified by the Execution Plan.

To Run an Execution Plan Using Oracle BI Applications Configuration Manager:

1. Start Oracle BI Applications Configuration Manager (or more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer Execution Plans** link.
3. In the Execution Plans list, select the Execution Plan that you created earlier (or more information, see Section 4.7.2.2, "How to Create an Execution Plan In Oracle BI Applications Configuration Manager").
4. Display the Package Structure tab.



5. Select the 0_MASTER_EXECUTION_PLAN package.
6. Click the Execute button to display the Execute Package Structure dialog.

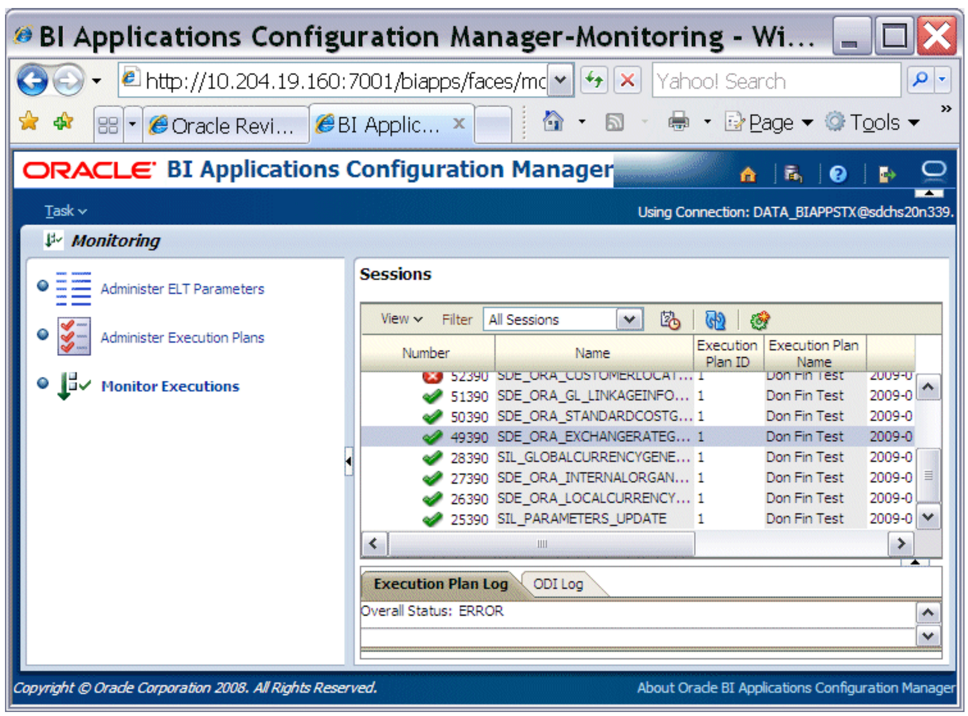


7. Enter the appropriate execution plan information, as described in the table below, then click OK to save the details.

Table 4–30 Execute Package Structure dialog fields

Field	Description
Host	Specify the fully qualified host name of the ODI machine, or 'localhost'. For example, US12345.us.company.com. For more information, see Section 4.5.5.2.2, "How to set up the WORKFLOW Agent".
Port	Specify the port number that you defined for the WORKFLOW agent. For example, 20910. For more information, see Section 4.5.5.2.2, "How to set up the WORKFLOW Agent".
ODI Context	Select the appropriate Context in which you are working.
User	Specify SUPERVISOR.
Password	Specify SUPERVISOR.

8. Use the Sessions page to monitor the process (select the **Monitor Execution** link).



Notes

- You can only monitor sessions from Execution Plans that you created in Oracle BI Applications Configuration Manager. You cannot monitor packages that you executed in ODI Designer.
- You can also execute an Execution Plan from the Monitor Executions page. Select the Monitor Executions link to display the Sessions page. Then, click the Create New Session icon to display the Execute Package Structure dialog, and select the Execution Plan from the **Execution Plan** drop down list.

- Oracle Business Intelligence Applications truncates tables, drops and creates indexes, and collects statistics for the Knowledge Module.
- During a E-LT data load, you can also find information about table truncation, indexes, and statistics collection in the log directory, located in $ODI_HOME\oracledi\biapps_odi\logs. For example, D:\Ora_Home1\oracledi\biapps_odi\logs. For a full list of log files, see Section 4.6.15, "List of Log Files".

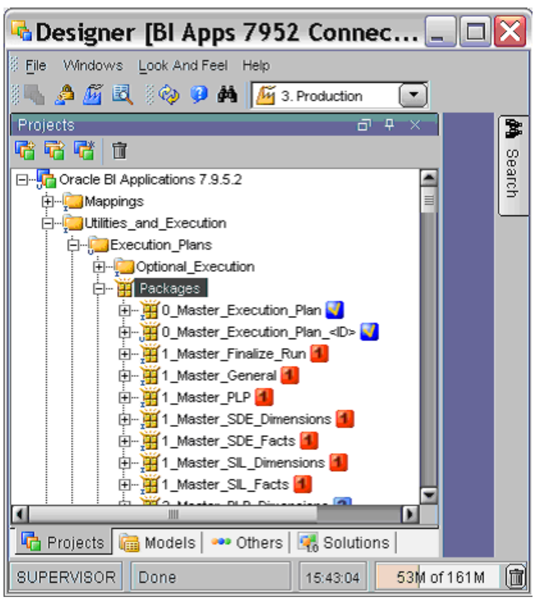
4.7.3 How to perform E-LT Using ODI Designer

You can use ODI Designer to run an Execution Plan once or schedule it to perform more than once.

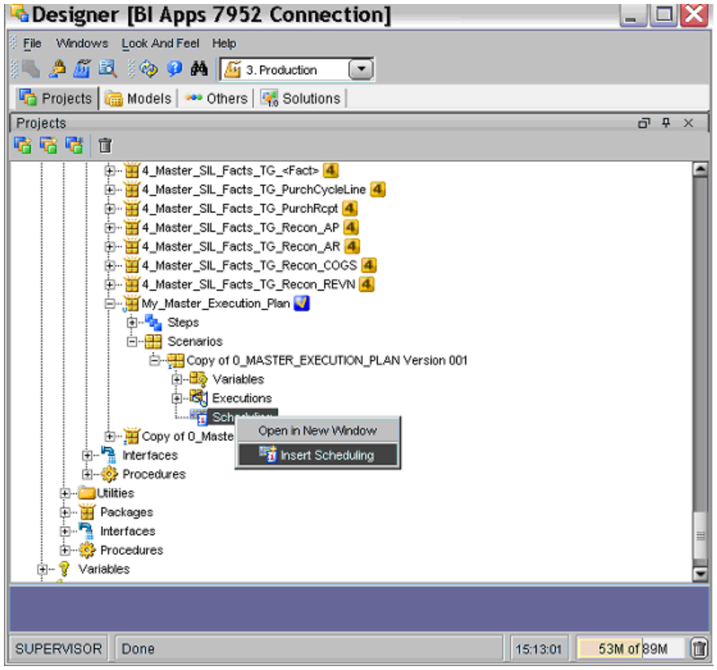
Note: Before you can perform E-LT using ODI Designer, you must have set up Master Packages in ODI Designer to run an Execution Plans (for more information, see Section 4.7.1, "How to set up Master Packages to run an Execution Plan").

To perform a full-load E-LT process using ODI Designer:

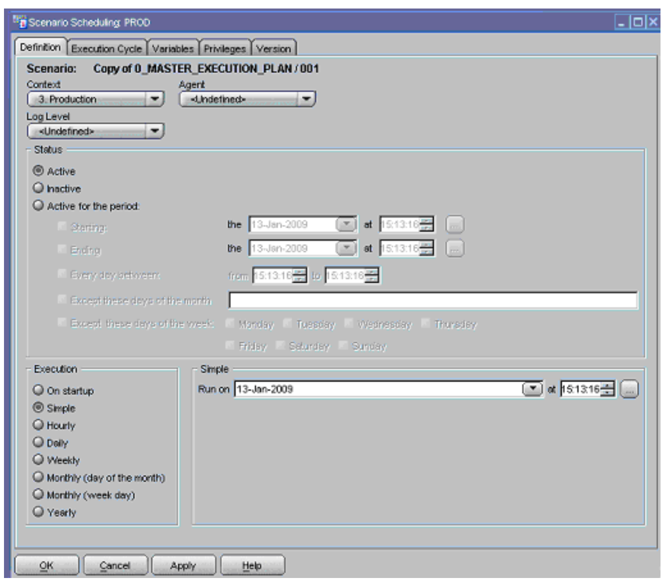
1. In ODI Designer, display the Projects view.
2. Expand the Oracle BI Applications 7.9.5.2 project.
3. Select Utilities_and_Execution, then Execution Plans, then Packages.



4. Do one of the following:
   - To perform a single full-load E-LT process:
     a. Right-click on the package that you set up earlier as described in Section 4.7.1, "How to set up Master Packages to run an Execution Plan" (for example, 0_Master_Execution_plam_<ID>) and select Execute.
     b. Use ODI Operator to monitor the processes.
   - To schedule a full-load E-LT process:
     a. Expand the package that you want to schedule.
     b. Expand the Scenario for that package.



c. Right-click on the Scheduling node and select Insert Scheduling to display the Scenario Scheduling: <*Name*> dialog.



d. Use the Scenario Scheduling: <*Name*> dialog to specify the scheduling details.

e. Save the details.

f. When the process starts, use ODI Operator to monitor the process.

For more detailed information about using the ODI to perform E-LT, refer to the Oracle Data Integrator documentation.

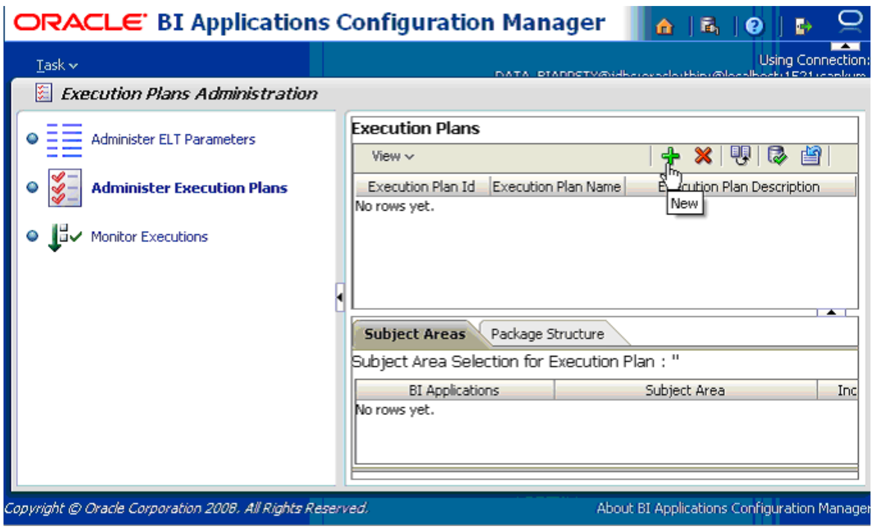
Note: ODI Designer does not support concurrent package execution for the Master Package. You can only execute Master Packages one-at-a-time.

4.7.4 Example of Running A Full Load E-LT in Oracle BI Applications Configuration Manager

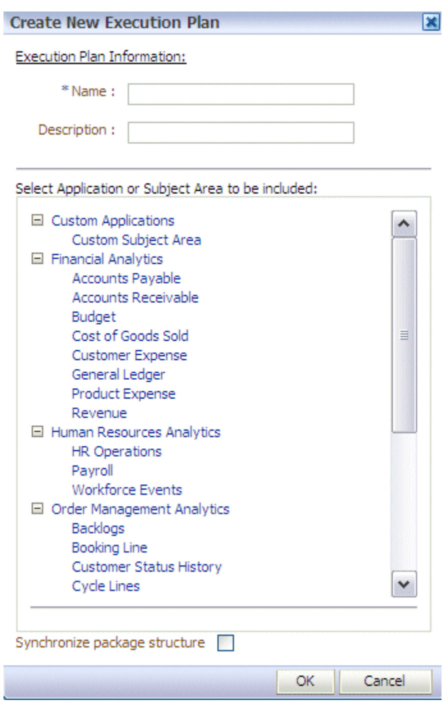
This example shows you how to use Oracle BI Applications Configuration Manager to load source system data by creating and running an Execution Plan.

In this example, you have installed Oracle Financial Analytics with an Oracle EBS 11.5.10 OLTP data source, and you want to load OLTP data for the subject area General Ledger, as follows:

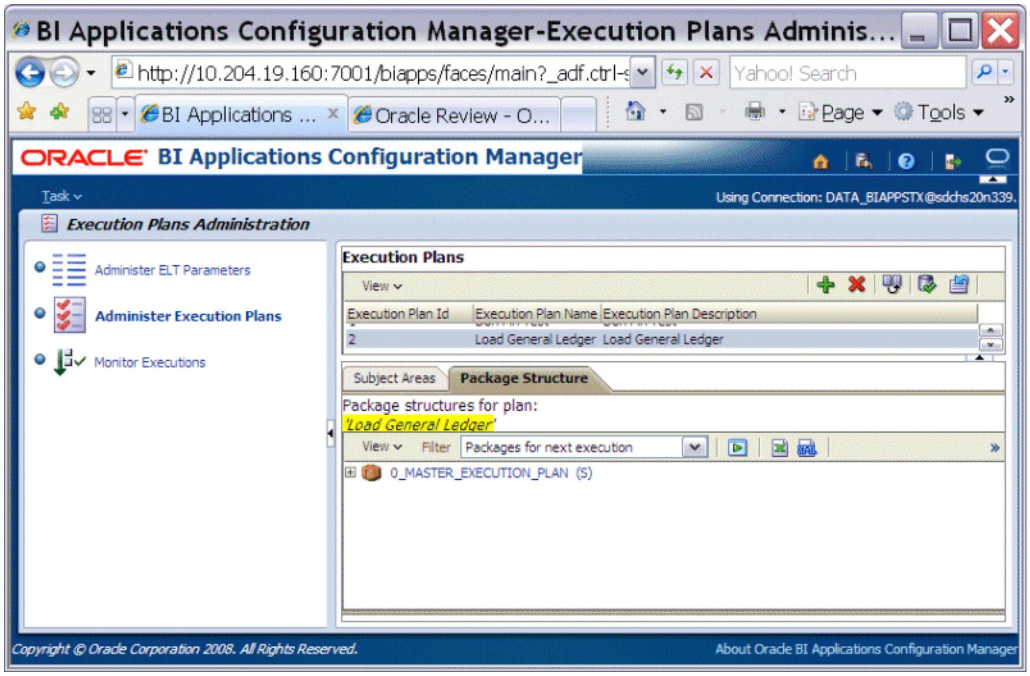
1. Start Oracle BI Applications Configuration Manager (or more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer Execution Plans** link.



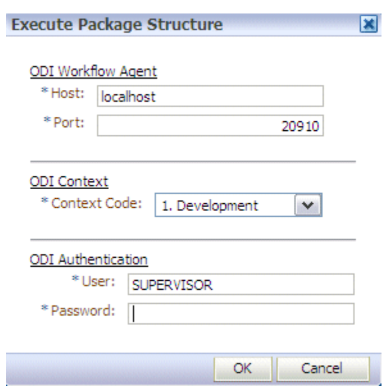
3. Click the New button (+) to display the Create New Execution Plan page.



4. Enter 'Financials - General Ledger' in the **Name** field.
5. In the **Select Application or Subject Area to be included** box, select General Ledger.
6. Click OK.
7. Select the **Administer Execution Plans** link.
8. In the Execution Plans list, select the 'Financials - General Ledger' Execution Plan.
9. Display the Package Structure tab.



10. Click the Execute button to display the Execute Package Structure dialog.

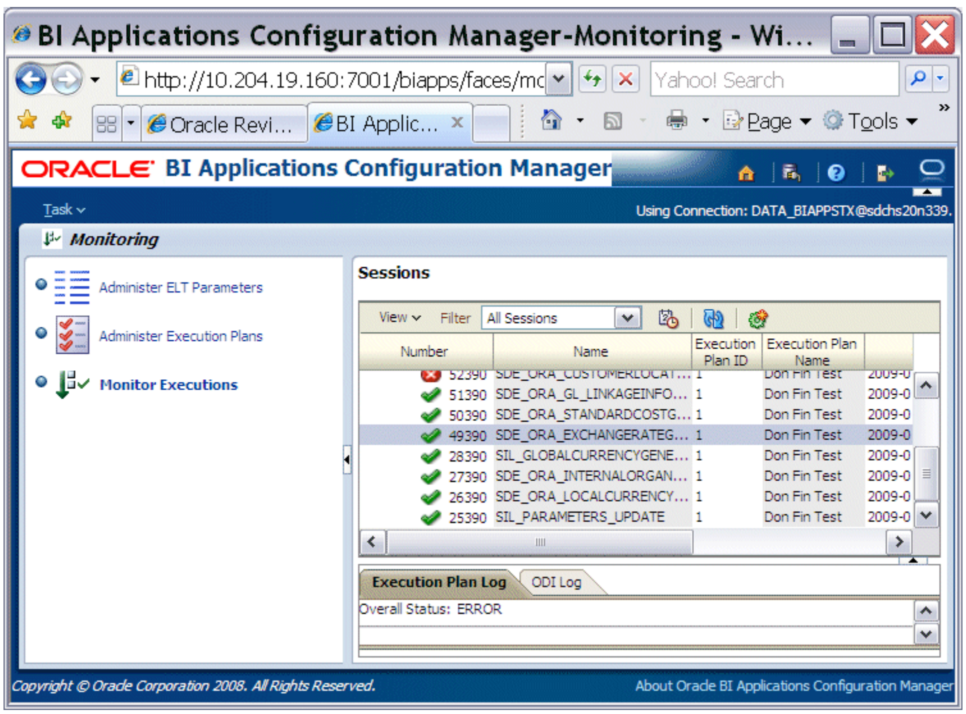


11. Enter the appropriate execution plan information, as described in the table below, then click OK to save the details.

Table 4–31 Execute Package Structure dialog fields

Field	Description
Host	Specify the fully qualified host name of the ODI machine, or 'localhost'. For example, US12345.us.company.com. For more information, see Section 4.5.5.2.2, "How to set up the WORKFLOW Agent".
Port	Specify the port number that you defined for the WORKFLOW agent. For example, 20910. For more information, see Section 4.5.5.2.2, "How to set up the WORKFLOW Agent".
ODI Context	Select the appropriate Context in which you are working.
User	Specify SUPERVISOR.
Password	Specify SUPERVISOR.

12. Use the Sessions page to monitor the process (select the **Monitor Execution** link).



Part III

Configuring Your Analytical Applications

Part III explains how to configure and modify the out-of-the-box functionality in Oracle Business Intelligence Applications, and contains the following chapters:

- Chapter 5, "Configuring Common Areas and Dimensions"
- Chapter 6, "Configuring Oracle Procurement and Spend Analytics"
- Chapter 7, "Configuring Oracle Financial Analytics"
- Chapter 8, "Configuring Oracle Supply Chain and Order Management Analytics"
- Chapter 9, "Configuring Oracle Human Resources Analytics"
- Chapter 10, "Configuring the Oracle Business Intelligence Applications Repository"

Note: For a high level road map for installation, configuration, and customization steps for Oracle Business Intelligence Applications, see Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle Business Intelligence Applications With ODI".

5

Configuring Common Areas and Dimensions

This chapter contains configuration steps for Oracle Business Intelligence Applications that you need to follow for any applications you deploy (for example, Oracle Financial Analytics, Oracle Human Resources Analytics, Oracle Procurement and Spend Analytics), and contains the following topics:

- Section 5.1, "Steps Required Before A Full Load"

To find out about other possible tasks required to deploy Oracle Business Intelligence Applications, see Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle Business Intelligence Applications With ODI".

5.1 Steps Required Before A Full Load

This section contains configuration steps required before a full data load that apply to Oracle Business Intelligence Applications deployed with an Oracle EBS 11i source system, and contains the following topics:

- Section 5.1.1, "How to Configure Initial Extract Date"
- Section 5.1.2, "Configuring Global Currencies"
- Section 5.1.3, "How to Configure Fiscal Calendars"
- Section 5.1.4, "Configuring the General Ledger Account Hierarchies"
- Section 5.1.5, "Configuring Product Hierarchy, Master Inventory Organization, and Group Accounts"
- Section 5.1.6, "Configuration Steps for Controlling Your Data Set"

5.1.1 How to Configure Initial Extract Date

Initial Extract Date is required when you extract data for a full load. It reduces the volume of data in the initial load. The specified initial extract date will be used as a filter on the creation date of OLTP data in the selected full extract mapping.

When you set the Initial Extract Date parameter, make sure that you set it to the beginning of an accounting period, and not a date in the middle of an accounting period. For example, if you decide to extract data from June 2005, and the June 2005 accounting period starts from 5th June, set the date to 5th June, 2005.

To configure the initial extract date in your warehouse:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").

2. Select the **Administer ELT Parameters** link.
3. Display the Global tab.
4. Locate the following parameter and use the **Parameter Value** field to set the value:
   - INITIAL_EXTRACT_DATE
5. Save your changes.

   For more information about specifying parameter values, see Section 4.6.3.3.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

5.1.2 Configuring Global Currencies

Currency conversions are required because your business might have transactions involving multiple currencies. To create a meaningful report, you have to use a common currency. The Oracle Business Analytics Warehouse stores amounts in the following currencies:

- Document currency. The currency of the transaction. For example, if you purchase a chair from a supplier in Mexico, the document currency is probably the Mexican Peso. Or, if you made a business trip to the UK and filed an expense report for meal expenses in the UK, the document currency of the expense report will be in GBP.
- Local currency. This is the base currency of your ledger, the currency in which your accounting entries are recorded.
- Global currencies. The three out of the box global currencies provided by Oracle Business Intelligence Applications.

  For example, if your organization is a multinational enterprise that has its headquarters in the United States, you probably want to choose US dollars (USD) as one of the three global currencies. The global currency is useful when creating enterprise-wide reports. For example, a user might want to view enterprise-wide data in other currencies. For every monetary amount extracted from the source, the load mapping loads the document amounts and local amounts into the target table.

  To configure Global Currencies, follow the steps in Section 5.1.2.1, "How to configure Global Currencies".

  The load mapping also loads the exchange rates required to convert the document amount into each of the three global currencies. In the target table, there will be two amount columns, and three exchange rate columns. Typically, the source system provides the document currency amount, which is the default currency handling setup for the Oracle Business Analytics Warehouse. If the source system provides only the document currency amount, the Source Adapter performs lookups to identify the local currency codes based on the source system. Based on the source system the appropriate currencies are assigned. After the lookups occur, the extract mapping provides the load mapping with the document currency amount and the document and local currency codes. The load mapping will then use the provided local currency codes and perform currency conversion to derive the local amount. The load mapping will also fetch the global currencies that are specified by parameters in Oracle BI Applications Configuration Manager and look up the corresponding exchange rates to each of the three global currencies.

  To configure Exchange Rate Types, follow the steps in Section 5.1.2.2, "How to Configure Exchange Rate Types".

5.1.2.1 How to configure Global Currencies

To configure global currencies:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Common tab.
4. Locate the following parameters and use the **Parameter Value** field to set the value:
   - GLOBAL1_CURR_CODE
   - GLOBAL2_CURR_CODE
   - GLOBAL3_CURR_CODE

   Make sure you spell the exchange rate type values as they are spelled in your source OLTP system. For example, USD.
5. Save your changes.

   For more information about specifying parameter values, see Section 4.6.3.3.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

5.1.2.2 How to Configure Exchange Rate Types

When Oracle Business Intelligence Applications converts your transaction records' amount from document currency to global currencies, it also requires the exchange rate types to use to perform the conversion. For each of the global currencies, Oracle Business Intelligence Applications also allows you to specify the exchange rate type to use to perform the conversion. Oracle Business Intelligence Applications also provides three global exchange rate types for you to configure.

Oracle Business Intelligence Applications also converts your transaction records' amount from document currency to local currency. Local currencies are the base currencies in which your accounting entries and accounting reports are recorded. In order to perform this conversion, Oracle BI Application also allows you to configure the rate type that you want to use when converting the document currency to the local currency.

To configure the exchange rate types:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Common tab.
4. Locate the following parameters and use the **Parameter Value** field to set the value:
   - GLOBAL1_RATE_TYPE
   - GLOBAL2_RATE_TYPE
   - GLOBAL3_RATE_TYPE
   - DEFAULT_LOC_RATE_TYPE (the conversion rate type for document currency to local currency conversion).

Make sure that you spell the exchange rate type values as they are spelled in your source OLTP system. For example, Corporate.

5. Save your changes

For more information about specifying parameter values, see Section 4.6.3.3.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

5.1.3 How to Configure Fiscal Calendars

This sections explains how you configure the time dimension in the Oracle Business Analytics Warehouse, and contains the following sections:

- Section 5.1.3.1, "Overview To Setting Up The Time Dimension"
- Section 5.1.3.2, "Setting Up Fiscal Calendar"
- Section 5.1.3.3, "Setting Up Fiscal Calendar By Fiscal Week"
- Section 5.1.3.4, "Setting Up Fiscal Calendar By Fiscal Month"
- Section 5.1.3.5, "How to Set Up The Fiscal Calendar"
- Section 5.1.3.6, "Reloading the Time Dimension Tables After Your Data Warehouse is Populated"
- Section 5.1.3.7, "Notes"

5.1.3.1 Overview To Setting Up The Time Dimension

When you configure Time Dimension tables, W_DAY_D is the base table that represents the time dimension in the Oracle Business Analytics Warehouse. There are two lines of aggregate dimension tables built on this table, as follows:

- Regular Calendar tables.
- Fiscal Calendar tables.

Table 5–1 shows the tables used to store calendar information.

Table 5–1 Tables in base table W_DAY_D that are used to store calendar information

Regular calendar tables in W_DAY_D	Fiscal calendar tables in W_DAY_D
W_WEEK_D	W_FSCL_WEEK_D
W_MONTH_D	W_FSCL_MONTH_D
W_QTR_D	W_FSCL_QTR_D
W_YEAR_D	W_FSCL_YEAR_D

There are two parameters START_DATE and END_DATE for the task SIL_DayDimension that need to be setup to load the calendar data in W_DAY_D. The SIL mappings use standard time functions to create records for each calendar day falling within the boundary defined by these two parameters. Once the records are created in W_DAY_D, the aggregate calendar tables are loaded by their respective SIL mapping.

5.1.3.2 Setting Up Fiscal Calendar

Installed out of the box, Oracle Business Intelligence Applications supports one fiscal calendar. Fiscal data is first loaded in the W_DAY_D table and then the SIL mappings read data from W_DAY_D and load data into the aggregate Fiscal Time Dimension tables such as Fiscal Week, Fiscal Month, Fiscal Quarter and Fiscal Year.

You may choose to provide Fiscal calendar information in terms of the Fiscal Weeks of your organization or in terms of the Fiscal months of your organization. In either case, The SIL mappings are designed to derive the Fiscal Week from the Start and End Date of a Fiscal Month by grouping into periods of seven days each.

5.1.3.3 Setting Up Fiscal Calendar By Fiscal Week

In this option you provide data for the Fiscal Year, Fiscal Month, Fiscal Week and Start Date of Fiscal Week. The Fiscal Month information is derived using the 4-4-5 rule for grouping weeks into months. The Fiscal Week End Date is derived based on the start date of the next week that is supplied in the data file. Fiscal Months are grouped into sets of 4 months each to determine the Fiscal Quarter.

5.1.3.4 Setting Up Fiscal Calendar By Fiscal Month

In this option you can provide data at the level of Fiscal Month. The SIL_DayDimension_FiscalMonth_Extract task divides the Fiscal Month into Fiscal Weeks of seven days each. If the number of days in the Fiscal Month is not in multiples of seven, the last week will have less number of days.

5.1.3.5 How to Set Up The Fiscal Calendar

To set up Fiscal Calendar by Fiscal Week:

1. Open the file fiscal_week.csv using a text editor in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\srcfiles directory.
2. Enter the Fiscal Year, Fiscal Month, Fiscal Week and the Start Date of Fiscal Week in the format YYYYMMDD.

   Records must be inserted in ascending order of Fiscal Year, Fiscal Month, Fiscal Week.
3. Save the fiscal_week.csv file.
4. In Oracle BI Applications Configuration manager, click Parameters, display the Global tab, and do the following:
   - Set the value of LOAD_FISCAL_CAL_BY_WEEK to Y.
   - Set the value of LOAD_FISCAL_CAL_BY_MONTH to N.

To set up Fiscal Calendar by Fiscal Month:

1. Open the file fiscal_month.csv using a text editor in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\srcfiles directory.
2. Enter the Fiscal Year and Fiscal Month and the Start Date of Fiscal Month in the format YYYYMMDD.

   Records must be inserted in ascending order of Fiscal Year and Fiscal Month.
3. Save the fiscal_month.csv file.
4. In Oracle BI Applications Configuration manager, select the **Administer ELT Parameters** link, display the Global tab, and do the following:
   - Set the value of LOAD_FISCAL_CAL_BY_WEEK to N.
   - Set the value of LOAD_FISCAL_CAL_BY_MONTH to Y.

5.1.3.6 Reloading the Time Dimension Tables After Your Data Warehouse is Populated

The data in the time dimension is loaded once during the initial full load. Subsequently, the SIL_*_UpdateFlag mappings run everyday to update the domain value codes, which indicate whether a day, week, month, quarter, or year is 'Current', 'Next' or 'Previous' as of the current day. The SIL_Fiscal_UpdateFlag mappings also update the flags that indicate whether a fiscal week, month, quarter or year is 'Current', 'Previous' or 'Next' with respect to the system date.

You might want to extend the range of data that you have in your time dimension some time after the warehouse is in production. In order to achieve this, please follow the steps below that will start a full load ELT run of the W_DAY_D and all of the aggregate time dimension tables.

To set up the load strategy of the time dimension table:

1. Edit the C_LOAD_DATES table in the data warehouse schema.
2. Refresh the dates (ETL_LOAD_DATE and LAST_MAX_DATE) for all the rows in the table where TARGET_TABLE_NAME is in one of the following:
   - W_DAY_D
   - W_WEEK_D
   - W_QTR_D
   - W_MONTH_D
   - W_YEAR_D
   - W_FSCL_WEEK_D
   - W_FSCL_QTR_D
   - W_FSCL_MONTH_D
   - W_FSCL_YEAR_ D
3. In Oracle BI Applications Configuration Manager, select the **Administer ELT Parameters** link, display the Common tab, and set the START_DATE and END_DATE parameters to specify the new date range.
4. Save your changes

   For more information about specifying parameter values, see Section 4.6.3.3.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

5.1.3.7 Notes

The following notes pertain to the process of setting up the fiscal calendar:

- If there is a week (starting on a Sunday and ending on a Saturday) that falls across two Calendar years, the week is counted in both years. For example the week that starts on 12/30/2007 will be counted in both 2007 and 2008. In 2007 the week Start Date will 12/30/2007 and the End Date will be 12/31/2007. In 2008 this will be the first week with Start Date as 01/01/2008 and End Date as 01/05/2008.
- W_DAY_D stores 31 records for each month irrespective of whether the month actually has 31 days or not. If the month actually has less number of days, there will be records with Null values in the Calendar Date and Day Date columns. These extra records are loaded for the calculation of Period Ago metrics in the RPD, and will not affect the ELT or reporting.

- There are some attributes on the W_DAY_D table that are not mapped in the physical layer of the RPD. Therefore, before creating any new attribute in the RPD, check whether the attribute is already available in the physical layer and if it can be mapped directly.
- If your fiscal calendar contains more than 12 months, the extra months will be assigned a value of 0 for the Fiscal Quarter. The same holds for the Fiscal Trimester and Fiscal Half values.

5.1.4 Configuring the General Ledger Account Hierarchies

These configuration steps are required if you are deploying Oracle Financial Analytics, Oracle Procurement and Spend Analytics, or Oracle Supply Chain and Order Management Analytics. This section contains the following topics:

- Section 5.1.4.1, "Overview to General Ledger Account Hierarchies"
- Section 5.1.4.2, "Configuring General Ledger Account Hierarchies Using General Ledger Accounting Flexfield value sets definitions"
- Section 5.1.4.3, "Configuring General Ledger Account Hierarchies Using Financial Statement Generator (FSG) Report Definition"

5.1.4.1 Overview to General Ledger Account Hierarchies

The are two ways to set up hierarchies in Oracle Financial Analytics:

- Using General Ledger Accounting Flexfield Value Sets Definitions (for more information, see Section 5.1.4.2, "Configuring General Ledger Account Hierarchies Using General Ledger Accounting Flexfield value sets definitions").
- Using the Financial Statement Generator (FSG) Report Definition (for more information, see Section 5.1.4.3, "Configuring General Ledger Account Hierarchies Using Financial Statement Generator (FSG) Report Definition").

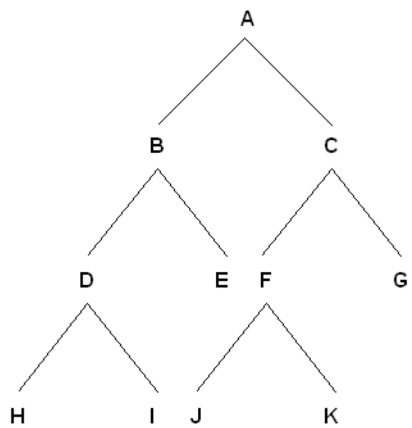
Whichever method you choose to set up General Ledger Account hierarchies, you store the hierarchy information in the W_HIERARCHY_D table.

As an example, the hierarchy for US Acct might have the following structure:

- Node A has child nodes B and C.
- Node B has child nodes D and E.
- Node C has child nodes F and G.
- Node D has child nodes H and I.
- Node F has child nodes J and K.

The figure below shows the example hierarchy for US Acct.

Figure 5–1 Example hierarchy US Acct



The table below shows how the hierarchy US Acct would be stored in the W_HIERARCHY_D table.

Table 5–2 Example hierarchy US Acct stored in W_HIERARCHY_D

HIER_KEY	HIER_ NAME	HIER1_ CODE	HIER2_ CODE	HIER3_ CODE	HIER4_ CODE	HIER5_ CODE	6 - 19	HIER20_ CODE
1	US Acct	A	B	D	H	H	H	H
2	US Acct	A	B	D	I	I	I	I
3	US Acct	A	B	E	E	E	E	E
4	US Acct	A	C	F	J	J	J	J
5	US Acct	A	C	F	K	K	K	K
6	US Acct	A	C	G	G	G	G	G

5.1.4.2 Configuring General Ledger Account Hierarchies Using General Ledger Accounting Flexfield value sets definitions

These configuration steps are required if you are deploying Oracle Financial Analytics, Oracle Procurement and Spend Analytics, or Oracle Supply Chain and Order Management Analytics. This section contains the following topics:

- Section 5.1.4.2.1, "Overview"
- Section 5.1.4.2.2, "About Configuring the ELT Process for GL Accounting flexfields"
- Section 5.1.4.2.3, "How To Set Up Hierarchies With General Ledger Accounting flexfields"

5.1.4.2.1 Overview Oracle EBS supports up to 30 segments in which to store accounting flexfields. flexfields are flexible enough to support complex data configurations, for example:

- You can store data in any segments.
- You can use more or fewer segments per chart of account, as required.
- You can specify multiple segments for the same chart of account.

An Example Data Configuration for a Chart of Accounts

A single company might have a US Chart of Account and an APAC Chart of Account with the following data configuration:

Table 5–3 Example Chart of Accounts

Segment Type	US Chart of Account (4256) value	APAC Chart of Account (4257) value
Company	Stores in segment 3	Stores in segment 1
Natural Account	Stores in segment 4	Stores in segment 3
Cost Center	Stores in segment 5	Stores in segment 2
Geography	Stores in segment 2	Stores in segment 2
Line of Business (LOB)	Stores in segment 1	Stores in segment 4

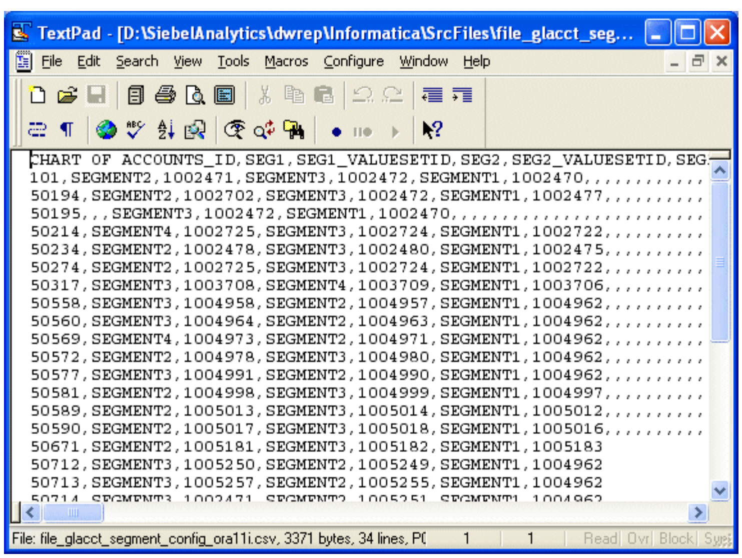
The example shows that in Chart Of Account 4256, Company is stored in the segment3 column in the Oracle EBS table GL_CODE_COMBINATIONS_ALL. In Chart Of Account COA4257, Company is stored in segment1 column in GL_CODE_COMBINATIONS_ALL table. The objective of this configuration file is to make sure that when segment information is extracted into the warehouse table W_GL_ACCOUNT_D, segments with the same nature from different chart of accounts are stored in the same column in W_GL_ACCOUNT_D.

For example, we can store Company segments from COA 4256 and 4257 in segment1 column in W_GL_ACCOUNT_D; and Cost Center segments from COA 4256 and 4257 in segment2 column in W_GL_ACCOUNT_D, and so forth.

5.1.4.2.2 About Configuring the ELT Process for GL Accounting flexfields Before you run the ELT process for General Ledger Accounts, you need to specify the segments that you want to analyze. To specify the segments that you want to analyze, use the following ELT configuration file:

$ODI_HOME\biapps_odi\odifiles\odidatafiles\srcfiles\file_glacct_segment_configur_ora.csv

Figure 5–2 Screen Shot of file_glacct_segment_config_ora.csv File Opened in a Text Editor



In the file_glacct_segment_config_ora.csv file, you need to specify the segments of the same type in the same column. For example, you might store all Cost Center segments from all chart of accounts in one column, and all Company segments from all chart of accounts in another column.

For example, you might want to do the following:

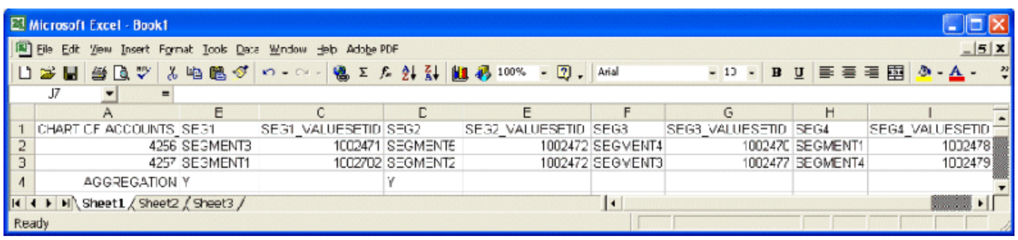
- Analyze GL account hierarchies using only Company, Cost Center, Natural Account, and LOB.

  You are not interested in using Geography for hierarchy analysis.

- Store all Cost Center segments from all COAs in ACCOUNT_SEG2_CODE column in W_GL_ACCOUNT_D.
- Store all Natural Account segments from all COAs in ACCOUNT_SEG3_CODE column in W_GL_ACCOUNT_D.
- Store all LOB segments from all COAs in ACCOUNT_SEG4_CODE column in W_GL_ACCOUNT_D.
- In W_GL_BALANCE_A (where you store GL account balances at aggregated level), you want to store GL account balances at Company and Cost Center level instead of at GL Code Combination level.

The screenshot below shows how the file_glacct_segment_config_ora.csv would be configured to implement the business requirements specified above.

Figure 5–3 Screenshot of file_glacct_segment_config_ora.csv in text editor



	A	B	C	D	E	F	G	H	I
1	CHART OF ACCOUNTS_	SEG1	SEG1_VALUESETID	SEG2	SEG2_VALUESETID	SEG3	SEG3_VALUESETID	SEG4	SEG4_VALUESETID
2	4256	SEGMENT3	1002471	SEGMENT6	1002472	SEGMENT4	1002470	SEGMENT1	1002478
3	4257	SEGMENT1	1002702	SEGMENT2	1002472	SEGMENT3	1002477	SEGMENT4	1002479
4	AGGREGATION	Y		Y					

5.1.4.2.3 How To Set Up Hierarchies With General Ledger Accounting flexfields To set up hierarchies with General Ledger Accounting flexfields:

1. Use the $ODI_HOME\biapps_odi\odifiles\odidatafiles\srcfiles\file_glacct_segment_configur_ora.csv file to specify the segments that you want to analyze.
2. Do the following:

   a. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").

   b. Select the **Administer ELT Parameters** link.

   c. Display the Global tab.

   d. Set the value of the LOAD_VALUESET_HIERARCHY parameter to 'Y'.

   e. Set the value of the LOAD_FSG_HIERARCHY parameter is set to 'N'.

   f. Save your changes

   For more information about specifying parameter values, see Section 4.6.3.3.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

3. Run the ELT process for General Ledger Accounts.
4. Using the Oracle BI Administration Tool, in the RPD Physical Layer, create additional aliases or change the names of the existing alias against the table W_HIERARCHY_D.

   For example, if the existing aliases are numbered 1 to 3, name the new alias Dim_W_HIERARCHY_D_ValueSetHierarchy4, and rename the existing aliases as described in the table below:

Old alias name	New alias name
Dim_W_HIERARCHY_D_ValueSetHierarchy1	Dim_Company_Hierarchy_D
Dim_W_HIERARCHY_D_ValueSetHierarchy2	Dim_CostCenter_Hierarchy_D
Dim_W_HIERARCHY_D_ValueSetHierarchy3	Dim_NaturalAccount_Hierarchy_D

5. Create a new alias against W_HIERARCHY_D and name the new alias Dim_LOB_Hierarchy_D.
6. Using the Oracle BI Administration Tool, in the Physical Layer, create joins in the physical layer from the new aliases that you created in the previous step, as follows:
   – Company Hierarchy will join to the segment column in W_GL_ACCOUNT_D that stores the Company segment.
     * Dim_Company_Hierarchy_D. HIER20_CODE = W_GL_ACCOUNT_D.ACCOUNT_SEG1_CODE
     * Dim_Compnay_Hierarchy_D.HIER_CODE = W_GL_ACCOUNT_D.ACCOUNT_SEG1_ATTRIB
   – Cost Center Hierarchy will join to the segment column in W_GL_ACCOUNT_D that stores the Cost Center segment.
     * Dim_CostCenter_Hierarchy_D. HIER20_CODE = W_GL_ACCOUNT_D.ACCOUNT_SEG2_CODE
     * Dim_CostCenter_Hierarchy_D.HIER_CODE = W_GL_ACCOUNT_D.ACCOUNT_SEG2_ATTRIB
   – Natural Account Hierarchy will join to the segment column in W_GL_ACCOUNT_D, that stores the Natural Account segment.
     * Dim_NaturalAccount_Hierarchy_D. HIER20_CODE = W_GL_ACCOUNT_D.ACCOUNT_SEG3_CODE
     * Dim_NaturalAccount_Hierarchy_D.HIER_CODE = W_GL_ACCOUNT_D.ACCOUNT_SEG3_ATTRIB
   – LOB Hierarchy will join to the segment column in W_GL_ACCOUNT_D that stores the LOB segment.
     * Dim_LOB_Hierarchy_D. HIER20_CODE = W_GL_ACCOUNT_D.ACCOUNT_SEG4_CODE
     * Dim_LOB_Hierarchy_D.HIER_CODE = W_GL_ACCOUNT_D.ACCOUNT_SEG4_ATTRIB

Note: Hierarchies are linked to HIER20_CODE, because it is leaf node of the hierarchy

7. Join the aliases of W_HIERARCHY_D to the corresponding columns in the table W_GL_BALANCE_A.

   - Dim_Company_Hierarchy_D.HIER20_CODE = Fact_Agg_W_GL_BALANCE_A.ACCOUNT_SEG1_CODE AND
   - Dim_Company_Hierarchy_D.HIER_CODE = Fact_Agg_W_GL_BALANCE_A.ACCOUNT_SEG1_ATTRIB

   Note: W_GL_BALANCE_A has only 6 segment columns. So, if you have more than 6 hierarchies, join only the first 6 to W_GL_BALANCE_A but join all hierarchies to W_GL_ACCOUNT_D in the previous step.

8. A HIER_CODE filter should be specified in the Business Model Layer to restrain the output of logical table to be one Hierarchy only. You must set the HIER_CODE to be filtered by the Valueset IDs that are applicable for that particular hierarchy. The list of the valueset ids would be same as the valueset ids you configured in the csv file in the first set.

   To specify a filter in the Business Model Layer to restrain the output of logical table, do the following:

   a. Expand each logical table for the hierarchies, say Dim - GL ValueSetHierarchy1, and open the logical table source under it.
   b. Go to the Content tab.
   c. In the 'use WHERE clause…' text box, apply a filter on the corresponding physical table alias of W_HIERARCHY_D.

      For example:

      ```
      "Oracle Data Warehouse"."Catalog"."dbo"." Dim_W_HIERARCHY_D_
      ValueSetHierarchy1"."HIER_CODE" IN (<comma seperated valuesetids>)
      ```

9. A HIER_CODE filter should be specified in the Business Model Layer to restrain the output of the logical table to be one Hierarchy only. To specify a filter in the Business Model Layer to restrain the output of the logical table, do the following:

   a. Right click logical table Dim_W_Hierarchy_D_ValueSet1.
   b. Choose Properties.
   c. Choose the Source tab.
   d. Select 'Dim_W_Hierarchy_D_Valueset1'.
   e. Click Edit.
   f. Choose the Content tab.
   g. Insert the following code into the 'use WHERE clause…' text box:

      ```
      "Oracle Data Warehouse"."Catalog"."dbo"."Dim_W_Hierarchy_
      D_Valueset1"."HIER_CODE" = <Value Set Hierarchy Id>
      ```

      Where <*Value Set Hierarchy ID*> is the Value Set Hierarchy ID of the segment for which you are creating a hierarchy, for example, 1001.

10. Using the Oracle BI Administration Tool, in the Business Model Layer of the Oracle BI Analytics Warehouse, create additional hierarchies using the dimensions that you created in the previous step.

    For more information about example hierarchies that are installed with Oracle Business Intelligence Applications, refer to the following hierarchies in the Oracle Business Analytics Warehouse:

   – GL ValueSetHierarchy1
   – GL ValueSetHierarchy2
   – GL ValueSetHierarchy3

11. In the Business Model Layer, look for all the logical fact table that has logical join to the logical hierarchy table Dim - GL ValueSetHierarchy1. You will need to create a similar logical table between the new logical hierarchy dimensions that you created and these logical facts, as follows:

    a. Under each of the logical fact table, open the logical table sources and go to the content tab. Under aggregation content, select the **Show unmapped** check box. It will show all the hierarchies you created in the previous step. For each of these hierarchies, select the logical level to 'Detail'.

    b. In the business model diagram, create a new Complex Join between each of the new logical hierarchy tables and each of the logical facts. In the join, make sure the cardinality is (0,1) on the dimension side and N on the fact side.

12. Using the Oracle BI Administration Tool, in the Presentation Layer of the Oracle BI Analytics Warehouse, drag the new hierarchies into the presentation folder.

> **Note:** You can rename the hierarchies in the Presentation Layer if required.

5.1.4.3 Configuring General Ledger Account Hierarchies Using Financial Statement Generator (FSG) Report Definition

These configuration steps are required if you are deploying Oracle Financial Analytics, Oracle Procurement and Spend Analytics, or Oracle Supply Chain and Order Management Analytics. This section contains the following topics:

- Section 5.1.4.3.1, "Overview"
- Section 5.1.4.3.2, "About Configuring the ELT Process for Oracle Financial Statement Generator Report"
- Section 5.1.4.3.3, "How To Set Up Hierarchies With Financial Statement Generator Report Definition"

5.1.4.3.1 Overview If you need to define GL account hierarchies based on multiple segments within a chart of accounts, you can use the Financial Statement Generator report definition UI in Oracle EBS to define them.

You should first use the Oracle EBS Financial Statement Generator (FSG) form to define a row set or a column set, then Oracle Business Intelligence Applications will extract the row set or column set definition and convert them into hierarchies.

Oracle Financial Statement Generator hierarchies are extracted from following EBS source tables:

- RG_REPORT_AXIS_CONTENTS

  This table defines the relationship between the FSG report axis and GL code combinations. The GL code combinations with segment values within the value range defined for that axis are categorized as children of that axis.

- RG_REPORT_AXIS_SETS

  This table stores the information for each of the row set or column set you defined. There is one record in this table for each row or column set you defined. Each row includes an axis set identifier, a row or column set name, and a structure identifier to assign a specific chart of accounts to the row set or column set.

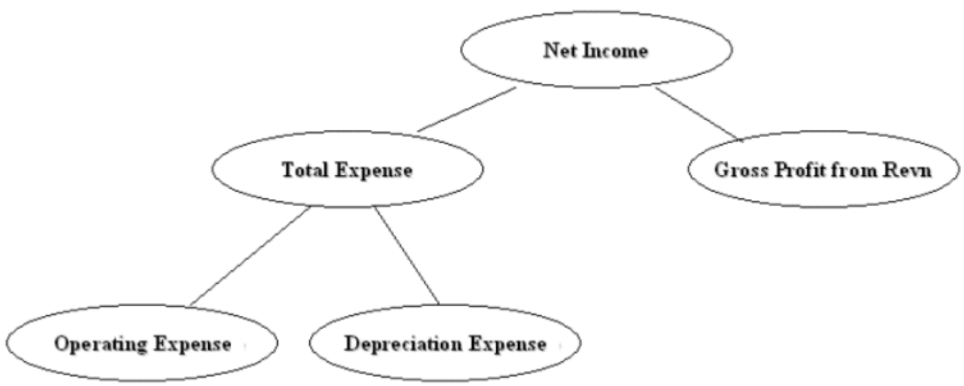
- RG_REPORT_CALCULATIONS

  This table stores formulas for calculating each row or column in the row or column set. An example of a row calculation might be to sum up the amount from the previous five rows. An example of a columns calculation might be to calculate column five by subtracting column four from column three.

For example, in Income Statement, 'Net Income' is the calculation result of 'Gross Profit from Revenue' minus 'Total Expense'. When converting to hierarchy, Net Income becomes the parent of 'Gross Profit from Revenue' and 'Total Expense'. Therefore, hierarchy can be derived based on the information in RG_REPORT_CALCULATION.

The diagram below shows an example hierarchy, with the top level Net Income node having two child nodes, Total Expense, and Gross Profit from Revn, and the Total Expense node having two child nodes, Operating Expense, and Depreciation Expense.

The diagram below shows how an income state is derived from a hierarchy.



The hierarchy above would be converted into a flattened hierarchy and stored in W_HIERARCHY_D in the following format:

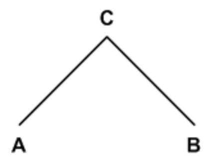
Table 5–4 Example flattened hierarchy stored in W_HIERARCHY_D

HIER Name	HIER1	HIER2	HIER3	HIER4	HIER20
Income Statement	Net Income	Gross Profit...	Gross Profit...	Gross Profit...	Gross Profit...
Income Statement	Net Income	Total Expenses	Operating Expenses	Operating Expenses	Operating Expenses
Income Statement	Net Income	Total Expenses	Depreciation Expense	Depreciation Expense	Depreciation Expense

Fact tables join to the W_HIERARCHY_D table via the General Ledger Account dimension table (W_GL_ACCOUNT_D).

The W_GL_ACCOUNT_D table contains six fields (HIER1_WID, HIER2_WID, HIER3_WID, ..., HIER6_WID), which are foreign keys to the W_HIERARCHY_D.row_wid. Therefore, each General Ledger Code combination can participate in up to six different hierarchies. You can decide which of the six hierarchies to drill on based on the column you use to join to W_HIERARCHY_D. For example, if you want to drill using the third hierarchy, you use W_GL_ACCOUNT_D.hier3_wid = W_HIERARCHY_D.row_wid.

Note: Mathematical operators, such as '+', '-', '*', '/' (addition, subtraction, multiplication, division, etc) are not extracted from the FSG definitions. For example, both A + B = C and A - B = C would give the same hierarchy, with a node C having two child nodes A and B (see diagram below).

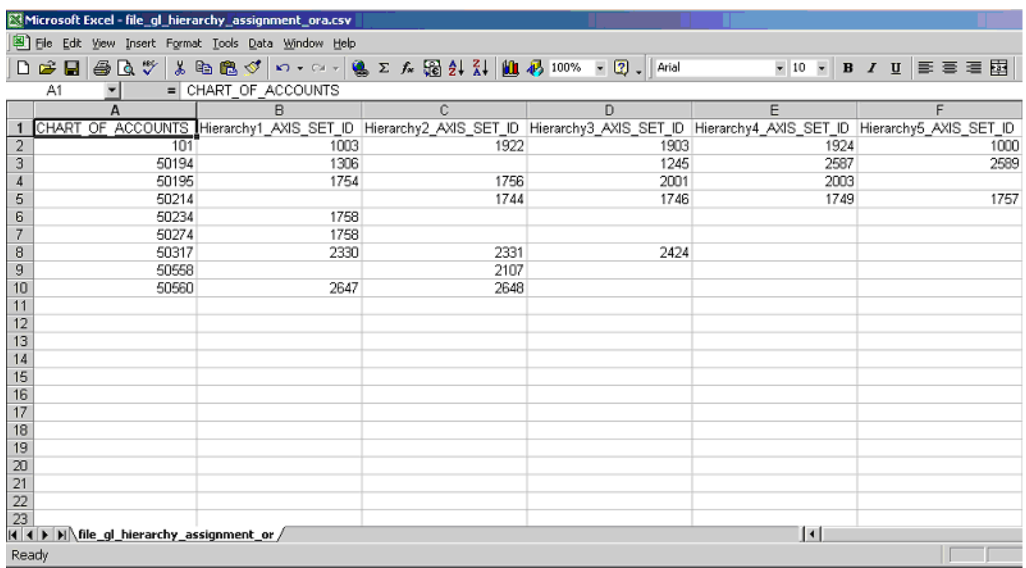


5.1.4.3.2 About Configuring the ELT Process for Oracle Financial Statement Generator Report

Before you run the ELT process for General Ledger Accounts, you need to specify the hierarchies that you want to reference. To specify the FSG hierarchies that you want to reference, use the following ELT configuration file:

$ODI_HOME\biapps_odi\odifiles\odidatafiles\srcfiles\file_gl_hierarchy_assignment_ora.csv

Figure 5–4 Screen Shot of file_gl_hierarchy_assignment_ora.csv File Opened in a Spreadsheet Editor



CHART_OF_ACCOUNTS	Hierarchy1_AXIS_SET_ID	Hierarchy2_AXIS_SET_ID	Hierarchy3_AXIS_SET_ID	Hierarchy4_AXIS_SET_ID	Hierarchy5_AXIS_SET_ID
101	1003	1922	1903	1924	1000
50194	1306		1245	2587	2589
50195	1754	1756	2001	2003	
50214		1744	1746	1749	1757
50234	1758				
50274	1758				
50317	2330	2331	2424		
50558		2107			
50560	2647	2648			

In this file, for each chart of accounts, you can specify six FSG hierarchies (using axis_set_id; Axis_set_id is a column from RG_REPORT_AXIS_SETS table. It is the unique ID of a row set or column set) you want to store in the GL account dimension table for the code combinations that belong to that chart of accounts.

The DATASOURCE_NUM_ID field specifies the data source to which the configurations apply. If you have multiple source systems, there might be a chart of accounts across the multiple source systems with the same ID. Therefore, you need to use the DATASOURCE_NUM_ID value to distinguish between them.

For example, we have an Income Statement FSG report and a Balance Sheet FSG report which we want to derive both of their hierarchy structures into OLAP.

Oracle Business Intelligence Applications assumes that both reports are derived from same set of GL accounts with CHART_OF_ACCOUNTS=101. The axis_set_id of Income Statement is 1001, and for Balance Sheet, it is 1003. The DATASOURCE_NUM_ID for this application is 2.

In addition, for those GL ACCOUNTS that belong to the two reports (by which we used to calculate the reports), we want to associate their HIER1 column (in GL_ACCOUNT_D) with Income Statement hierarchy structure, HIER3 column with Balance Sheet hierarchy structure.

In this case, we would add one row into file_gl_hierarchy_assignment_ora.csv with the following fields set as below:

CHART OF ACCOUNTS - 101

HIER1_AXIS_SET_ID - 1001

HIER3_AXIS_SET_ID - 1003

DATASOURCE_NUM_ID - 2

(Leave the other row values blank.)

This row indicates that for all of the GL Accounts with CHART_OF_ACCOUNTS=101 and DATASOURCE_NUM_ID=2, assigning hierarchies with axis_set_id=1001, null, 1003, null, null, null to HIER1~HIER6 columns respectively. Therefore, after extraction and loading, for those affected GL Account rows, HIER1 column will be the foreign key to Income Statement hierarchy's row ID in W_HIERARCHY_D and HIER3 column will be the foreign key to Balance Sheet hierarchy's row ID in W_HIERARCHY_D.

Note: Financial Analytics will not load those hierarchies with axis_set_id not been specified in file_gl_hierarchy_assignment_ora.csv.

5.1.4.3.3 How To Set Up Hierarchies With Financial Statement Generator Report Definition To set up hierarchies with Financial Statement Generator Report Definition:

1. Use the $ODI_HOME\biapps_odi\odifiles\odidatafiles\srcfiles\file_gl_hierarchy_assignment_ora.csv file to specify the hierarchies you want to reference for each CHART_OF_ACCOUNTS.
2. Do the following:
   a. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
   b. Select the **Administer ELT Parameters** link.
   c. Display the Global tab.
   d. Set the value of the LOAD_VALUESET_HIERARCHY parameter to 'N'.
   e. Set the value of the LOAD_FSG_HIERARCHY parameter is set to 'Y'.
   f. Save your changes

      For more information about specifying parameter values, see Section 4.6.3.3.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".
3. Run the ELT process for General Ledger Accounts by doing one of the following:

- If you have already created an Execution plan for General Ledger Accounts in Oracle BI Applications Configuration Manager, run this execution plan.

  For more information about how to create an Execution Plan, see Section 4.7.2.2, "How to Create an Execution Plan In Oracle BI Applications Configuration Manager").

- Use ODI Designer to perform an ELT process. for General Ledger Accounts.

  For more information about how to perform an ELT process in ODI Designer, see Section 4.7.3, "How to perform E-LT Using ODI Designer").

4. Using the Oracle BI Administration Tool, in the RPD Physical Layer, create additional aliases or change the names of the existing alias against the table W_HIERARCHY_D.

   For example, if you want to create an income statement hierarchy, create an additional alias Dim_IncomeStatement_FSGHierarchy_D against the table W_HIERARCHY_D.

5. Using the Oracle BI Administration Tool, in the Physical Layer of the Oracle BI Analytics Warehouse, create joins in the physical layer from the new aliases that you created in the previous step, as follows:

   a. Income Statement Hierarchy will join to one of the HIER1~6 columns that you have specified in file file_gl_hierarchy_assignment_ora.csv for Income Statement.

   b. In this case, we join it to HIER1 column.Dim_W_GL_ACCOUNT_D.HIER1_WID = Dim_IncomeStatement_FSGHierarchy_D.ROW_WID

6. Using the Oracle BI Administration Tool, in the Business Model Layer of the Oracle BI Analytics Warehouse, create additional dimensions using the new alias.

   For the Income Statement hierarchy case, we create a new logical table Dim_IncomeStatement_FSGHierarchy_D, choose Dim_IncomeStatement_FSGHierarchy_D in physical layer as source. Mapping ROW_WID, HIER_CODE, and HIER1~HIER20 (both name and code) from physical table to logical key.

   Then, set HIER_CODE=1001 (this is the Axis_set_id of Income Statement hierarchy) in logical table to restrain the output of logical table to be Income Statement Hierarchy only (right click logical table Dim_IncomeStatement_FSGHierarchy_D – click properties – choose Source tab – select Dim_IncomeStatement_FSGHierarchy_D – click Edit button – choose Content tab – fill ("Oracle Data Warehouse"."Catalog"."dbo"."Dim_W_HIERARCHY_D_FSG1"."HIER_CODE" =1001) into 'use WHERE clause…' text box).

   For more information about this process, please refer to the following pre-installed example logical table Dim - FSG Hierarchy 1 in the Oracle Business Analytics Warehouse

7. Using the Oracle BI Administration Tool, in the Business Model Layer of the Oracle BI Analytics Warehouse, create new dimension based on the logical table that you created in the previous step.

   Please refer to 'FSG Hierarchy 1' as an example.

8. In the Business Model Layer, look for all the logical fact table that has logical join to the logical hierarchy table Dim - FSG Hierarchy1. You will need to create a similar logical join between the new logical hierarchy dimensions that you created and these logical facts, as follows:

a. Under each of the logical fact table, open the logical table sources and go to the content tab. Under aggregation content, select the **Show unmapped** check box. It will show all the hierarchies you created in the previous step. For each of these hierarchies, select the logical level to 'Detail'.

b. In the business model diagram, create a new Complex Join between each of the new logical hierarchy tables and each of the logical facts. In the join, make sure the cardinality is (0,1) on the dimension side and N on the fact side.

9. Using the Oracle BI Administration Tool, in the Presentation Layer of the Oracle BI Analytics Warehouse, drag the new hierarchies into the presentation folder.

   Note: You can rename the hierarchies in the Presentation Layer if required.

5.1.5 Configuring Product Hierarchy, Master Inventory Organization, and Group Accounts

This section explains how to configure Product Hierarchy, Master Inventory Organization, and Group Accounts, and contains the following topics.

- Section 5.1.5.1, "Configuration of Product Hierarchy (except for GL, HR modules)"
- Section 5.1.5.2, "Configure the Master Inventory Organization in Product dimension Extract for Oracle 11i adapter (Except for GL & HR Modules)"
- Section 5.1.5.3, "How to Map Oracle General Ledger Natural Accounts to Group Account Numbers"
- Section 5.1.5.4, "How to do Group Account correction for Oracle Application"

5.1.5.1 Configuration of Product Hierarchy (except for GL, HR modules)

This section contains configuration points for product hierarchy in the Product dimension table and the Inventory Product dimension table.

Oracle E-Business Suite enables users to groups the products using categories and category sets. While E-Business Suite provides pre-packaged category sets, you can also define your own category sets.

As configured out-of-the-box, the Oracle Business Analytics Warehouse extracts product categories where the CATEGORY_SET_ID is 2 or 27. However, it is likely that the categories you extract from the source system are different from these prepackaged category sets.

To configure your product category sets, do the following:

- Identify the category sets you want to report (for more information, see Section 5.1.5.1.1, "How to identify Category Sets from EBS").
- Configure the parameters for product hierarchy (for more information, see Section 5.1.5.1.2, "Configure the ODI Parameters for Product Hierarchy").
- Configure the hierarchy mapping with the segments (for more information, see Section 5.1.5.1.3, "Configure the Hierarchy Mapping With the Segments").

5.1.5.1.1 How to identify Category Sets from EBS These steps are part of the task in Section 5.1.5.1, "Configuration of Product Hierarchy (except for GL, HR modules)".

To find out the category sets that your organization is using:

1. Log in to the EBS instance.
2. Navigate to Setup > Items > Categories > Default Category Sets.

3. Look for functional Area Inventory and place the cursor in the **Category Set** field.
4. Choose Help, then Diagnostics, then Examine, and specify the apps user password.
5. Click the Field LOV button and select CATEGORY_SET_ID, and note down the value.
6. Repeat steps 3-5 for the Purchasing functional area.

5.1.5.1.2 Configure the ODI Parameters for Product Hierarchy These steps are part of the task in Section 5.1.5.1, "Configuration of Product Hierarchy (except for GL, HR modules)".

To configure ODI Parameters for product hierarchy:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Common tab.
4. Locate the following parameters and use the **Parameter Value** field to set the value:
   - INVPROD_CAT_SET_ID

     Use the CATEGORY_SET_ID value that you returned in step 5 in the task Section 5.1.5.1.1, "How to identify Category Sets from EBS".
   - PROD_CAT_SET_ID

     Use the CATEGORY_SET_ID value that you returned in step 6 in the task Section 5.1.5.1.1, "How to identify Category Sets from EBS".
5. Save your changes.

Note: The grain of the Product dimension is at the Master level. Therefore, the category set chosen as a value for the Product Dimension parameter (PROD_CAT_SET_ID) must be a Category Set controlled at a Master Level but not at the Org level.

5.1.5.1.3 Configure the Hierarchy Mapping With the Segments These steps are part of the task in Section 5.1.5.1, "Configuration of Product Hierarchy (except for GL, HR modules)".

To configure the hierarchy mapping with the segments:

1. In ODI Designer, open the Conversion folder.
2. Select the folder 'SDE_ORA11510_Adaptor', then the 'Packages' folder
3. In the Packages folder locate the package 'SDE_ORA_ProductDimension', and edit the interfaces 'Run PRODUCT_DS' and 'Run PRODUCT_DS_FULL', to edit the mapping of the SEGMENTS to the HIER columns.
4. In the Target Datastore object, scroll down to find the hierarchy code ports named PROD_HIER<X>_CODE.

   Hierarchy levels are named with the convention PROD_HIER<X>_CODE, where <X> denotes the level within the hierarchy.

For example, if you want to edit the first level of your hierarchy, you must edit the definition for PROD_ HIER1_CODE port.

The first level is pre-configured as follows:

```
PROD_HIER1_CODE = IIF(
ISNULL(SQ_W_ORA_PRODUCT_DS_TMP.PROD_HIER1_CODE)
OR ISNULL(SQ_W_ORA_PRODUCT_DS_TMP.PROD_HIER2_CODE), NULL,
SQ_W_ORA_PRODUCT_DS_TMP.PROD_HIER1_CODE||'~'||
SQ_W_ORA_PRODUCT_DS_TMP.PROD_HIER2_CODE )
```

The second level is pre-configured as follows:

```
PROD_HIER2_CODE = SQ_W_ORA_PRODUCT_DS_TMP.PROD_HIER1_CODE
```

5. Click the Expression field for the appropriate PROD_HIER<X>_CODE port, then in the Expression Editor, modify the expression.
6. Apply and save your changes.

Note: The concatenation of the Segments at level 1 is done to make the CODE column unique with the assumption of the Segment2 is a Dependent Segment on the parent Segment1.

5.1.5.2 Configure the Master Inventory Organization in Product dimension Extract for Oracle 11i adapter (Except for GL & HR Modules)

In Oracle 11i applications, the products are defined in a Master Organization and then copied into the other Inventory Organizations for transactions. The Product dimension Extract mapping 'SDE_ORA_ProductDimension_Derive' has been enabled for configuration of this Master Organization based on the configuration in the OLTP. As installed out-of-the-box, the organization ID (that is set by the $$MASTER_ORG parameter) is set to 204. This organization ID 204 needs to be changed based on the individual implementation of OLTP in your deployment.

Note: This E-LT implementation supports the best practice prescribed by Oracle for the creation of Single Master Organization for defining the Product master. This E-LT implementation does not support the multiple master Organizations if the same product is defined in multiple master organizations.

To set the Master Inventory Organization in Product dimension Extract:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Common tab.
4. Locate the following parameter and use the **Parameter Value** field to set the value:
   - MASTER_ORG

     For example, MASTER_ORG=204.
5. Save your changes.

5.1.5.3 How to Map Oracle General Ledger Natural Accounts to Group Account Numbers

> **Note:** It is critical that the General Ledger Account Numbers are mapped to the Group Account Numbers (or domain values) as the metrics in the General Ledger reporting layer uses these values. For a list of domain values for General Ledger Account Numbers, see Oracle Business Analytics Warehouse Data Model Reference.

You can categorize your Oracle General Ledger accounts into specific group account numbers. The group account number is used during data extraction as well as front-end reporting. The GROUP_ACCT_NUM field in the GL account dimension table W_GL_ACCOUNT_D denotes the nature the nature of the General Ledger accounts (for example, cash account, payroll account). Refer to the master_code column in the `file_group_acct_names_ora.csv` file for values you can use. For a list of the Group Account Number domain values, see Oracle Business Analytics Warehouse Data Model Reference. The mappings to General Ledger Accounts Numbers are important for both Profitability analysis and General Ledger analysis (for example, Balance Sheets).

The logic for assigning the accounts is located in the `file_group_acct_codes_ora.csv` file. The table below shows an example configuration of the `file_group_acct_codes_ora.csv` file.

Table 5–5 Example configuration of file_group_acct_codes_ora.csv

COA ID	FROM ACCT	TO ACCT	GROUP_ACCT_NUM
1	101010	101099	CA
1	131010	131939	FG INV
1	152121	152401	RM INV
1	171101	171901	WIP INV
1	173001	173001	PPE
1	240100	240120	ACC DEPCN
1	261000	261100	INT EXP
1	181011	181918	CASH
1	251100	251120	ST BORR

In the table above, in the first row, all accounts within the account number range from 101010 to 101099 that have a Chart of Account (COA) ID equal to 1 are assigned to Current Asset. Each row maps all accounts within the specified account number range and within the given chart of account ID.

If you need to create a new group of account numbers, you can create new rows in the `file_group_acct_names_ora.csv` file. You can then assign GL accounts to the new group of account numbers in the `file_group_acct_codes_ora11i.csv` file.

You also need to add a new row in the file_grpact_fstmt.csv file. This file specifies the relationship between a Group Account Number and a Financial Statement Item Code. You must map the new Group Account Number to one of the following Financial Statement Item codes:

- AP

- AR
- COGS
- REVENUE
- TAX
- OTHERS

These Financial Statement Item codes correspond to the following six base fact tables in the Financial Analytics product.

- AP base fact (W_AP_XACT_F)
- AR base fact (W_AR_XACT_F)
- Revenue base fact (W_GL_REVN_F)
- Cost of Goods Sold base fact (W_GL_COGS_F)
- Tax base fact (W_TAX_XACT_F)
- GL Journal base fact (W_GL_OTHER_F)

By mapping your GL accounts against the Group Account Numbers and then associating the Group Account Number to a Financial Statement Item code, you have indirectly associated the GL account numbers to Financial Statement Item codes as well.

Financial Statement Item codes are internal codes used by E-LT process to process the GL journal records during the GL reconciliation process against the sub-ledgers. When the E-LT process reconciles a GL journal record, it looks at the Financial Statement Item code associated with the GL account that the journal is charging against, and then uses the value of the Financial Statement item code to decide which base fact the GL journal should reconcile against. For example, when processing a GL journal that charges to a GL account which is associate to 'AP' Financial Statement Item code, then the E-LT process will try to go against AP base fact table (W_AP_XACT_F), and try to locate the corresponding matching AP accounting entry. If that GL account is associated with the 'REVENUE' Financial Statement Item code, then ODI will try to go against the Revenue base fact table (W_GL_REVN_F), and try to locate the corresponding matching Revenue accounting entry.

> **Note:** When you specify the Group Account Number, you must capitalize the letters and use the values in the `master_code` column of the `file_group_acct_names_ora.csv` file.

To map Oracle General Ledger account numbers to Group Account Numbers:

1. Open the `file_group_acct_codes_ora.csv` file with a text editor in the `$ODI_HOME\biapps_odi\odifiles\odidatafiles\srcfiles` directory.
2. Edit the fields in the following table:

Field Name	Description
COA ID	The ID of the General Ledger chart of account.
FROM ACCT and TO ACCT	The natural account range. This is based on the natural account segment of your GL accounts.

Field Name	Description
GROUP_ACCT_NUM	This field denotes the nature of the Oracle General Ledger accounts. For example, Cash account, Payroll account, and so on. Refer to the file_group_acct_names_ora.csv file for values you can use.

Note: It is important that you do not edit any other fields in the CSV files.

3. Save and close the CSV file.

5.1.5.4 How to do Group Account correction for Oracle Application

Note: Refer to the section Section 5.1.5.3, "How to Map Oracle General Ledger Natural Accounts to Group Account Numbers" for general concepts about Group Account Number and Financial Statement Item Code.

When a user mistakenly maps a GL natural account number to an incorrect Group Account Number, incorrect accounting entries might be inserted into the fact table. For example, natural account 1210 is classified as belonging to 'AR' Group Account Number when it should be classified as having 'AP' Group Account Number. When this happens, ODI will get all the journal lines charging to account 100 and try to reconcile these journal lines against sub-ledger accounting records in the AR fact table (W_AR_XACT_F). Since these journal lines did not come from AR, ODI will not be able to find the corresponding matching sub-ledger accounting records for these journal lines. In this case, ODI will insert 'Manual' records into the AR fact table because it thinks that these GL journal lines are 'Manual' journal entries created directly in GL system charging against the AR accounts. This entire process is call GL Reconciliation process.

In order to revert these 'Manual' entries in the AR fact, you will need to utilize the 'Group Account Number Cleanup' program provided in Oracle Business Intelligence Applications. This program will revert the 'Manual' entries in the fact table (in this case, AR fact table); and then try to do the GL reconciliation process again. This time, ODI will try to look for the corresponding matching sub-ledger accounting records in the AP fact (W_AP_XACT_F); provided that you've re-assign the natural account 1210 to the 'AP' group account number in the file_group_acct_codes_ora.csv file.

To do Group Account correction:

1. Open the file file_group_acct_codes_ora.csv with a text editor in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\srcfiles directory.
2. Change the mapping of the GL Natural account to the group account in the CSV file.

   For example, before the change, the CSV file might have the following values:

   CHART OF ACCOUNTS ID = 101

   FROM ACCT = 1110

   TO ACCT = 1110

   GROUP_ACCT_NUM = CASH

   After the change, if the account '1210' originally belonged to the 'AP' Group Acct Num and after correcting the GL Natural account to the group account, the CSV file would have the following values:

CHART OF ACCOUNTS ID = 101

FROM ACCT = 1210

TO ACCT = 1210

GROUP_ACCT_UM = AR

3. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
4. Select the **Administer ELT Parameters** link.
5. Display the Global tab.
6. Locate the following parameters and use the **Parameter Value** field to set the value:
   - LOAD_CALCULATE_GL_BALANCE

     Set this value to 'N'.

     Note: If 'LOAD_CALCULATE_GL_BALANCE' is set to 'Y', then the value of LOAD_EXTRACT_GL_BALANCE cannot be set to 'Y'.
7. Save your changes.
8. Run the corresponding Group Account Cleanup execution plan named ORA11510 Financials – Group Account Number Clean Up R12.

5.1.6 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps that apply to Oracle Business Intelligence Applications, and contains the following topics:

- Section 5.1.6.1, "How to Configure Data Source Num IDs"
- Section 5.1.6.2, "Configuring the Names of Country Region, State Region, State, or Region"
- Section 5.1.6.3, "Configuring the Configuring the Make-Buy Indicator"

5.1.6.1 How to Configure Data Source Num IDs

DATASOURCE_NUM_ID is a system column in the data warehouse that uniquely identifies a data source category and indicates which source systems the data comes from.

To configure a DATASOURCE_NUM_ID value:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Global tab.
4. Locate the following parameter and use the **Parameter Value** field to set the value:
   - WH_DATASOURCE_NUM_ID
5. Save your changes.

   For more information about specifying parameter values, see Section 4.6.3.3.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

5.1.6.2 Configuring the Names of Country Region, State Region, State, or Region

For Oracle 11i, you can reconfigure the names of the Country Region, State Region, State, and Region. This configuration information applies only to plant, storage, and supplier locations. By default, the Region Name column (EXT_REGION_NAME) is populated using the same code value as the Region Code column (EXT_REGION_CODE).

To configure the names of Country Region, State Region, State, or Region:

1. In ODI Designer, open the folder SDE_ORA11510_adapter.
2. Select the Interface you want to edit.

   The following is a list shows Interfaces that use the COUNTRY_REGION, COUNTRY_NAME, STATE_NAME and STATE_REGION columns:

   - SDE_ORA_SupplierDimension.SUPPLIER_DS
   - SDE_ORA_SupplierDimension_Full.SUPPLIER_DS_Full
   - SDE_ORA_BusinessLocationDimension_Plant.BUSN_LOCATION_DS
   - SDE_ORA_BusinessLocationDimension_Plant_Full.BUSN_LOCATION_DS_Full
   - SDE_ORA_BusinessLocationDimension_StorageLocation.BUSN_LOCATION_DS
   - SDE_ORA_BusinessLocationDimension_StorageLocation_Full.BUSN_LOCATION_DS_Full
3. Make sure that the name columns do not map to any source columns, as follows:

   a. Double-click the Interface to display the Interface: <Name> dialog.

   b. Display the Diagram tab.

   c. In the Target Datastore area, select the column name.

   d. In the Mapping: <Name> pane below, display the Implementation tab and delete any text in this tab.

   Deleting the text in the Implementation tab enables the lookup to resolve the Names from the W_CODE_D table for the corresponding CODE.
4. Save the changes.
5. Regenerate the SCENARIOS for the Packages that were using the Interfaces that you edited.

5.1.6.3 Configuring the Configuring the Make-Buy Indicator

The Make-Buy indicator specifies whether a material that was used to manufacture a product was made in-house or bought from an outside vendor. By default, the indicator is set using the INP_PLANNING_MAKE_BUY_CODE. If the code is set to 1, then the indicator is set to M (for make). However, if the code is set to 2, then the indicator is set to B (for buy). Otherwise, the indicator is set to null.

Your organization may require different indicator codes. If so, you can modify the indicator logic by reconfiguring the condition in the Interface mplt_SA_ORA_ProductDimension. For example, you may want your indicator code to be 0 for make, and 1 for buy.

To configure the Make-Buy Indicator:

1. In ODI Designer, open the folder SDE_ORA11510_adapter.

2. Select the Interface you want to edit.

   The following is a list shows Interfaces that use the Make_Buy_Ind column:

   - SDE_ORA_ProductDimension_Derive.ORA_PRODUCT_DS_TMP
   - SDE_ORA_ProductDimension_Derive_Full.ORA_PRODUCT_DS_TMP_FULL

3. Edit the Make_Buy_Ind columns, as follows:

   a. Double-click the Interface to display the Interface: <Name> dialog.

   b. Display the Diagram tab.

   c. In the Target Datastore area, select the column name.

   d. In the Mapping: <Name> pane below, display the Implementation tab and modify the text in this tab.

4. Save the changes.

5. Regenerate the SCENARIOS for the Packages that were using the Interfaces that you edited.

6 Configuring Oracle Procurement and Spend Analytics

This chapter describes how to configure Oracle Procurement and Spend Analytics for particular sources to meet your business needs, and contains the following topics:

- Section 6.1, "Overview of Oracle Procurement and Spend Analytics"
- Section 6.2, "Configuration Required Before A Full Load for Oracle Procurement and Spend Analytics"

To find out about other possible tasks required to deploy Oracle Business Intelligence Applications, see Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle Business Intelligence Applications With ODI".

6.1 Overview of Oracle Procurement and Spend Analytics

Oracle Procurement and Spend Analytics comprises the following:

- Oracle Procurement and Spend Analytics (for more information, see Section 6.1.1, "Oracle Procurement and Spend Analytics Module").
- Oracle Supplier Performance Analytics (for more information, see Section 6.1.2, "Supplier Performance Analytics Module").

Oracle Procurement and Spend Analytics enable organizations to optimize their supply chain performance by integrating data from across the enterprise supply chain and enabling executives, managers, and frontline employees to make more informed and actionable decisions. Organizations using Oracle Procurement and Spend Analytics benefit from increased visibility into the complete Procurement and Spend process, including comprehensive supplier performance analysis and supplier payables analysis. Through complete end-to-end insight into the factors that impact Procurement and Spend performance, organizations can significantly reduce costs, enhance profitability, increase customer satisfaction, and gain competitive advantage. Oracle Procurement and Spend Analytics also integrate with the other applications in Oracle Business Intelligence Applications product line. They deliver this insight across the organization to increase the company's effectiveness in managing its customers, suppliers, and financial decisions.

6.1.1 Oracle Procurement and Spend Analytics Module

Provides complete visibility into direct and indirect spend across the enterprise, payment, and employee expenses. Example analyses are spend by Commodity & Supplier, by Purchase Org, Cost Center, and expense by Employee, Buyer, etc.

The Oracle Procurement and Spend Analytics application is comprised of these subject areas:

- Total Spend: This is a summary subject area that provides the ability to do comparative analysis and report on requested spend, committed spend and actual spend across suppliers, company, products, commodities and associated hierarchies for both direct and indirect spend (indirect spend being MRO and employee expenses) in detail to allow complete visibility of spending across your organization.
- Purchase Orders: This is a detailed subject area that provides the ability to report on committed spend, and Purchase orders of the suppliers of an organization across suppliers, company, products, commodities and associated hierarchies at purchase order line level
- Purchase Order Costs: This is a detailed subject area that provides the ability to report on committed spend and Purchase orders of the suppliers of an organization across suppliers, company, products, and commodities and associated hierarchies at cost center (distribution line) level.
- Purchase Cycle Lines: This is a summary subject area that provides the ability to report cycle time performance such as Requisition to PO lead time, PO to Receipt lead time, P2P lead time of the Suppliers of an organization.
- Purchase Schedules: This is a detailed subject area that provides the ability to report on purchase order shipments of an organization across suppliers, company, products, commodities and associated hierarchies at purchase schedule line level
- Purchase Requisitions: This is a detailed subject area that provides the ability to report on requested spend and Purchase requisitions of the suppliers of an organization across suppliers, company, products, commodities and associated hierarchies at purchase requisition line level
- Purchase Requisition Status: This is a summary subject area that provides the ability to report on requisition status along the approval cycle of Purchase requisitions of the suppliers of an organization. It's populated only by Universal adapter.
- Purchase Receipts: This is a detailed subject area that provides the ability to report on actual spend and Purchase Receipts of the suppliers of an organization across suppliers, company, location, products, commodities and associated hierarchies at purchase receipt line level
- Employee Spend: This is a detailed subject area that provides the ability to report on employee spend of an organization across employees, company, cost center and associated hierarchies. The Expenses subject area contains targeted metrics and reports that examine travel and expense costs in relationship to your organization's overall spending patterns. In contrast to analyzing direct spending patterns, where you may review purchasing, Expenses examines indirect spending—the cost of employee related expenses. It's populated only by Universal adapter.

6.1.2 Supplier Performance Analytics Module

Enables organizations to have a complete picture of the performance of their suppliers, including complete supplier scorecards, procurement cycle times, supplier price performance, delivery performance, product receipt quality, on-time payment ratings, payment activity and volume and payments due / overdue analysis.

The Supplier Performance Analytics application is comprised of these subject areas:

- Supplier Performance. The Suppliers functional area contains targeted reports and metrics that allow you to analyze the timeliness, reliability, cost, and quality of goods provided by your suppliers. It helps you to understand how well suppliers are contributing to success of your organization, and to evaluate the price, quality, and delivery timing in procuring materials
- Supplier AP Transactions: This is a summary subject area that provides the ability to analyze payment performance and payment due analysis of the suppliers of an organization across suppliers, company, location, products, commodities and associated hierarchies. In addition to monitoring supplier performance, it is important to monitor organization's performance of making on time payments. This will help the Organizations to maintain better relationships with their best suppliers.

6.2 Configuration Required Before A Full Load for Oracle Procurement and Spend Analytics

This section contains configuration steps that you need to perform on Oracle Procurement and Spend Analytics before you do a full data load, and contains the following topics:

- Section 6.2.1, "How to Deploy Stored Procedures"
- Section 6.2.2, "How to Configure the Parameter for Purchase Cycle Line"
- Section 6.2.3, "Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics"
- Section 6.2.4, "Configuration Steps for Controlling Your Data Set"

6.2.1 How to Deploy Stored Procedures

Stored procedures are a group of SQL statements that perform particular tasks on the database. For example, stored procedures can help to improve the performance of the database.

You can deploy stored procedures by copying the stored procedure files from your Oracle Business Intelligence installation and deploying them to the target data warehouse.

Note: Some sessions may fail if these procedures are not compiled in the database before running the workflows.

To deploy stored procedures:

1. On the ODI machine, navigate to the `$ODI_HOME\biapps_odi\odifiles\odidatafiles\storedprocfiles` directory, and locate the files 1_OFS_BI_TIME_PACKAGE.sql and 2_OFS_BI_LOAD_PACKAGE.sql.
2. Copy the SQL commands in the 1_OFS_BI_TIME_PACKAGE.sql file or the 2_OFS_BI_LOAD_PACKAGE.sql file as required.
3. Compile the stored procedures in the target data warehouse database.

Note: If you have problems deploying the stored procedures, see your database reference guide, or contact your database administrator.

6.2.2 How to Configure the Parameter for Purchase Cycle Line

To load the purchase cycle line table (W_PURCH_CYCLE_LINE_F), the ELT tasks need to distinguish data that originates in Oracle 11i applications.

To configure the parameter for Purchase Cycle Line:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Application Specific tab, and select Supply Chain Analytics from the **Select BI Application** drop down list.
4. Locate the following parameters and use the **Parameter Value** field to set the value:
   - ORA_DATASOURCE_NUM_ID_LIST

     Change the values of parameter ORA_DATASOURCE_NUM_ID_LIST from 4 (the default value) to the list of Data Source NUM IDs that you defined for your Oracle data sources.
   - SIL_PURCHASECYCLELINESFACT
   - SIL_PURCHASECYCLELINESFACT_EXTRACT
5. Save your changes.

   For more information about specifying parameter values, see Section 4.6.3.3.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

6.2.3 Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics

If you have modify or extend a seeded list of values, you must configure the CSV files for Oracle Procurement and Spend Analytics by mapping values from your source systems to the domain values.

This section explains how to extract the lists of values from your source system, which you then compare with the seeded values. If the lists of values are different to the seeded values, you need to follow the instructions to configure the Domain Values and CSV Worksheet Files.

6.2.3.1 List of Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics

The table below lists the CSV worksheet files and the domain values for Oracle Procurement and Spend Analytics in the `$ODI_HOME\biapps_odi\odifiles\odidatafiles\LkpFiles` directory on the ODI machine.

Table 6–1 Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics

Worksheet File Name	Description	Session
domainValues_Status_Purch_Approve_ora11i.csv	Lists the Purchasing Approval Status column and the corresponding domain values for the Oracle 11i application. For information about how to edit this file, see Section 6.2.3.2, "To configure domainValues_Status_Purch_Approve_ora11i.csv".	SDE_ORA_StatusDimension_PurchaseApprove
domainValues_Status_Purch_Cycle_ora11i.csv	Lists the Purchasing Cycle Status column and the corresponding domain values for the Oracle 11i application. For information about how to edit this file, see Section 6.2.3.3, "To configure domainValues_Status_Purch_Cycle_ora11i.csv".	SDE_ORA_StatusDimension_PurchaseCycle
domainValues_Xact_Types_PO_Line_Type_CODE1_ora11i.csv	List the Purchase Basis Type and the corresponding domain values for the Oracle EBS application. For information about how to edit this file, see Section 6.2.3.7, "To configure domainValues_Xact_Types_PO_Line_Type_CODE1_ora11i.csv".	SDE_ORA_TransactionTypeDimension_PO_Line_Type
domainValues_Xact_Types_PO_Line_Type_ora11i.csv	Lists the Purchasing Line Type and the corresponding domain Values for the Oracle EBS application. For information about how to edit this file, see Section 6.2.3.6, "To configure domainValues_Xact_Types_PO_Line_Type_ora11i.csv".	SDE_ORA_TransactionTypeDimension_PO_Line_Type
domainValues_Xact_Types_Purch_Orders_ora11i.csv	Lists the Purchase Order Transaction Type column and the corresponding domain values for the Oracle EBS application. For information about how to edit this file, see Section 6.2.3.4, "To configure domainValues_Xact_Types_Purch_Orders_ora11i.csv".	SDE_ORA_TransactionTypeDimension_PurchaseOrder
domainValues_Xact_Types_Purch_Requisitions_ora11i.csv	Lists the Purchase Requisition Transaction Type column and the corresponding domain values for the Oracle EBS application. For information about how to edit this file, see Section 6.2.3.5, "To configure domainValues_Xact_Types_Purch_Requisitions_ora11i.csv".	SDE_ORA_TransactionTypeDimension_PurchaseRequest

6.2.3.2 To configure domainValues_Status_Purch_Approve_ora11i.csv

This section explains how to configure domainValues_Status_Purch_Approve_ora11i.csv.

1. Identify the Purchase Approval Status in your Oracle EBS source system by using the following SQL:

```
SELECT A.LOOKUP_CODE FROM   FND_LOOKUP_VALUES A WHERE
A.LOOKUP_TYPE='AUTHORIZATION STATUS' AND A.LANGUAGE = 'US'
```

2. Open the domainValues_Status_Purch_Approve_ora11i.csv file in a text editor.
3. Copy the LOOKUP_CODE to the STATUS_CODE column in the file.

The data must be copied starting from the 8th line.

4. Map each LOOKUP_CODE to one Purchase Approval (PURCH_APPROVAL) domain value.

   Use commas to separate the entries.

5. Save and close the file.

6.2.3.3 To configure domainValues_Status_Purch_Cycle_ora11i.csv

This section explains how to configure domainValues_Status_Purch_Cycle_ora11i.csv.

1. Identify the Purchase Cycle Status in your Oracle EBS source system by using the following SQL:

   ```
   SELECT A.LOOKUP_CODE FROM   FND_LOOKUP_VALUES A WHERE
   A.LOOKUP_TYPE='DOCUMENT STATE' AND A.LANGUAGE = 'US'
   ```

2. Open the domainValues_Status_Purch_Cycle_ora11i.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles directory.

3. Copy the LOOKUP_CODE to the STATUS_CODE column in the file.

   The data must be copied starting from the 8th line.

4. Map each LOOKUP_CODE to one Purchase Cycle (PURCH_CYCLE) domain value.

   Use commas to separate the entries.

5. Save and close the file.

6.2.3.4 To configure domainValues_Xact_Types_Purch_Orders_ora11i.csv

This section explains how to configure domainValues_Xact_Types_Purch_Orders_ora11i.csv.

1. Identify the Purchase Order Types in your Oracle EBS source system by using the following SQL:

   ```
   SELECT DISTINCT PO_DOCUMENT_TYPES_ALL_TL.DOCUMENT_SUBTYPE

   FROM PO_DOCUMENT_TYPES_ALL_TL

   WHERE PO_DOCUMENT_TYPES_ALL_TL.LANGUAGE='US' AND PO_DOCUMENT_
   TYPES_ALL_TL.DOCUMENT_TYPE_CODE IN ('PO', 'PA') AND   PO_
   DOCUMENT_TYPES_ALL_TL.DOCUMENT_SUBTYPE <> 'CONTRACT'

   UNION SELECT 'COMPLEXWORK_ACTUAL' FROM DUAL

   UNION SELECT 'COMPLEXWORK_FINANCING' FROM DUAL;
   ```

2. Open the domainValues_Xact_Types_Purch_Orders_ora11i.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles directory.

3. Copy the DOCUMENT_SUBTYPE to the XACT_SUBTYPE_CODE column in the file.

   The data must be copied starting from the 8th line.

4. Map each DOCUMENT_SUBTYPE to one Purchase Order transaction type (PURCH_ORDERS) domain value.

Use commas to separate the entries.

5. Save and close the file.

6.2.3.5 To configure domainValues_Xact_Types_Purch_Requisitions_ora11i.csv

This section explains how to configure domainValues_Xact_Types_Purch_Requisitions_ora11i.csv.

1. Identify the Purchase Requisition Type in your Oracle EBS source system by using the following SQL:

   ```
   SELECT DISTINCT PO_DOCUMENT_TYPES_ALL_TL.DOCUMENT_SUBTYPE

   FROM PO_DOCUMENT_TYPES_ALL_TL

   WHERE PO_DOCUMENT_TYPES_ALL_TL.LANGUAGE='US' AND PO_DOCUMENT_
   TYPES_ALL_TL.DOCUMENT_TYPE_CODE = 'REQUISITION'
   ```

2. Open the domainValues_Xact_Types_Purch_Requisitions_ora11i.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles directory.

3. Copy the DOCUMENT_SUBTYPE to the XACT_SUBTYPE_CODE column in the file.

   The data must be copied starting from the 7th line.

4. Map each DOCUMENT_SUBTYPE to one Purchase Requisition Type (PURCH_RQLNS) domain value.

   Use commas to separate the entries.

5. Save and close the file.

6.2.3.6 To configure domainValues_Xact_Types_PO_Line_Type_ora11i.csv

This section explains how to configure domainValues_Xact_Types_PO_Line_Type_ora11i.csv.

1. Identify the Purchase Order Line Type in your Oracle EBS source system by using the following SQL:

   ```
   SELECT DISTINCT PO_LINE_TYPES_V.ORDER_TYPE_LOOKUP_CODE FROM
   PO_LINE_TYPES_V
   ```

2. Open domainValues_Xact_Types_PO_Line_Type_ora11i.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles directory.

3. Copy the ORDER_TYPE_LOOKUP_CODE to the XACT_TYPE_CODE column in the file.

   The data must be copied starting from the 8th line.

4. Map each ORDER_TYPE_LOOKUP_CODE to one PO Line Type transaction type (PO_LINE_TYPE) domain value.

   Use commas to separate the entries.

5. Save and close the file.

6.2.3.7 To configure domainValues_Xact_Types_PO_Line_Type_CODE1_ora11i.csv

This section explains how to configure domainValues_Xact_Types_Code1_PO_Line_Type_ora11i.csv.

1. Identify the Purchase Order Line Purchase Basis in your Oracle EBS source system by using the following SQL:

   ```
   SELECT DISTINCT PO_LINE_TYPES_V.PURCHASE_BASIS FROM   PO_
   LINE_TYPES_V
   ```

2. Open the domainValues_Xact_Types_Code1_PO_Line_Type_ora11i.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles directory.

3. Copy the PURCHASE_BASIS to the XACT_TYPE_CODE1 column in the file.

   The data must be copied starting from the 8th line.

4. Map each PURCHASE_BASIS to one PO Line Type Purchase Basis Type (PO_LINE_TYPE) domain value.

   Use commas to separate the entries.

5. Save and close the file.

6.2.4 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle Procurement and Spend Analytics.

- Section 6.2.4.1, "About Configuring the Purchase Receipts Aggregate Table"
- Section 6.2.4.2, "How to Configure the Purchase Receipts Aggregate Table"
- Section 6.2.4.3, "About Configuring the Purchase Cycle Lines Aggregate Table"
- Section 6.2.4.4, "How to Configure the Purchase Cycle Lines Aggregate Table"

6.2.4.1 About Configuring the Purchase Receipts Aggregate Table

The Purchase Receipts aggregate table (`W_PURCH_RCPT_A`) is used to capture information about the product receipts received from your suppliers and the purchase orders placed on them by your purchasing organization.

For your initial ELT run, you need to configure the GRAIN parameter for the time aggregation level in the Purchase Receipts Aggregate fact table.

For the incremental ELT run, you need to configure the time aggregation level and the source identification. The source identification value represents the source system you are sourcing data from.

You need to configure two parameters to aggregate the Purchase Receipts table for your incremental run:

- GRAIN
- TIME_GRAIN

These parameters have a preconfigured value of Month. The possible values for the parameter are:

- DAY
- WEEK

- MONTH
- QUARTER
- YEAR

The Purchase Receipt Lines aggregate table is fully loaded from the base table in the initial ELT run. The table can grow to millions of records. Thus, the Purchase Receipts aggregate table is not fully reloaded from the base table after each incremental ELT run. Oracle Business Analytics Warehouse minimizes the incremental aggregation effort, by modifying the aggregate table incrementally as the base table is updated. This process is done in four steps:

1. Oracle Business Analytics Warehouse finds the records to be deleted in the base table since the last ELT run, and loads them into the `W_PURCH_RCPT_TMP` table. The measures in these records are multiplied by (-1). The mapping responsible for this task is suffixed with _Derive_PreSoftDeleteImage, and it is run before the records are deleted from the base table. The mapping is run in the source-specific workflow.
2. Oracle Business Analytics Warehouse finds the records to be updated in the base table since the last ELT run, and loads them into the `W_PURCH_RCPT_TMP` table. The measures in these records are multiplied by (-1). The mapping responsible for this task is suffixed with _Derive_PreLoadImage, and it is run before the records are updated in the base table. It is run in the source-specific workflow.
3. Oracle Business Analytics Warehouse finds the inserted or updated records in the base table since the last ELT run, and loads them into the `W_PURCH_RCPT_TMP` table, without changing their sign. The mapping responsible for this task is suffixed with _Derive_PostLoadImage, and it is run after the records are updated or inserted into the base table. It is run in the post load-processing workflow.
4. Oracle Business Analytics Warehouse aggregates the `W_PURCH_RCPT_TMP` table, and joins it with the `W_PURCH_RCPT_A` aggregate table to insert new or update existing buckets to the aggregate table. This step is part of the post load-processing workflow, and the mapping is suffixed with _Derive.

6.2.4.2 How to Configure the Purchase Receipts Aggregate Table

To load the Purchase Receipts aggregate table (`W_PURCH_RCPT_A`), you need to configure the post-load-processing parameter file and the source system parameter files, and run the initial workflow and then the incremental workflow.

To configure the Purchase Receipts Aggregate Table:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Application Specific tab, and select Order Management Analytics from the **Select BI Application** drop down list.
4. Locate the following parameters and use the **Parameter Value** field to set the value:
   - TIME_GRAIN (default is MONTH, for the SIL_PurchaseReceiptAggregate_Derive_PreLoadImage scenario).
5. Save your changes.

For more information about specifying parameter values, see Section 4.6.3.3.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

6.2.4.3 About Configuring the Purchase Cycle Lines Aggregate Table

To aggregate the Purchase Cycle Lines table (W_PURCH_CYCLE_LINE_A), you need to configure the parameters in Oracle BI Applications Configuration Manager, and run the initial ELT workflow and then the incremental ELT workflow.

For your initial ELT run, you need to configure the GRAIN parameter for the time aggregation level in the Purchase Cycle Lines Aggregate fact table.

For the incremental ELT run, you need to configure the time aggregation level and the source identification. The source identification value represents the source system you are sourcing data from.

You need to configure two parameters to aggregate the Purchase Cycle Lines table for your incremental run:

- GRAIN
- TIME_GRAIN

These parameters have a preconfigured value of Month. The possible values for parameters are:

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

The Purchase Cycle Lines aggregate table is fully loaded from the base table in the initial ELT run. The table can grow to millions of records. The Purchase Cycle Lines aggregate table is not fully reloaded from the base table after an ELT run. Oracle Business Analytics Warehouse minimize the incremental aggregation effort, by modifying the aggregate table incrementally as the base table gets updated. This process is done in four steps:

1. Oracle Business Analytics Warehouse finds the records to be deleted in the base table since the last ELT run, and loads them into the `W_PURCH_CYCLE_LINE_TMP` table. The measures in these records are multiplied by (-1). The mapping responsible for this task is suffixed with _Derive_PreSoftDeleteImage, and it is run before the records are deleted from the base table. It is run in the source-specific workflow.
2. Oracle Business Analytics Warehouse finds the records to be updated in the base table since the last ELT run, and loads them into the `W_PURCH_CYCLE_LINE_TMP` table. The measures in these records are multiplied by (-1). The mapping responsible for this task is suffixed with _Derive_PreLoadImage, and it is run before the records are updated in the base table. It is run in the source-specific workflow.
3. Oracle Business Analytics Warehouse finds the inserted or updated records in the base table since the last ELT run, and loads them into the `W_PURCH_CYCLE_LINE_TMP` table, without changing their sign. The mapping responsible for this task is suffixed with _Derive_PostLoadImage, and it is run after the records are updated or inserted into the base table. It is run in the post load-processing workflow.

4. Oracle Business Analytics Warehouse aggregates the `W_PURCH_CYCLE_LINE_TMP` table, and joins it with the W_PURCH_CYCLE_LINE_A aggregate table to insert new or update existing buckets to the aggregate table. This step is part of the post load-processing workflow, and the mapping is suffixed with _Derive.

6.2.4.4 How to Configure the Purchase Cycle Lines Aggregate Table

Before you load the Purchase Cycle Lines aggregate table (`W_PURCH_CYCLE_LINE_A`), you need to configure the post-load-processing parameter run the initial workflow and then the incremental workflow, as follows.

To configure the Purchase Cycle Lines Aggregate Table:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Application Specific tab, and select Order Management Analytics from the **Select BI Application** field.
4. Locate the following parameters and use the **Parameter Value** field to set the value:
   - TIME_GRAIN (default is MONTH, for the SIL_PurchaseCycleLinesAggregate_Derive_PreLoadImage scenario).
   - GRAIN (default is MONTH, for the PLP_PurchaseCycleLinesAggregate_Load_Full scenario).
5. Save your changes.

   For more information about specifying parameter values, see Section 4.6.3.3.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

7 Configuring Oracle Financial Analytics

This chapter describes how to configure Oracle Financial Analytics for particular sources to meet your business needs, and contains the following topics:

- Section 7.1, "Overview of Oracle Financial Analytics"
- Section 7.2, "Configuration Required Before A Full Load for Financial Analytics"

To find out about other possible tasks required to deploy Oracle Business Intelligence Applications, see Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle Business Intelligence Applications With ODI".

7.1 Overview of Oracle Financial Analytics

Oracle Financial Analytics consists of the following:

- **Oracle General Ledger and Profitability Analytics**. The General Ledger Analytics application provides information to support your enterprise's balance sheet and provides a detailed analysis on the overall health of your company. The default configuration for the General Ledger Analytics application is based on what is identified as the most-common level of detail or granularity. However, you can configure and modify the extracts to best meet your business requirements.

  The Profitability Analytics application provides cost analysis, revenue trends, and sales performance to provide an accurate picture of profit and loss. The information found in the Profitability Analytics application pertains to data found in the revenue and expense account groupings of your financial statements and chart of accounts. The Profitability Analytics application is designed to provide insight into your enterprise's revenue and profitability information, which ties into your accounts receivable. The default configuration for the Profitability Analytics application is based on what is identified as the most-common level of detail, or granularity. However, the extracts are configurable and you can modify the extracts to meet your business requirements. The Profitability Analytics application provides cost analysis, revenue trends, and profitability analysis at the products and customer level, and the income statement at the company and business divisions level.

- **Oracle Payables Analytics**. The Oracle Payables Analytics application provides information about your enterprise's accounts payable information and identifies the cash requirements to meet your obligations.

  The information found in the Oracle Payables Analytics application pertains to data found exclusively under Accounts Payable (AP) in your financial statements and chart of accounts. Analysis of your payables allows you to evaluate the efficiency of your cash outflows. The need for analysis is increasingly important because suppliers are becoming strategic business partners with the focus on increased efficiency for just in time, and quality purchasing relationships.

The default configuration for the Oracle Payables Analytics application is based on what is identified as the most- common level of detail, or granularity. However, you can configure or modify the extracts to best meet your business requirements.

- **Oracle Receivables Analytics**. The Oracle Receivables Analytics application provides information to support your credit and collection activities, and to monitor and identify potential, receivables problems.

  The information found in the Oracle Receivables Analytics application pertains to data found exclusively in the Accounts Receivable (AR) account grouping of your financial statements and chart of accounts. Each day that your receivables are past the due date represents a significant, opportunity-cost to your company. Keeping a close eye on the trends, and clearing of AR is one way to assess the efficiency of your sales operations, the quality of your receivables, and the value of key customers.

  The default configuration for the Oracle Receivables Analytics application is based on what is identified as the most-common level of detail or granularity. However, you may configure and modify the extracts to best meet your business requirements.

7.2 Configuration Required Before A Full Load for Financial Analytics

This section contains configuration steps that you need to perform on Oracle Financial Analytics before you do a full data load, and contains the following topics:

- Section 7.2.1, "About Configuring Domain Values and CSV Worksheet Files for Oracle Financial Analytics"
- Section 7.2.2, "How to configure domainValues_Xact_Types_DocTypes_ora11i.csv for Oracle General Ledger and Profitability Analytics"
- Section 7.2.3, "How to Specify the Ledger or Set of Books for which GL Data is Extracted"
- Section 7.2.4, "Configuration Steps for Controlling Your Data Set"

7.2.1 About Configuring Domain Values and CSV Worksheet Files for Oracle Financial Analytics

If you modify or extend a seeded list of values, you must configure the CSV files for Oracle Financial Analytics by mapping values from your source systems to the domain values.

The following sections explain how to extract the lists of values from your source system, which you then compare with the seeded values. If the lists of values are different to the seeded values, you need to follow the instructions to configure the Domain Values and CSV Worksheet Files.

Typically, the number of rows returned by the source system will not match the number of rows provided in the domain valuesXXX.csv files. If so, you need to delete extra rows or insert additional rows in the CSV file to accommodate all the values returned by the source system in context.

The table below lists the CSV worksheet files and the domain values for Financial Analytics and Oracle EBS in the $ODI_HOME\biapps_
odi\odifiles\odidatafiles\lkpfiles folder.

Table 7–1 Domain Values and CSV Worksheet Files for Oracle Financial Analytics and Oracle EBS

Worksheet File Name	Description	Package/Interface
domainValues_Xact_Types_DocTypes_ora11i.csv	List the Transaction types and the corresponding domain values for the Oracle EBS 11i application. For more information about updating the values in this file, see Section 7.2.2, "How to configure domainValues_Xact_Types_DocTypes_ora11i.csv for Oracle General Ledger and Profitability Analytics").	SDE_ORA_TransactionTypeDimension_GLCOGSDerive

7.2.2 How to configure domainValues_Xact_Types_DocTypes_ora11i.csv for Oracle General Ledger and Profitability Analytics

This section explains how to configure domainValues_Xact_Types_DocTypes_ora11i.csv for Oracle General Ledger and Profitability Analytics, Release 11i.

To configure domainValues_Xact_Types_DocTypes_ora11i.csv for Oracle General Ledger and Profitability Analytics:

1. Identify the Entry Types in your Oracle Inventory Application by using the following SQL:

   ```
   select mtt.transaction_type_id, mtt.description from mtl_
   transaction_types mtt;
   ```

   This query gives the transaction type codes in your Oracle Inventory Application and their corresponding descriptions. Oracle Inventory Application is a data source from which Oracle General Ledger and Profitability Analytics extract data.

2. Open the domainValues_Xact_Types_DocTypes_ora11i.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles directory.

   Look for only the rows in the file which has xact_cat_code = 'COGS'. From among these records, look for those with W_XACT_TYPE_CODE = 'DELIVERY' or 'RETURN'. The column XACT_TYPE_CODE will contain the entry type codes from Oracle Inventory application, and the column W_XACT_TYPE_CODE is the corresponding domain value to which this entry type code will be mapped to.

3. Map the entry type codes returned by the query above from your Oracle Inventory Application to the domain values.

   Make sure the entry type codes go into the XACT_TYPE_CODE column in the file. Note that it is not necessarily a one-to-one mapping between the domain values and the entry types. For example, you can have multiple entry types mapped into the DELIVERY/RETURN domain value. Be careful not to map the same entry type to two different domain values.

4. Save and close the file.

7.2.3 How to Specify the Ledger or Set of Books for which GL Data is Extracted

If you have an Oracle EBS source system, you can specify from which set of books or ledgers you extract the GL data.

Oracle Business Intelligence Applications enables you to configure:

- The list of ledgers or set of books from which to extract the GL data.
- The types of ledgers or set of books for which to extract the GL data for.

You can use either of these configuration points separately and combine them. When installed out-of-the-box, Oracle Business Intelligence Applications extract all GL data for all ledgers or set of books.

For Oracle 11i customers, to configure the list of set of books to extract the GL data for, do the following:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Application Specific tab, and select Financial Analytics from the **Select BI Application** field.
4. Locate the following parameters and use the **Parameter Value** field to set the value:
   - FILTER_BY_SET_OF_BOOKS_TYPE

     Set the value to 'Yes'.
   - SET_OF_BOOKS_ID_LIST

     Enter the IDs of the set of books for which you want to extract GL data for.

     Specify the list of set of book IDs separated by commas (do not specify single or double quotes).

     For example, if you want to extract GL data for set of books with IDs: 101, 207, and 303, then set the value of this parameter to the following:

     ```
     101, 207, 303
     ```
5. Save your changes.

   For more information about specifying parameter values, see Section 4.6.3.3.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

You can also specify the Set of Books type you want to extract GL data for. In Oracle 11i, there are three set of books types (this is based on the column GL_SETS_OF_BOOKS.mrc_sob_type_code column):

- P (Parent)
- R (Reporting)
- N (None)

For Oracle 11i customers, to configure the types of set of books to extract the GL data, do the following:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Application Specific tab, and select Financial Analytics from the **Select BI Application** field.
4. Locate the following parameters and use the **Parameter Value** field to set the value:

- FILTER_BY_SET_OF_BOOKS_TYPE

  Set the value to 'Yes'.

- SET_OF_BOOKS_TYPE_LIST

  Enter the types of set of books for which you want to extract GL data for.

  Specify the list of set of book IDs in single quotes and separated by commas.

  For example, if you want to extract GL data for all Parent set of books and all Reporting set of books, then set the value of this parameter as follows:

  ```
  'P', 'R'
  ```

5. Save your changes.

   For more information about specifying parameter values, see Section 4.6.3.3.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

7.2.4 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle Financial Analytics, and contains the following topics:

7.2.4.1 How to include UnApplied Payments in AR Aging Calculations

To include UnApplied Payments in Aging Calculations for AR Aging tables

1. In ODI Designer, open the package PLP_ARSnapshotInvoiceAging in the PLP folder.
2. Open the Interface prep SQ_IA_AR_XACTS.
3. Display the Diagram tab and locate the table W_XACT_TYPE_D.
4. Selects the filter on the W_XACT_SUBTYPE_CODE column.

5. Change the SQL to the following:

```
W_XACT_TYPE_D.W_XACT_SUBTYPE_CODE IN ('INVOICE','CR MEMO','DR MEMO', 'PAYMENT')
```

6. Save the changes.
7. Regenerate the scenario of this package (expand the scenario folder under this package, right click the scenario and choose regenerate, keeping all the parameters as default).

   The next E-LT will populate the Aging tables using UnApplied payments in the calculations.

7.2.4.2 How to Configure the AP/AR Aging Tables

This section explains how to control the lengths of the aging buckets in the AP and AR aging snapshot tables. These tables are:

- W_AP_AGING_INVOICE_A
- W_AR_AGING_INVOICE_A
- W_AP_AGING_SUPPLIER_A
- W_AR_AGING_CUSTOMER_A

In these four tables, outstanding AP/AR balance information is broken out into rows. Each row represents the outstanding balance information for a particular aging bucket. Four aging buckets are provided out-of-the-box, with the following durations:

- Bucket 1: 0 – 30 days
- Bucket 2: 31 – 60 days
- Bucket 3: 61 – 90 days
- Bucket 4: 90+ days

To configure the length of aging buckets:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Application Specific tab, and select Financial Analytics from the **Select BI Application** field.
4. Locate the following parameters and use the **Parameter Value** field to set the following values:
   - BUCKET1_START and BUCKET1_END
   - BUCKET2_START and BUCKET2_END
   - BUCKET3_START and BUCKET3_END
   - BUCKET4_START and BUCKET4_END
5. Save your changes.

   For more information about specifying parameter values, see Section 4.6.3.3.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

These aging tables are snapshot tables. You can also control how many historic month end snapshots you want to keep in these tables (the default value is 24 months). You can increase or decrease the number by adjusting the $$HISTORY_MONTHS parameter.

7.2.4.3 How to Configure How GL Balances Is Populated in Oracle EBS

To populate the GL balances (stored in the W_GL_BALANCE_F table), you can either extract them directly from Oracle General Ledger, or calculate them based on the records in the W_GL_OTHER_F table (which stores all journal lines).

To configure how GL balances is populated in Oracle EBS:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Global tab.
4. Locate the following parameters and use the **Parameter Value** field to set the values:

   To extract the GL balances directly from Oracle General Ledger:

   - Set 'LOAD_EXTRACT_GL_BALANCE' to Y.
   - Set 'LOAD_CALCULATE_GL_BALANCE' to N.

   To calculate GL balances based on the records in the W_GL_OTHER_F table:

   - Set 'LOAD_EXTRACT_GL_BALANCE' to N.
   - Set 'LOAD_CALCULATE_GL_BALANCE' to Y.
5. Save your changes.

   For more information about specifying parameter values, see Section 4.6.3.3.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

7.2.4.4 How to Configure Oracle Profitability Analytics Transaction Extracts

There are two separate transaction extracts for Oracle General Ledger and Profitability Analytics—General Ledger Revenue and General Ledger COGS. By default, the Oracle General Ledger Analytics application extracts only Completed revenue and COGS that have been posted to the general ledger. Completed revenue transactions are those where the `RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG = Y`. If you want to extract incomplete revenue transactions, you can remove the filter in the extraction Interface.

To modify the extract filter for Oracle General Ledger and Profitability Analytics Revenue:

1. In ODI Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA11510_Adaptor).
2. Open the Package `mplt_BC_ORA_GLRevenueFact`.
3. Edit the Interface Prep SQ_GL_REVENUE_EXTRACT.
4. Display the Diagram tab.
5. Locate the source table 'RA_CUSTOMER_TRX_ALL' and delete its filter (with the value RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG='Y').
6. Save your changes.
7. Repeat steps 3 to 6 for the Interface Prep SQ_GL_REVENUE_EXTRACT_FULL.

8. Re-generate the scenario of this package (expand the scenario folder under this package, right click the scenario and choose regenerate, keep all the parameters as default).

7.2.4.5 How to Configure Cost Of Goods Extract (Oracle EBS 11i-specific)

For customers of EBS 11i, 11.5.8, 11.5.9, and 11.5.10, the Oracle General Ledger Analytics application extracts only COGS transactions that have been posted to the General Ledger, by default. All COGS transactions that have been transferred satisfy the following condition— MTL_TRANSACTION_ACCOUNTS.GL_BATCH_ID <> -1. If you want to extract all transactions, you can remove the filter in the extraction Interfaces. You need to modify the extraction interfaces for both the regular extract package as well as the primary extract package. This does not apply to EBS R12, in which by default all transactions are extracted. The following section applies only to EBS 11i customers.

To modify the extract filter for General Ledger COGS:

1. In ODI Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA11510_Adaptor).
2. Open the Package `SDE_ORA_GLCOGSFact`.
3. Edit the Interface Prep SQ_MTL_TRANSACTION_ACCOUNTS.
4. Display the Diagram tab.
5. Locate the source table Find source table 'RA_CUSTOMER_TRX_ALL' and delete its filter (with the value MTL_TRANSACTION_ACCOUNTS.GL_BATCH_ID<>-1).
6. Save your changes.
7. Repeat steps 3 to 6 for the Interface Prep SQ_MTL_TRANSACTION_ACCOUNTS_FU.
8. Re-generate the scenario of this package (expand the scenario folder under this package, right click the scenario and choose regenerate, keep all the parameters as default).
9. Open the Package `SDE_ORA_GLCOGSFact_Primary`.
10. Edit the Interface Prep SQ_MTL_TRANSACTION_ACCOUNTS_PR.
11. Display the Diagram tab.
12. Locate the source table 'RA_CUSTOMER_TRX_ALL' and delete its filter (with the value MTL_TRANSACTION_ACCOUNTS.GL_BATCH_ID<>-1).
13. Save your changes.
14. Re-generate the scenario of this package (expand the scenario folder under this package, right click the scenario and choose regenerate, keep all the parameters as default).

7.2.4.6 How to Configure AP Balance ID for Oracle Payables Analytics

The AP Balance ID controls the level at which the balance in W_AP_BALANCE_F is maintained.

This section contains configuration information for Oracle Payables Analytics that is specific to Oracle.

By default, the Accounts Payable (AP) Balance ID is maintained at the following granularity:

```
SET_OF_BOOKS_ID||'~'||CODE_COMBINATION_ID||'~'||VENDOR_SITE_ID||'~'||INP_ORG_ID
```

However, if you want to maintain your AP balance at a different grain, you can redefine the Balance ID value in the applicable packages.

To modify the Accounts Payable Balance ID:

Note: To modify the Accounts Payable Balance ID, you must modify the following packages:

- SDE_ORA_APTransactionFact_LiabilityDistribution
- SDE_ORA_APTransactionFact_ExpenseDistribution
- SDE_ORA_APTransactionFact_Payment
- SDE_ORA_APTransactionFact_PaymentSchedule
- SDE_ORA_Stage_APTransactionFact_DiffManDerive

---

1. In ODI Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA11510_Adaptor).
2. Open the appropriate Package `SDE_ORA_ARTransactionFact_Adjustments`, and display the Diagram tab.
3. Edit the Interface at the end of the flow (for example, Run AR_XACT_FS).
4. Display the Diagram tab.

   On the right of the pane, you will see the 'Target Datastore' panel with the column definition of the target table (e.g. W_AR_XACT_FS).
5. Select BALANCE_ID, and change the expression in the text editor in the lower pane.
6. Save your changes.
7. In the same package, make the same change for the full load interface.

   For example, the full load interface for the Run AR_XACT_FS package is Run AR_XACT_FS_Full.
8. Re-generate the scenario of this package (expand the scenario folder under this package, right click the scenario and choose regenerate, keep all the parameters as default).
9. Repeat steps 1 to 8 for each of the packages listed above.

7.2.4.7 How to Configure AR Balance ID for Oracle Receivables Analytics and Oracle General Ledger and Profitability Analytics

The AR Balance ID controls the level at which the balance in W_AR_BALANCE_F is maintained.

By default, the AR Balance ID is maintained at the following granularity:

```
set_of_books_id || '~' || code_combination_id || '~' || customer_id || '~' ||
customer_site_use_id || '~' ||transaction_currency_code || '~' || org_id
```

However, if you want to maintain your AR balance at a different grain, you can redefine the Balance ID value in the applicable packages.

To modify the AR Balance ID:

Note: To modify the AR Balance ID, you must modify the following packages:

- SDE_ORA_ARTransactionFact_Adjust
- SDE_ORA_ARTransactionFact_ARScheduleDerive
- SDE_ORA_ARTransactionFact_CreditMemoApplication
- SDE_ORA_ARTransactionFact_ReceivableApplication
- SDE_ORA_Stage_ARTransactionFact_DiffManDerive

---

1. In ODI Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA11510_Adaptor).
2. Open the appropriate Package `SDE_ORA_APTransactionFact_LiabilityDistribution`, and display the Diagram tab.
3. Edit the Interface at the end of the flow (for example, Run AP_XACT_FS).
4. Display the Diagram tab.

   On the right of the pane, you will see the 'Target Datastore' panel with the column definition of the target table (e.g. W_AP_XACT_FS).
5. Select BALANCE_ID, and change the expression in the text editor in the lower pane.
6. Save your changes.
7. In the same package, make the same change for the full load interface.

   For example, the full load interface for the Run AP_XACT_FS package is Run AP_XACT_FS_Full.
8. Re-generate the scenario of this package (expand the scenario folder under this package, right click the scenario and choose regenerate, keep all the parameters as default).
9. Repeat steps 1 to 8 for each of the packages listed above.

7.2.4.8 How to Configure the AR Adjustments Extract for Oracle Receivables Analytics

By default, Oracle Receivables Analytics extracts only approved adjustment entries against accounts receivable transactions. *Approved adjustments* are entries where the `AR_ADJUSTMENTS_ALL.STATUS = A.` If you want to extract additional types of AR adjustment entries, you can remove the filter in the extraction interface. By modifying or removing the filter, you can extract other entries, such as those that require more research, those that are rejected, or those that are not accrued charges.

To modify the extract filter for Accounts Receivable adjustments:

1. In ODI Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA11510_Adaptor).
2. Open the Package SDE_ORA_ARTransactionFact_Adjustments.
3. Edit the Interface Prep SQ_AR_XACTS_ADJ.
4. Display the Diagram tab.
5. Locate the source table ADJUSTMENTS_ALL.

6. Select the filter on this source table with the value AR_ADJUSTMENTS_ALL.STATUS='A' to display the 'Filter on datastore <*Name*>' pane, and remove or modify the filter value in the **Implementation** box.
7. Save your changes.
8. Repeat steps 3 to 7 for the Interface Prep SQ_AR_XACTS_ADJ_Full.
9. Re-generate the scenario of this package (expand the scenario folder under this package, right click the scenario and choose regenerate, keep all the parameters as default).
10. Repeat steps 1 to 8 for the Package SDE_ORA_ARTransactionFact_Adjustments_Primary.

7.2.4.9 How to Configure the AR Schedules Extract

By default, Oracle Receivables Analytics extracts only completed schedules; that is, transactions where the `RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG(+) = Y`. If you want to extract additional types of AR schedule entries, you must remove the filter in the extraction interface. By modifying or removing the filter, you can extract other entries, such as those that were marked as incomplete.

To modify the extract filter for Accounts Receivable schedules:

1. In ODI Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA11510_Adaptor).
2. Open the Package SDE_ORA_Stage_ARTransactionFact_ARSchedules.
3. Edit the Interface Prep SQ_AR_XACTS_AR_SCH.
4. Display the Diagram tab.
5. Locate the source table RA_CUSTOMER_TRX_ALL.
6. Select the filter on this source table with the value RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG(+)='Y' to display the 'Filter on datastore <*Name*>' pane, and remove or modify the filter value in the **Implementation** box.
7. Save your changes.
8. Repeat steps 3 to 7 for the Interface Prep SQ_AR_XACTS_AR_SCH_Full.
9. Re-generate the scenario of this package (expand the scenario folder under this package, right click the scenario and choose regenerate, keep all the parameters as default).

7.2.4.10 How to Configure the AR Cash Receipt Application Extract for Oracle Receivables Analytics

By default, Oracle Receivables Analytics extracts only confirmed, cash-receipt application entries against accounts receivable transactions. *Confirmed receipts* are entries where the `AR_RECEIVABLE_APPLICATIONS_ALL.CONFIRMED_FLAG = Y OR NULL`. If you want to extract additional types of cash-receipt application entries, you can remove the filter in the extraction interface. By modifying or removing the filter, you can extract other entries, such as unconfirmed applications.

You must modify both the regular package(SDE_ORA_ARTransactionFact_ReceivableApplication) as well as the primary extract package(SDE_ORA_ARTransactionFact_ReceivableApplication_Primary).

To modify the extract filter for AR cash receipt application:

1. In ODI Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA11510_Adaptor).
2. Open the Package SDE_ORA_ARTransactionFact_ReceivableApplication.
3. Edit the Interface Prep SQ_AR_XACTS_APPREC.
4. Display the Diagram tab.
5. Locate the source table AR_RECEIVABLE_APPLICATIONS_ALL.
6. Select the filter on this source table with the value NVL(AR_RECEIVABLE_APPLICATIONS_ALL.CONFIRMED_FLAG,'Y')='Y' to display the 'Filter on datastore <*Name*>' pane, and remove or modify the filter value in the **Implementation** box.
7. Save your changes.
8. Repeat steps 3 to 7 for the Interface Prep SQ_AR_XACTS_APPREC_Full.
9. Re-generate the scenario of this package (expand the scenario folder under this package, right click the scenario and choose regenerate, keep all the parameters as default).
10. Repeat steps 1 to 8 for the Package SDE_ORA_ARTransactionFact_ReceivableApplication_Primary.

7.2.4.11 How to Configure the AR Credit-Memo Application Extract for Oracle Receivables Analytics

By default, Oracle Receivables Analytics extracts only confirmed, credit-memo application entries against accounts receivable transactions. *Confirmed credit memos* are entries where the `AR_RECEIVABLE_APPLICATIONS_ALL.CONFIRMED_FLAG = Y OR NULL`. If you want to extract additional types of AR credit-memo application entries, you can remove the filter. By modifying or removing the filter, you can extract other entries such as unconfirmed, credit memos.

You must modify both the regular package SDE_ORA_ARTransactionFact_CreditmemoApplication as well as the primary extract package SDE_ORA_ARTransactionFact_CreditmemoApplication.

To modify the extract filter for AR cash receipt application:

1. In ODI Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA11510_Adaptor).
2. Open the Package SDE_ORA_ARTransactionFact_ CreditmemoApplication.
3. Edit the Interface Prep SQ_AR_XACTS_APPCM.
4. Display the Diagram tab.
5. Locate the source table AR_RECEIVABLE_APPLICATIONS_ALL.
6. Select the filter on this source table with the value NVL(AR_RECEIVABLE_APPLICATIONS_ALL.CONFIRMED_FLAG,'Y')='Y' to display the 'Filter on datastore <*Name*>' pane, and remove or modify the filter value in the **Implementation** box.
7. Save your changes.
8. Repeat steps 3 to 7 for the Interface Prep SQ_AR_XACTS_APPCM_Full.
9. Re-generate the scenario of this package (expand the scenario folder under this package, right click the scenario and choose regenerate, keep all the parameters as default).

10. Repeat steps 1 to 8 for the Package SDE_ORA_ARTransactionFact_ReceivableApplication_Primary.

7.2.4.12 How to Setup Drill Down in Oracle BI Answers from General Ledger to Sub-ledger

Oracle Business Intelligence Applications enables you to trace a GL Journal to the Sub-ledger transaction that created that journal. This ability (using drill down) is achieved through the Navigation feature in Oracle BI Answers. This feature is available for AP if the source is Oracle EBS 11i.

To set up drill down in Oracle BI Answers from General Ledger to Sub-ledger:

1. Create your sub-ledger request from 'Financials - AP Transactions' or 'Financials - AR Transactions' catalog as applicable.
2. In your request, add a filter on the column 'GL Journal ID' under the 'Document Details' folder and the set the operator of the filter to 'Is Prompted'.
3. Build your GL Journal request from the 'Financials - GL Detail Transactions' catalog.
4. To your request, add the column 'GL Journal ID' under the 'Document Details' folder.
5. Navigate to the Column Properties of this column, and set the Value Interaction property in the Column Format tab to 'Navigate'.
6. Add a navigation target and set the target location to the sub-ledger request you created earlier.

8 Configuring Oracle Supply Chain and Order Management Analytics

This chapter describes how to configure Oracle Supply Chain and Order Management Analytics for particular sources to meet your business needs, and contains the following topics:

- Section 8.1, "Overview of Oracle Supply Chain and Order Management Analytics"
- Section 8.2, "Configuration Required Before A Full Load for Oracle Supply Chain and Order Management Analytics"

To find out about other possible tasks required to deploy Oracle Business Intelligence Applications, see Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle Business Intelligence Applications With ODI".

8.1 Overview of Oracle Supply Chain and Order Management Analytics

The Oracle Supply Chain and Order Management Analytics application allows you to analyze:

- Bill of materials.
- Bookings.
- Financial and Operational Backlogs.
- Inventory held by an organization.
- Inventory movements in, out, and through manufacturing plants, distribution centers, or storage locations.
- Invoices.
- The movement of sales orders through different stages of the sales cycle.

The Oracle Supply Chain and Order Management Analytics application consists of orders, invoices, backlogs and inventory. Sales orders are the entry point for the sales process. Invoices are the exit point from the fulfillment process. Backlogs are points of congestion in your fulfillment process. This coverage includes insight into which items are booked, backlogged, and invoiced. This allows you to evaluate the sales performance of individual sales representatives or departments. Oracle Supply Chain and Order Management Analytics also provides you with information on Inventory Transactions, Inventory Balances, Bill of Materials and Customer and Supplier Returns. This enables companies to monitor inventory levels trend to Sales performance to improve cost exposure, increase turnover through inventory level reduction and increased velocity, deploy inventory at the right place and the right time. and better understand Customer and Supplier Returns to maintain quality.

The Oracle Supply Chain and Order Management Analytics application also requires post-load processing mappings to populate its aggregate and derived tables.

8.2 Configuration Required Before A Full Load for Oracle Supply Chain and Order Management Analytics

This section contains configuration steps that you need to perform on Oracle Supply Chain and Order Management Analytics before you do a full data load, and contains the following topics:

- Section 8.2.1, "About Configuring Domain Values and CSV Worksheet Files for Oracle Supply Chain and Order Management Analytics".
- Section 8.2.2, "How to Configure Invoice Type Domain Values".
- Section 8.2.3, "How to Configure Pick Types Domain Values".
- Section 8.2.4, "How to Configure Order Types Domain Values"
- Section 8.2.5, "How to Configure Pick Status Domain Values"
- Section 8.2.6, "How to Configure Invoice Status Domain Values"
- Section 8.2.7, "How to Configure Order Overall Status Domain Values"
- Section 8.2.8, "How to Configure Pay Method Domain Values"
- Section 8.2.9, "To Configure Movement Types Domain Values"
- Section 8.2.10, "How to Configure Quantity Types for Product Transactions"
- Section 8.2.11, "Configuration Steps for Controlling Your Data Set"

8.2.1 About Configuring Domain Values and CSV Worksheet Files for Oracle Supply Chain and Order Management Analytics

The table below lists the CSV worksheet files containing domain values for Oracle Supply Chain and Order Management Analytics in the `$ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles` directory.

Table 8–1 Domain Values and CSV Worksheet Files for Oracle Supply Chain and Order Management Analytics

Worksheet File Name	Description	Session
domainValues_InvoiceTypes_ora11i.csv	Lists the Invoice Document Type column and the corresponding domain values for the Oracle 11i or Oracle R12 application. For more information about updating the values in this file, see Section 8.2.2, "How to Configure Invoice Type Domain Values".	SDE_ORA_TransactionTypeDimension_SalesInvoiceLines

Table 8–1 (Cont.) Domain Values and CSV Worksheet Files for Oracle Supply Chain and Order Management Analytics

Worksheet File Name	Description	Session
domainValues_PickTypes_ora11i.csv	Lists the Picking Document Type column and the corresponding domain values for the Oracle 11i or Oracle R12 application. For more information about updating the values in this file, see Section 8.2.3, "How to Configure Pick Types Domain Values".	SDE_ORA_TransactionTypeDimension_SalesPickLines
domainValues_OrderTypes_ora11i.csv	Lists the Order Document Type column and the corresponding domain values for the Oracle 11i or Oracle R12 application. For more information about updating the values in this file, see Section 8.2.4, "How to Configure Order Types Domain Values".	SDE_ORA_TransactionTypeDimension_SalesOrderLines
domainValues_PickStatus_ora11i.csv	Lists the Picking Status Code and the Status Desc columns, and the corresponding domain values for the Oracle 11i or Oracle R12 application. For more information about updating the values in this file, see Section 8.2.5, "How to Configure Pick Status Domain Values".	SDE_ORA_StatusDimension_SalesPickLines
domainValues_PayMethodCode_ora11i.csv	Lists the method code column and the corresponding domain value for the application.	SDE_ORA_PaymentMethodDimension
domainValues_InvoiceStatus_ora11i.csv	Lists the Invoice Status Code and the Status Desc columns, and the corresponding domain values for the Oracle 11i or Oracle R12 application. For more information about updating the values in this file, see Section 8.2.6, "How to Configure Invoice Status Domain Values".	SDE_ORA_StatusDimension_SalesInvoiceLine
DomainValue_OrderOverallStatus_ora11i.csv	List the Order Status Code column and the corresponding domain values for the Oracle 11i or Oracle R12 application. For more information about updating the values in this file, see Section 8.2.7, "How to Configure Order Overall Status Domain Values".	SDE_ORA_StatusDimension_SalesOrderLineCycle

8.2.2 How to Configure Invoice Type Domain Values

This section explains how to configure Invoice Type Domain Values using the domainValues_InvoiceTypes_ora11i.csv file.

To configure Invoice Type Domain Values:

1. Identify the Invoice Types in your Oracle 11i source system by using the following SQL:

   ```
   SELECT DISTINCT RA_CUST_TRX_TYPES_ALL.TYPE
   FROM RA_CUST_TRX_TYPES_ALL
   ORDER BY 1;
   ```

2. Open the domainValues_InvoiceType_ora11i.csv file in a text editor

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles directory.

3. Copy the TYPE column to the XACT_TYPE_CODE column in the file.

   The data must be copied starting from the 2nd line. The first line is the column header.

4. Map each Transaction Type Code to one domain value.

   For more information on Transaction Type Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

5. Save and close the file.

8.2.3 How to Configure Pick Types Domain Values

This section explains how to configure Pick Types Domain Values using the domainValues_PickTypes_ora11i.csv file.

To configure Pick Types Domain Values:

1. Identify the Pick Types in your Oracle 11i source system.

2. Open the domainValues_PickType_ora11i.csv file in a text editor

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles directory.

3. Put 'STANDARD' in the XACT_TYPE_CODE column in the file.

   The data must be copied starting from the 2nd line. The first line is the column header.

4. Map each Transaction Type Code to one domain value.

   For more information on Transaction Type Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

5. Save and close the file.

8.2.4 How to Configure Order Types Domain Values

This section explains how to configure Order Types Domain Values using the domainValues_OrderTypes_ora11i.csv file.

To configure Order Types Domain Values:

1. Identify the Pick Types in your Oracle 11i source system by using the following SQL:

   ```
   SELECT DISTINCT FND_LOOKUP_VALUES.LOOKUP_CODE
   FROM FND_LOOKUP_VALUES
   ```

```
WHERE FND_LOOKUP_VALUES.VIEW_APPLICATION_ID = 660
AND FND_LOOKUP_VALUES.LANGUAGE = 'US'
AND FND_LOOKUP_VALUES.LOOKUP_TYPE = 'LINE_CATEGORY'
ORDER BY 1;
```

2. Open the domainValues_OrderType_ora11i.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles directory.

3. Copy the LOOKUP_TYPE column to the XACT_TYPE_CODE column in the file.

   The data must be copied starting from the 2nd line. The first line is the column header.

4. Map each Transaction Type Code to one domain value.

   For more information on Transaction Type Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

5. Save and close the file.

8.2.5 How to Configure Pick Status Domain Values

This section explains how to configure Pick Status Domain Values using the domainValues_PickStatus_ora11i.csv file.

To configure Pick Status Domain Values:

1. Identify the Pick Statuses in your Oracle 11i source system by using the following SQL:

```
SELECT DISTINCT FND_LOOKUP_VALUES.LOOKUP_CODE
FROM FND_LOOKUP_VALUES
WHERE FND_LOOKUP_VALUES.LOOKUP_TYPE= 'PICK_STATUS'
AND FND_LOOKUP_VALUES.LANGUAGE = 'US'
AND FND_LOOKUP_VALUES.VIEW_APPLICATION_ID = 665
AND FND_LOOKUP_VALUES.SECURITY_GROUP_ID = 0
ORDER BY 1;
```

2. Open the domainValues_PickStatus_ora11i.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles directory.

3. Copy the LOOKUP_CODE column to the STATUS_CODE column in the file.

   The data must be copied starting from the 2nd line. The first line is the column header.

4. Map each Status Code to one domain value.

   For more information on Status Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

5. Save and close the file.

8.2.6 How to Configure Invoice Status Domain Values

This section explains how to configure Invoice Status Domain Values using the domainValues_InvoiceStatus_ora11i.csv file.

To configure Invoice Status Domain Values:

1. Identify the Invoice Statuses in your Oracle 11i source system by using the following SQL:

```
SELECT DISTINCT FND_LOOKUP_VALUES.LOOKUP_CODE
FROM FND_LOOKUP_VALUES
WHERE FND_LOOKUP_VALUES.LOOKUP_TYPE= 'INVOICE_TRX_STATUS' AND
FND_LOOKUP_VALUES.LANGUAGE = 'US'
AND FND_LOOKUP_VALUES.VIEW_APPLICATION_ID = 222
AND FND_LOOKUP_VALUES.SECURITY_GROUP_ID = 0
ORDER BY 1;
```

2. Open the domainValues_InvoiceStatus_ora11i.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles directory.

3. Copy the LOOKUP_CODE column to the STATUS_CODE column in the file.

   The data must be copied starting from the 2nd line. The first line is the column header.

4. Map each Status Code to one domain value.

   For more information on Status Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

5. Save and close the file.

8.2.7 How to Configure Order Overall Status Domain Values

This section explains how to configure Order Overall Status Domain Values using the domainValues_OrderOverallStatus_ora11i.csv file.

To configure Order Overall Status Domain Values:

1. Identify the Order Overall Statuses in your Oracle 11i source system by using the following SQL:

```
SELECT DISTINCT FND_LOOKUP_VALUES.LOOKUP_CODE
FROM FND_LOOKUP_VALUES
WHERE FND_LOOKUP_VALUES.LOOKUP_TYPE = 'LINE_FLOW_STATUS'
 AND FND_LOOKUP_VALUES.LANGUAGE = 'US'
 AND FND_LOOKUP_VALUES.VIEW_APPLICATION_ID = 660
 AND FND_LOOKUP_VALUES.SECURITY_GROUP_ID = 0
ORDER BY 1;
```

2. Open the domainValues_OrderOverallStatus_ora11i.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles directory.

3. Copy the LOOKUP_CODE column to the STATUS_CODE column in the file.

   The data must be copied starting from the 2nd line. The first line is the column header.

4. Map each Status Code to one domain value.

   For more information on Status Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

5. Save and close the file.

8.2.8 How to Configure Pay Method Domain Values

This section explains how to configure Pay Method Status Domain Values using the domainValues_PayMethodCode_ora11i.csv file.

To configure Pay Method Domain Values:

1. Identify the Pay Methods in your Oracle 11i source system by using the following SQL:

```
SELECT DISTINCT FND_LOOKUP_VALUES.LOOKUP_CODE
FROM FND_LOOKUP_VALUES
WHERE LOOKUP_TYPE = 'PAYMENT TYPE'
 AND VIEW_APPLICATION_ID = 660
 AND LANGUAGE = 'US'
 AND FND_LOOKUP_VALUES.SECURITY_GROUP_ID = 0
ORDER BY 1;
```

2. Open the domainValues_PayMethodCode_ora11i.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles directory.

3. Copy the LOOKUP_CODE column to the METHOD_CODE column in the file.

   The data must be copied starting from the 2nd line. The first line is the column header.

4. Map each Method Code to one domain value.

   For more information on Method Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

5. Save and close the file.

8.2.9 To Configure Movement Types Domain Values

This section explains how to configure Movement Types domain values.

1. Identify the Inventory Movement Type in your Oracle EBS source system by using the following SQL:

   SELECT DISTINCT MTL_TRANSACTION_TYPES.TRANSACTION_TYPE_NAME FROM MTL_TRANSACTION_TYPES

2. Open the domainValues_Movement_Types_ora11i.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles directory.

3. Copy the TRANSACTION_TYPE_NAME to the TRANSACTION_TYPE_NAME column in the file.

   The data must be copied starting from the 2nd line.

4. Map each TRANSACTION_TYPE_NAME to one Inventory Movement Type domain value.

   Use commas to separate the entries.

5. Save and close the file.

8.2.10 How to Configure Quantity Types for Product Transactions

Oracle 11i categorize quantities into three different types:

- **Goods Received quantities**. *Goods Received quantity* refers to the number of goods received.
- **Delivery quantities**. *Delivery quantity* refers to the number of goods delivered.
- **Base quantities**. *Base quantity* refers to any transaction quantity.

The Oracle Business Analytics Warehouse extracts the transaction type and loads this value into the `XACT_SRC_TYPE` column. In this column, the value `1` denotes a Goods Received quantity, and `2` denotes a Delivery quantity.

To find out more about XACT_SRC_TYPE column, execute the following SQL against your EBS instance:

```
select TRANSACTION_SOURCE_TYPE_ID, TRANSACTION_SOURCE_TYPE_NAME,
DESCRIPTION from MTL_TXN_SOURCE_TYPES order by 1
```

If you have rows equivalent to Purchase order (1), you should include the TRANSACTION_SOURCE_TYP E_ID in the Goods Received quantity column (EXT_GR_QTY). If you have rows equivalent to Sales Order (2), you should include the TRANSACTION_SOURCE_TYPE_ID in the Delivery quantity column (EXT_DELIVERY_QTY).

All quantities extracted from the source system are always loaded into the Base quantity column (`EXT_BASE_QTY`). However, only the receipt quantity is loaded into the Goods Received quantity column (`EXT_GR_QTY`), and only delivered quantities are loaded into the Delivery quantity column (`EXT_DELIVERY_QTY`).

If your definition of goods received or delivery quantity is different from the prepackaged condition, then you can edit the condition to suit your business needs.

To configure the Quantity type:

1. In ODI Designer, open the Project and the folder SDE_ORA11510_Adaptor.
2. Open the package SDE_ORA_ProductTransactionFact.
3. Edit the interface Run PRODUCT_XACT_FS_FULL to display the Interface: <**Name**> dialog and display the Diagram tab.
4. In the Target Datastore, select GR_QTY to display the Mapping pane, and use the **Implementation** field to change the value.
5. In the Target Datastore, select DELIVERY_QTY to display the Mapping pane, and use the **Implementation** field to change the value.
6. Save the changes.
7. Repeat steps 3 to 6 for the interface Run PRODUCT_XACT_FS

8.2.11 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle Supply Chain and Order Management Analytics, and contains the following topics:

8.2.11.1 Tracking Multiple Attribute Changes in Bookings

When you modify the default VAR_BOOKING_ID column, the SQL statement is configured as follows for Oracle 11i and Oracle R12:

```
TO_CHAR(INP_LINE_ID)||'~'||TO_CHAR(INP_INV_ITEM_ID)||'~'||to_char(INP_WAREHOUSE_
ID)
```

However, if you want to track changes based on more than one attribute, in the SQL statement you must concatenate the attribute column IDs in the VAR_BOOKING_ID column. For example, if you want to track changes in Salespersons and Customer, then concatenate the technical name IDs in the VAR_BOOKING_ID column as follows:

```
TO_CHAR(INP_LINE_ID)||'~'||TO_CHAR(INP_INV_ITEM_ID)||'~'||TO_CHAR(INP_WAREHOUSE_
ID)||'~'||TO_CHAR(INP_SALESREP_ID)||'~'||TO_CHAR(INP_CUSTOMER_ID)
```

To track dimensional attribute changes in bookings:

1. In ODI Designer, open the SDE_ORA11510_Adaptor folder.
2. Open the Interface SDE_ORA_SalesOrderLinesFact_Full.SALES_ORDER_LINE_FS_FULL.
3. Double-click the column BOOKING_ID on the target datastore.
4. In the implementation tab, edit the expression, and enter the ID of the attribute for which you want to track changes.

   If you want to track changes in multiple attributes, concatenate the IDs of the attributes and put the concatenated value in the BOOKING_ID column. Be sure to convert numeric columns to varchar type 5.
5. Repeat steps 2 to 4 for the following Interfaces:
   - SDE_ORA_SalesOrderLinesFact.SALES_ORDER_LINE_FS
   - SDE_ORA_SalesScheduleLinesFact_Full.SALES_SCHEDULE_LINE_FS_FULL
   - SDE_ORA_SalesScheduleLinesFact.SALES_SCHEDULE_LINE_FS
6. Save your changes to the repository.

8.2.11.2 Process of Aggregating Oracle Supply Chain and Order Management Analytics Tables

This section contains configuration points for Oracle Supply Chain and Order Management Analytics for aggregating the Sales Invoice Lines and Sales Order Lines tables. This section contains the following topics:

- Section 8.2.11.2.1, "About Configuring the Sales Invoice Lines Aggregate Table"
- Section 8.2.11.2.2, "About Configuring the Sales Order Lines Aggregate Table"
- Section 8.2.11.2.3, "How to Configure the Sales Invoice Lines and Sales Order Lines Aggregate Tables"

Note: The aggregation processes use the following Teradata parameters:

- Hint_Tera_Pre_Cast
- Hint_Tera_Post_Cast

8.2.11.2.1 About Configuring the Sales Invoice Lines Aggregate Table The Sales Invoice Lines aggregate table (W_SALES_INVOICE_LINE_F_A) is used to capture information about the invoices issued for your sales orders. You use the TIME_GRAIN parameter to configure the Sales Invoice Lines aggregate tables in order to run initial ELT and incremental ELT.

The TIME_GRAIN parameter is set to MONTH out-of-the-box, but can also be set to DAY, WEEK, QUARTER, and YEAR.

The Sales Invoice Lines aggregate table is fully loaded from the base table in the initial ELT run. The table can grow to millions of records. Thus, the Sales Invoice aggregate table is not fully reloaded from the base table after each incremental ELT run. Oracle Business Analytics Warehouse minimizes the incremental aggregation effort, by modifying the aggregate table incrementally as the base table is updated. This process is described below.

- The Oracle Business Analytics Warehouse finds the records to be deleted in the base table since the last ELT run, and loads them into the `W_SALES_INVOICE_LINE_TMP` table. The measures in these records are multiplied by (-1). The mapping responsible for this task is SIL_SalesInvoiceLinesAggregate_Derive_PreSoftDeleteImage, which is run before SIL_SalesInvoiceLinesFact_SoftDelete deletes the records from the base table.
- The Oracle Business Analytics Warehouse finds the records to be updated in the base table since the last ELT run, and loads them into the `W_SALES_INVOICE_LINE_TMP` table. The measures in these records are multiplied by (-1). The mapping responsible for this task is SIL_SalesInvoiceLinesFact_Derive_PreLoadImage, which is run before SIL_SalesInvoiceFact deletes the records from the base table.
- The Oracle Business Analytics Warehouse finds the inserted or updated records in the base table since the last ELT run, and loads them into the `W_SALES_INVOICE_LINE_TMP` table, without changing their sign. The mapping responsible for this task is SIL_SalesInvoiceLinesFact_Derive_PreLoadImage, which is run before PLP_SalesInvoiceLinesFact_Derive_PostLoadImage updates or inserts records in the base table.
- The Oracle Business Analytics Warehouse aggregates the `W_SALES_INVOICE_LINE_TMP` table and load to W_SALES_INVOICE_LINE_A_TMP, which has the same granularity as the W_SALES_INVOICE_LINE_A table.
- The PLP_SalesInvoiceLinesAggregate_Derive mapping looks up the W_SALES_INVOICE_LINE_A aggregate table to update existing buckets or insert new buckets in the aggregate table (the mapping is PLP_SalesInvoiceLinesAggregate_Load).

8.2.11.2.2 About Configuring the Sales Order Lines Aggregate Table The Sales Order Lines aggregate table (W_SALES_ORDER_LINE_A) is used to capture information about the invoices issued for your sales orders. You use the TIME_GRAIN parameter to configure the Sales Invoice Lines aggregate tables in order to run initial ELT and incremental ELT.

The TIME_GRAIN parameter is set to MONTH out-of-the-box, but can also be set to DAY, WEEK, QUARTER, and YEAR.

The Sales Order Lines aggregate table is fully loaded from the base table in the initial ELT run. The table can grow to millions of records. Thus, the Sales Order aggregate table is not fully reloaded from the base table after each incremental ELT run. Oracle Business Analytics Warehouse minimizes the incremental aggregation effort, by modifying the aggregate table incrementally as the base table is updated. This process is described below.

- Oracle Business Analytics Warehouse finds the records to be deleted in the base table since the last ELT run, and loads them into the `W_SALES_ORDER_LINE_TMP` table. The measures in these records are multiplied by (-1). The mapping responsible for this task is SIL_SalesOrderLinesAggregate_Derive_PreSoftDeleteImage, which is run before SIL_SalesOrderLinesFact_SoftDelete deletes the records from the base table.
- Oracle Business Analytics Warehouse finds the records to be updated in the base table since the last ELT run, and loads them into the `W_SALES_ORDER_LINE_TMP` table. The measures in these records are multiplied by (-1). The mapping responsible for this task is SIL_SalesOrderLinesFact_Derive_PreLoadImage, which is run before SIL_SalesOrderFact updates the records from the base table.
- Oracle Business Analytics Warehouse finds the inserted or updated records in the base table since the last ELT run, and loads them into the W_SALES_ORDER_LINE_TMP table, without changing their sign. The mapping responsible for this task is SIL_SalesOrderLinesFact_Derive_PreLoadImage, which is run before PLP_SalesOrderLinesFact_Derive_PostLoadImage updates or inserts records in the base table.
- Oracle Business Analytics Warehouse uses the PLP_SalesOrderLinesAggregate_Derive mapping to aggregate the `W_SALES_ORDER_LINE_TMP` table and load to W_SALES_ORDER_LINE_A_TMP, which has the same granularity as the W_SALES_ORDER_LINE_A table.
- W_SALES_ORDER_LINE_A_TMP looks up the W_SALES_ORDER_LINE_A aggregate table to update existing buckets or insert new buckets in the aggregate table (the mapping is PLP_SalesOrderLinesAggregate_Load).

8.2.11.2.3 How to Configure the Sales Invoice Lines and Sales Order Lines Aggregate Tables To configure the Sales Invoice Lines or Sales Order Lines Aggregate Tables:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Application Specific tab, and select Order Management Analytics from the **Select BI Application** field.
4. Locate the following parameter and use the **Parameter Value** field to set the value:
   - TIME_GRAIN
5. Save your changes.

   For more information about specifying parameter values, see Section 4.6.3.3.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

8.2.11.3 About Tracking Multiple Products for Oracle Supply Chain and Order Management Analytics

The Sales Order Lines table contains two columns, `ORDHD_KEY_ID` and `ORDLN_KEY_ID`, that track individual products when they are grouped and sold as a single package. These two columns allow you to analyze the relationship of all products sold as a single unit. The `ORDHD_KEY_ID` column stores the Order ID of the entire sales order. The `ORDLN_KEY_ID` column stores the Line Item ID of the parent product.

For example, assume a customer purchases a package that includes a computer, scanner, and printer. In addition, the customer purchases a monitor separately. In this case, there are two parent items: the package and the monitor. The computer, scanner, and printer are all child orders of the parent order *package*, while the parent order *monitor* is a single-item purchase.

Your data warehouse may store this sales information in the Sales Order Lines table as seen in Table 8–2. The `ORDLN_KEY_ID` field contains the Line Item ID of the parent product in order to maintain the relationship between the parent and child products in a package. In this example, the `ORDLN_KEY_ID` field is Line_1 for each of the three child products (A1, A2, A3) that were sold as a part of the parent package, Parent A.

Table 8–2 Sales Order Table Columns With Parent/Child Relationships

Key_ID	SALES_ ORDER_NUM	PRODUCT _ID	ORDHD_ KEY_ID	ORDLN_KEY _ID	Relationship (Not a column in the table.)
Line_1	1000	Package	1000	Line_1	Parent A
Line_2	1000	Computer	1000	Line_1	Child A1
Line_3	1000	Scanner	1000	Line_1	Child A2
Line_4	1000	Printer	1000	Line_1	Child A3
Line_5	1000	Monitor	1000	Line_5	Parent B (no children)

In contrast, if each of the four items described in Table 8–2 were bought individually, the ORDLN_KEY_ID would have a different Line Item ID for every row. In this case, the Sales Order Lines table would look like Table 8–3.

Table 8–3 Sales Order Table Columns Without Parent/Child Relationships

Key_ID	SALES_ORDER_ NUM	PRODUCT_ ID	ORDHD_ KEY_ID	ORDLN_ KEY _ID	Relationship (Not a column in the table.)
Line_1	1000	Computer	1000	Line_1	None
Line_2	1000	Scanner	1000	Line_2	None
Line_3	1000	Printer	1000	Line_3	None
Line_4	1000	Monitor	1000	Line_4	None

8.2.11.4 Adding Dates to the Order Cycle Time Table for Post-Load Processing

To add more dates, you need to understand how the Order Cycle Times table is populated. Thus, if you want to change the dates loaded into the Order Cycle Time table (`W_SALES_CYCLE_LINE_F`), then you have to modify the `PLP_ SalesCycleLinesFact_Load and PLP_SalesCycleLinesFact_Load_Full` mappings that take the dates from the `W_*` tables and load them into the Cycle Time table.

To add dates to the Cycle Time table load:

1. In ODI Designer, open the PLP folder.
2. Modify the table definition for the target table to verify that it has a field to store this date.

   For example, if you are loading the Validated on Date in the W_SALES_CYCLE_ LINE_F table, then you might create a new column, VALIDATED_ON_DT, and modify the target definition of the W_SALES_CYCLE_LINE_F table.

3. Depending on which source table you want to extract from (see table below), you need to open the corresponding interfaces to add the columns in the target datastore.

Table 8–4 Source tables and associated Interfaces

Source Table	Full load interface	Incremental Load interface
W_SALES_ORDER_LINE_F	PLP_SalesCycleLinesFact_Load_Full.SQ_W_SALES_ORDER_LINE_F_FULL	PLP_SalesCycleLinesFact_Load.SQ_W_SALES_ORDER_LINE_F
W_SALES_INVOICE_LINE_F	PLP_SalesCycleLinesFact_Load_Full.SQ_W_SALES_ORDER_LINE_F_FULL.IVCLINE_FULL	PLP_SalesCycleLinesFact_Load.SQ_W_SALES_ORDER_LINE_F.IVCLINE
W_SALES_SCHEDULE_LINE_F	PLP_SalesCycleLinesFact_Load_Full.SQ_W_SALES_ORDER_LINE_F_FULL.SCHLINE_FULL	PLP_SalesCycleLinesFact_Load.SQ_W_SALES_ORDER_LINE_F.SCHLINE
W_SALES_PICK_LINE_F	PLP_SalesCycleLinesFact_Load_Full.SQ_W_SALES_ORDER_LINE_F_FULL.PICKLINE_FULL	PLP_SalesCycleLinesFact_Load.SQ_W_SALES_ORDER_LINE_F.PICKLINE

4. Modify all the interfaces inside the package PLP_SalesCycleLinesFact_Load to map the new column to the target W_SALES_CYCLE_LINE_F.

8.2.11.5 About Configuring the Backlog Period Date for Oracle Supply Chain and Order Management Analytics

The Backlog table (`W_SALES_BACKLOG_LINE_F`) stores backlog data for the current month. In contrast, the Backlog History table (`W_SALES_BACKLOG_LINE_F`) stores snapshots of all previous months' historical backlog data. The periods for which the Backlog History table tracks backlog data is defined by the Backlog Period Date. By default, the date is set as the last calendar day of the month; however you may configure this date. You may want to view backlog history at a more detailed level, such as by day or by week, instead of by month. The following example describes how historical backlog data is stored and what the implications are for changing the backlog time period.

8.2.11.6 Example of How Backlog Data Is Stored in the Backlog History Table

Assume you represent a manufacturing company where financial backlog is defined as any item that is ordered, but not invoiced. On February 1, 2001, you received an order (Sales Order #1) for 30 products. 20 were shipped and invoiced and 10 were shipped, but not invoiced. At the end of the day, there is an entry in the Backlog table and in the Backlog History table. The entry in the Backlog History table looks like that shown in Table 8–5.

Table 8–5 Oracle 11i and Oracle R12: Backlog History Table Entry as of February 1, 2001

SALES_ORDER_NUM(Sales Order Number)	BACKLOG _DK(Backlog Date)	BACKLOG_PERIOD_DK(Backlog Period Date)	OPEN_QTY(Backlog Quantity)
1	02/01/2001	02/28/2001	10

On February 2, 5 of the 10 financial backlog items are invoiced and, thus, removed from the backlog. Thus, there is an update to the existing row in the Backlog History table, as shown in Table 8–6.

Table 8–6 Oracle 11i and Oracle R12: Backlog History Table Entry as of February 2, 2001

SALES_ORDER_NUM (Sales Order Number)	BACKLOG _DK (Backlog Date)	BACKLOG_PERIOD_DK (Backlog Period Date)	OPEN_QTY (Backlog Quantity)
1	02/01/2001	02/28/2001	Old value: 10 New value: 5

No further activity happens until March 1st. On March 1st, the remaining 5 items on financial backlog are invoiced and removed from financial backlog. In addition, a new sales order (Sales Order #2) comes in for 50 new items. All of the items are put on financial backlog.

Even though all items from Sales Order #1 are cleared from financial backlog, the last backlog row remains in the Backlog History table. The purpose in retaining the last row is to indicate that there was backlog for this particular order. The quantity, in this case 5 items, does not tell you how many items were initially on backlog, which was 10.

For the 50 new financial backlog items, there is a new entry into the Backlog History table. So, as of February 28, 2001, the Backlog History table looks like the Table 8–7.

Table 8–7 Oracle 11i: Backlog History Table Entry as of February 28, 2001

SALES_ORDER_NUM (Sales Order Number)	BACKLOG _DK (Backlog Date)	BACKLOG_PERIOD_DK (Backlog Period Date)	OPEN_QTY (Backlog Quantity)
1	Old value: 02/01/2001 New value: 02/02/2001	02/28/2001	Old value: 10 New value: 5

On March 1, 30 more items are ordered (Sales Order #3), all of which are on financial backlog. The resulting Backlog History table looks like Table 8–8.

Table 8–8 Oracle 11i and Oracle R12: Backlog History Table Entry as of March 1, 2001

SALES_ORDER_NUM (Sales Order Number)	BACKLOG _DK (Backlog Date)	BACKLOG_PERIOD_DK (Backlog Period Date)	OPEN_QTY (Backlog Quantity)
1	Old value: 02/01/2001 New value: 02/02/2001	02/28/2001	5

Table 8–8 (Cont.) Oracle 11i and Oracle R12: Backlog History Table Entry as of March 1,

SALES_ORDER_NUM (Sales Order Number)	BACKLOG _DK (Backlog Date)	BACKLOG_PERIOD_DK (Backlog Period Date)	OPEN_QTY (Backlog Quantity)
2	03/01/2001	03/31/2001	50
3	03/01/2001	03/31/2001	30

Because backlog history is maintained at the monthly level, you have a partial history of your backlogs. Based on the latest state of the Backlog History table shown in Table 8–8, you can see that sales order number 1 ended up with 5 financial backlogged items. You do not have visibility into what the initial financial backlogged item quantities were for the sales orders; you only have their ending quantities.

If you decide that you want to track more details on how the items moved out of backlog, then you'll have to maintain the history at a more granular level. For instance, if you want to know the number of items that were on backlog when the it was first opened, you would have to track the backlog history by day, instead of by month.

For example, if you maintained backlog history at the daily level you would be able to capture that sales order 1 had an initial backlog of 10 as of February 1 and the backlog quantity shrank to 5 as of February 2. So, by capturing history at the daily level, you could then compute cycle times on how long it took to move items out of backlog. However, if you decide to capture backlog history at a more detailed level, you may compromise performance because tracking backlog history at the daily level can increase the size of the Backlog History table exponentially.

If you choose to change the time period for which historical backlog data is kept, you must verify that all types of backlog are being stored at the same grain; which requires modification to multiple mappings. Table 8–9 provides a list of all applicable mappings and their corresponding Expression transformations that you must modify.

Table 8–9 Oracle 11i and Oracle R12: Backlog History Applicable Mappings and Expression Transformations

Mapping	Expression Transformation
`PLP_SalesBacklogLinesfact_LoadOrderLines`	`EXP_SALES_ORNLNS_BACKLOG`
`PLP_SalesBacklogLinesfact_LoadScheduleLines`	`EXP_SALES_SCHLNS_BACKLOG`

The backlog history period is monthly by default. The default SQL statement in the Expression transformation for the port BACKLOG_PERIOD_DK is:

```
TO_DECIMAL(TO_CHAR(LAST_DAY(CALENDAR_DATE),'YYYYMMDD'))
```

You can edit the backlog period date so that you can capture a more detailed backlog history with the following procedure. Possible periods include daily (`CAL_DAY_DT`), weekly (`CAL_WEEK_DT`), monthly (`CAL_MONTH_DT`), and quarterly (`CAL_QTR_DT`).

8.2.11.7 About Configuring the Customer Status History Fact for Post-Load Processing In Oracle Supply Chain and Order Management Analytics

In Oracle Supply Chain and Order Management Analytics, W_CUSTOMER_STATUS_HIST_F is a fact table that tracks the status of customers based on the frequency of orders they place with the organization. Possible statuses are NEW, RECENT, DORMANT and LOST. The time duration for each status bucket is configurable, out of the box being a calendar year. The grain of this table is at a Customer, Customer Status and the Status Start Date level. This section explains the possible configurations available for this table, what they mean and how to implement them.

8.2.11.8 Configuring the Customer Status History Fact table

This section talks about the following configurations that are available for the Customer Status History Fact table:

- Configure the Data Warehouse Identifier
- Configure the Period for each status bucket

Configuring the Data Warehouse Identifier

This table uses some of the Oracle Business Intelligence Applications defined statuses, like NEW, RECENT, DORMANT and LOST. These status data gets loaded into the Data Warehouse directly through an out of box pre-packaged CSV file. The data in the file is independent of any specific OLTP source systems where your Customer or Sales data resides. In order to differentiate between source-based statuses from the pre-packaged out of box Data Warehouse statuses, a definite identifier is required. The mapping parameter WH_DATASOURCE_NUM_ID serves that purpose.

A pre-packaged value equal to 999 is set out of the box. Ideally you would not need to configure this value unless you have chosen to use this number (999) for a specific data source of yours, like Oracle EBS 11.5.10, etc.

For information about how to configure the WH_DATASOURCE_NUM_ID value, see:Section 8.2.11.9, "How to Configure the Customer Status History Fact table".

Configuring the Period for each status bucket

When a customer orders some products/services from your organization for the first time, Oracle Business Intelligence Applications sets the status for the customer as NEW. The customer maintains the same status if he/she shows a constant order pattern, as long as the duration between any two of his/her orders is less than a configurable/business defined period. The value (out of box being 365 days) of this parameter PERIOD is configurable. An use case for that would be a Fast Moving / Retail Goods company many define 30 days as their choice of period, whereas a Slow Moving company may be even happy with 730 days as period.

In case the customer is seen to have not ordered anything for more than one period, he/she is moved to the next status, RECENT. Similarly, no orders for one more period since he/she became RECENT would make him/her DORMANT. And lastly, he/she is set to LOST if no orders were seen for more than one more period after he/she became DORMANT.

However, if a customer orders while in DORMANT status, for instance, Oracle Business Intelligence Applications would upgrade his/her status back to RECENT. If the customer were in LOST status, and he/she orders, then he/she will be upgraded back to RECENT.

All these examples above illustrate how important it is for the business to set the right value for the period. Organizations would tend to launch different campaigns targetting to different customers based on their current status, or order-patterns, putting it in a different way.

For information about how to configure the PERIOD parameter, see:Section 8.2.11.9, "How to Configure the Customer Status History Fact table".

8.2.11.9 How to Configure the Customer Status History Fact table

This section explains how to configure the Customer Status History Fact table using the WH_DATASOURCE_NUM_ID and PERIOD parameters (for more information about these variables, see Section 8.2.11.8, "Configuring the Customer Status History Fact table").

To configure the Customer Status History Fact table:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Application Specific tab, and select Order Management Analytics from the **Select BI Application** field.
4. Locate the following parameter and use the **Parameter Value** field to set the value:
   - PERIOD
5. Display the Global tab
6. Locate the following parameter and use the **Parameter Value** field to set the value:
   - WH_DATASOURCE_NUM_ID
7. Save your changes.

   For more information about specifying parameter values, see Section 4.6.3.3.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

8.2.11.10 About Configuring the Inventory Monthly Balance Table

To configure the Inventory Monthly Balance (`W_INVENTORY_DAILY_BALANCE_F_A1`) aggregate table, you need to consider the aggregation level, the time period to update the aggregation, and the time period to keep records in the Inventory Balance table.

You need to configure three parameters to configure the Inventory Monthly Balance table:

- GRAIN
- KEEP_PERIOD
- NUM_OF_PERIOD

The GRAIN parameter has a preconfigured value of Month. The possible values for the GRAIN parameter are:

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

The KEEP_PERIOD parameter has a preconfigured value of Month. Values for the KEEP_PERIOD parameter include:

- DAY
- WEEK

- MONTH
- QUARTER
- YEAR

The NUM_OF_PERIOD parameter has a preconfigured value of 3. The value for the NUM_OF_PERIOD parameter is a positive integer, for example, 1, 2, 3, and so on.

8.2.11.11 How to Configure the Inventory Monthly Balance

Before you run the initial ELT session or incremental ELT sessions to load the Inventory Monthly Balance table, configure the Inventory Monthly Balance as follows.

To configure the Inventory Monthly Balance:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Application Specific tab, and select Supply Chain Analytics from the **Select BI Application** field.
4. Locate the following parameters and use the **Parameter Value** field to set the value:
   - GRAIN (default is MONTH, for PLP_InventoryMonthlyBalance)
   - KEEP_PERIOD (default is MON, for PLP_InventoryDailyBalance_Trim)
   - NUM_OF_PERIOD (default is 3, for PLP_InventoryDailyBalance_Trim)
5. Save your changes.

   For more information about specifying parameter values, see Section 4.6.3.3.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

To incrementally refresh the Inventory Monthly Balance table:

1. Run the PLP_InventoryMonthlyBalance interface to delete the records from the Monthly Balance (W_INVENTORY_MONTHLY_BAL_F) aggregate table for the time period that you want.

   The GRAIN parameter determines the time period for the deletion. For example, if GRAIN=MONTH, and the date is May 15, 2005, then all records for April and the current month (May) are deleted in the Monthly Balance (W_INVENTORY_MONTHLY_BAL_F) table.

2. Run the PLP_InventoryMonthlyBalance interface to retrieve the records in the Inventory Balance (W_INVENTORY_DAILY_BALANCE_F) fact table and load the records to the Monthly Balance (W_INVENTORY_MONTHLY_BAL_F) table at the required level of granularity.

   For example, if GRAIN=MONTH, then the month end balance records in the W_INVENTORY_DAILY_BALANCE_F fact table are stored in and aggregated to theMonthly Balance (W_INVENTORY_MONTHLY_BAL_F). Running the S_M_PLP_INV_BALANCE_A1_AGG session, and the M_PLP_INV_BALANCE_A1_AGG mapping implements this step. For the current month balance, balance records of the previous day (if it is in the same month) are deleted from W_INVENTORY_MONTHLY_BAL_F, and balance records of the current day will be loaded from W_INVENTORY_BALANCE_F to W_INVENTORY_MONTHLY_BAL_F.

3. Run the PLP_InventoryDailyBalance_Trim interface to the old records from the W_INVENTORY_DAILY_BALANCE_F fact table.

   To remove old records you need to use the KEEP_PERIOD and the NUM_OF_PERIOD parameters. For example, if KEEP_PERIOD=MONTH, NUM_OF_PERIOD=1, and the date is May 15, 2005, then the records for April and the current month (May) are kept and the older records are deleted.

8.2.11.12 About Configuring the Product Transaction Aggregate Table

There are two aggregation scenarios to configure the Product Transaction aggregate (`W_PRODUCT_XACT_A`) table—the initial ELT run and then the incremental ELT run.

For your initial ELT run, you need to configure the aggregation level, and the length of history kept in the Product Transaction fact table.

For your initial ELT run, you need to configure the aggregation grain, using the GRAIN parameter.

For the incremental ELT run, you need to configure the aggregation level, the update period in aggregation, and the length of history kept in the Product Transaction fact table, using the following parameters:

- GRAIN

  The GRAIN parameter specifies the aggregation level. Valid values are DAY, WEEK, MONTH (preconfigured value), QUARTER, YEAR.

- REFRESH_PERIOD

  The REFRESH_PERIOD parameter, together with NUM_OF_PERIOD, indicates the number of period of records that will be refresh from the transaction table to the aggregate table. Valid values are DAY, WEEK, MONTH (preconfigured value), QUARTER, YEAR.

- NUM_OF_PERIOD

  The NUM_OF_PERIOD parameter, together with REFRESH_METHOD, indicates the number of period of records that will be refresh from the transaction table to the aggregate table. Valid values are positive integers, for example, 1, 2, 3 (preconfigured value).

8.2.11.13 How to Configure the Product Transaction Aggregate Table

Before you run the initial ELT and then the incremental ELT to load the Product Transaction aggregate table, you need to configure the Product Transaction Aggregate Table, as follows.

To configure the Product Transaction Aggregate Table:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Application Specific tab, and select Supply Chain Analytics from the **Select BI Application** field.
4. Locate the following parameters and use the **Parameter Value** field to set the value:
   - REFRESH_PERIOD (default is MONTH, for PLP_ProductTransactionAggregate)

- GRAIN (default is MONTH, for PLP_ProductTransactionAggregate and PLP_ProductTransactionAggregate_Full)
- NUM_OF_PERIOD (default is 3, for PLP_ProductTransactionAggregate)

5. Save your changes.

   For more information about specifying parameter values, see Section 4.6.3.3.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

To configure the Product Transaction aggregate table for the initial ELT run:

1. Run the PLP_ProductTransactionAggregate_Full workflow to retrieve the records in the Product Transaction fact (W_PRODUCT_XACT_F) table, and aggregate the records to the Product Transaction aggregate (W_PRODUCT_XACT_A) table for the granularity that you want.

   For example, if GRAIN=MONTH then the records in the W_PRODUCT_XACT_F fact table are retrieved and aggregated to the W_PRODUCT_XACT_A table at a monthly level.

To configure the Product Transaction aggregate table for the incremental ELT run:

1. Run the PLP_ProductTransactionAggregate workflow to delete the refreshed records from the Product Transaction aggregate (W_PRODUCT_XACT_A) table for the time period that you want.

   The REFRESH_PERIOD and the NUM_OF_PERIOD parameters determine the time period for the deletion. For example, if REFRESH_PERIOD=MONTH, NUM_OF_PERIOD=1, and the date is May 15, 2005, then all records for April and the current month (May) are deleted in the W_PRODUCT_XACT_A table.

2. Run the PLP_ProductTransactionAggregate workflow to retrieve the records in the Product Transaction fact (W_PRODUCT_XACT_F) table, and aggregate the records to the W_PRODUCT_XACT_A table at the level of granularity that you want.

   For example, if GRAIN=MONTH, then the records in the W_PRODUCT_XACT_F fact table are retrieved and aggregated to the W_PRODUCT_XACT_A table at a monthly level.

8.2.11.14 How to Configure Sales Order Lines Data Storage

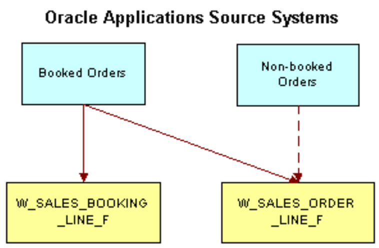
Sales order lines are the itemized lines that make up a sales order. This information is stored in the `W_SALES_ORDER_LINE_F` table. This topic describes how to modify the type of information stored in this table.

8.2.11.15 About the Handling of Booked and Nonbooked Orders in the Order Lines and Bookings Table

By default, only booked orders are extracted from the Oracle source system as shown in Figure 8–1. Therefore, all orders loaded into the Sales Order Lines and Bookings tables are booked.

However, if you want to load nonbooked orders into the Sales Order Lines table, you have to configure the extract so that it does not filter out nonbooked orders. In Oracle 11i and Oracle R12, the `OE_ORDER_LINES_ALL.BOOKED_FLAG = Y` condition indicates that an order is booked; therefore, this statement is used to filter out nonbooked orders. To load all orders, including nonbooked orders, remove the filter condition from the WHERE clause in the SDE_ORA_SalesOrderLinesFact and SDE_ORA_SalesOrderLinesFact_Primary mappings.

Figure 8–1 Handling Booked and Nonbooked Orders



By default, only booked orders are loaded into the Sales Order Lines (W_SALES_ORDER_LINES_F) and Sales Booking Lines (W_SALES_BOOKING_LINE_F) tables. However, you can also load non-booked orders in Sales Order Lines (W_SALES_ORDERS_LINES_F).

To include nonbooked orders in the Sales Order Lines tables:

1. In ODI Designer, open the SDE_ORA11510_Adaptor folder.
2. Open the Interface SDE_ORA_SalesOrderLinesFact_Full.SQ_BCI_SALES_ORDLNS_FULL.
3. Edit the filter attached to the source table OE_ORDER_LINES_All.
4. On the Implementation tab, remove the line 'OE_ORDER_LINES_ALL.BOOKED_FLAG='Y''.
5. Click OK to save the changes.
6. Repeat steps 2 - 5 for the following interfaces:
   - SDE_ORA_SalesOrderLinesFact.SQ_BCI_SALES_ORDLNS
   - SDE_ORA_SalesOrderLinesFact_Primary.SALES_ORDER_LINE_F_PE

8.2.11.16 How to Configure Sales Schedule Lines Data Storage

Sales schedule lines detail when each order's items are slated for shipment. Each sales order is broken into sales order lines, and each sales order line can have multiple schedule lines.

For example, you might not have enough stock to fulfill a particular sales order line, therefore you create two schedules to fulfill it. One schedule ships what you currently have in stock, and the other schedule includes enough time for you to manufacture and ship the remaining items of the sales order line. This information is stored in the `W_SALES_SCHEDULE_LINE_F` table. This topic describes how to modify the type of information stored in this table.

8.2.11.17 About the Handling of Booked and Nonbooked Orders in the Sales Schedule Lines Table

By default, all orders loaded into the Sales Schedule Lines are booked.

However, if you want to load nonbooked orders into the Sales Schedule Lines table, you have to configure the extract so that it does not filter out nonbooked orders. In Oracle 11i and Oracle R12, the `OE_ORDER_LINES_ALL.BOOKED_FLAG = Y` condition indicates that an order is booked; therefore, this statement is used to filter out nonbooked orders. To load all orders, including nonbooked orders, remove the filter condition from the WHERE clause in the SDE_ORA_SalesScheduleLinesFact and SDE_ORA_SalesScheduleLineLines_Fact_Primary mappings.

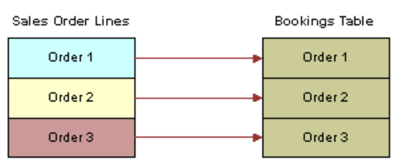
To include nonbooked orders in the Sales Schedule Lines tables:

1. In ODI Designer, open the SDE_ORA11510_Adaptor folder.
2. Open the Interface SDE_ORA_SalesScheduleLinesFact_Full.SQ_BCI_SALES_ORDLNS_FULL.
3. Edit the filter attached to the source table OE_ORDER_LINES_All.
4. On the Implementation tab, remove the line 'OE_ORDER_LINES_ALL.BOOKED_FLAG='Y''.
5. Click OK to save the changes.
6. Repeat steps 2 - 5 for the following interfaces:
   - SDE_ORA_SalesScheduleLinesFact.SQ_BCI_SALES_ORDLNS
   - SDE_ORA_SalesScheduleLinesFact_Primary.SALES_SCHEDULE_LINE_F_PE

8.2.11.18 About Loading Bookings at the Schedule Line Level

As initially configured for Oracle 11i, bookings are recorded at the Sales Order Line level. For each booked order, there is at least one row in the Bookings table, as shown in the figure below.

Figure 8–2 Sales Order Lines and Bookings Table



In the master package, the following two steps are used to populate W_SALES_BOOKING_LINE_F:

- 3_Master_SIL_Facts_OM_BkgLn_Credit
- 3_Master_SIL_Facts_OM_BkgLn_Debt

Within each step, you can choose either order line level or schedule line level, as described in the table below:

Table 8–10 Packages and associated Line Level

Package	Order Line Level	Schedule Line Level
3_Master_SIL_Facts_OM_BkgLn_Credit	SIL_SalesBookingLinesFact_Load_OrderLine_Credit	SIL_SalesBookingLinesFact_Load_ScheduleLine_Credit
3_Master_SIL_Facts_OM_BkgLn_Debt	SIL_SalesBookingLinesFact_Load_OrderLine_Debt	SIL_SalesBookingLinesFact_Load_ScheduleLine_Debt

Out-of-the-box, Oracle Business Intelligence Applications populates booking lines at the order line level. That is, the LOAD_BOOKINGLINE_ORDER parameter is set to 'Y' and the LOAD_BOOKINGLINE_SCHEDULE parameter is set to 'N'. If you want to load booking lines at the schedule level, change the LOAD_BOOKINGLINE_ORDER parameter value to 'N' and the LOAD_BOOKINGLINE_SCHEDULE parameter value to 'Y'.

Bookings may be recorded at the Sales Schedule Line level instead of the Sales OrderLine level. At the Sales Schedule Line level, bookings provide a more granular view, as the orders are segmented by schedule line. Bookings recorded at the Schedule Line level provide one row in the Bookings table for each schedule line. Oracle

Applications schedule lines have the same granularity as order lines. Therefore, if you pull booking lines from schedule lines, the booking lines are limited to scheduled order lines.

8.2.11.19 How to Configure Early and Late Tolerances for Shipments

You configure the definition of early and late shipments by editing the `EXP_SALES_PCKLNS` expression in the `mplt_SA_ORA_SalesPickLinesFact` Interface. The `mplt_SA_ORA_SalesPickLinesFact` Interface is used by the `SDE_ORASalesPickLinesFact` mapping.

This Interface compares the pick date and ship date with the scheduled ship date to determine whether or not orders are late.

To configure early and late tolerances for shipments:

1. In ODI Designer, open the SDE_ORA11510_Adaptor folder.
2. Launch the Oracle BI Applications Configuration Manager, and set the value of the following parameters:
   - PICK_EARLY_TIME_TOL (0, for SDE_ORA_SALESPICKLINESFACT)
   - PICK_LATE_TIME_TOL (0, for SDE_ORA_SALESPICKLINESFACT)
   - SHIP_EARLY_TIME_TOL(0, for SDE_ORA_SALESPICKLINESFACT)
   - SHIP_LATE_TIME_TOL (0, for SDE_ORA_SALESPICKLINESFACT)

   For example:

   – To allow two days after the scheduled pick date before you flag the pick as late, set the value of parameter PICK_LATE_TIME_TOL to 2.

   – To set the number of days before a pick is flagged as early, set the value of parameter PICK_EARLY_TIME_TOL.

   – To set the number of days before a pick is flagged as late, set the value of parameter PICK_LATE_TIME_TOL.

   – To change the shipping tolerances, set the value of parameters SHIP_LATE_TIME_TOL or SHIP_EARLY_TIME_TOL.

8.2.11.20 How to Configure Sales Invoice Lines Data Storage

Sales invoice lines are payments for items ordered by a customer. This information is stored in the `W_SALES_INVOICE_LINE_F` table. This topic describes how to modify the type of information stored in this table.

8.2.11.21 How to Configure the Sales Invoice Extract

By default, the Oracle Supply Chain and Order Management Analytics application is configured to extract completed sales invoices when performing the Sales Invoice data extract. Oracle 11i and Oracle R12 use a flag to indicate whether a sales invoice is complete. In particular, completed sales invoices are those where the `RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG = Y` in Oracle 11i.

To extract incomplete sales invoices, as well as complete invoices, remove the extract filter statement.

To remove the extract filter for sales invoices:

1. In ODI Designer, open the SDE_ORA11510_Adaptor folder.

2. Open the Interface SDE_ORA_SalesInvoiceLinesFact_Full.SQ_BCI_SALES_IVCLNS_FULL.
3. Edit the filter attached to the source table RA_CUSTOMER_TRX_ALL.
4. On the Implementation tab, remove the line 'RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG='Y''.
5. Click OK to save the changes.
6. Repeat steps 2 - 5 for the following interfaces:
   - SDE_ORA_SalesInvoiceLinesFact.SQ_BCI_SALES_IVCLNS
   - SDE_ORA_SalesInvoiceLinesFact_Primary.SQ_BCI_SALES_IVCLNS

8.2.11.22 How to Configure Procurement and Spend Analytics for Oracle Supply Chain and Order Management Analytics

The Oracle Supply Chain and Order Management Analytics application uses tables that are also used in Oracle's Procurement and Spend Analytics Family of Products.

For Oracle 11i and Oracle R12, you need to use the following configuration steps for Procurement and Spend Analytics to configure Oracle Supply Chain and Order Management Analytics:

- Section 5.1.6.2, "Configuring the Names of Country Region, State Region, State, or Region"
- Section 5.1.6.3, "Configuring the Configuring the Make-Buy Indicator"

8.2.11.23 How to Configure Oracle Financial Analytics for Oracle Supply Chain and Order Management Analytics

The Oracle Supply Chain and Order Management Analytics application uses tables that are also used in the Oracle Financial Analytics application.

For Oracle 11i and Oracle R12, you need to use the following configuration steps for Oracle Financial Analytics to configure Oracle Supply Chain and Order Management Analytics:

- Section 7.2.4.7, "How to Configure AR Balance ID for Oracle Receivables Analytics and Oracle General Ledger and Profitability Analytics"
- Section 7.2.4.8, "How to Configure the AR Adjustments Extract for Oracle Receivables Analytics"
- Section 7.2.4.9, "How to Configure the AR Schedules Extract"
- Section 7.2.4.10, "How to Configure the AR Cash Receipt Application Extract for Oracle Receivables Analytics"
- Section 7.2.4.11, "How to Configure the AR Credit-Memo Application Extract for Oracle Receivables Analytics"

8.2.11.24 About Tracking Attribute Changes in Bookings

Changes in booked orders are tracked in the Booking Lines table (W_SALES_BOOKING_LINE_F), not in the Sales Order Lines table (W_SALES_ORDER_LINE). By default, the only changes tracked in the W_SALES_BOOKING_LINE_F table are changes in the ordered amount, ordered quantity, or Booking ID. By default, the Booking ID is defined as:

```
TO_CHAR(SQ_BCI_SALES_ORDLNS.LINE_ID)||'~'||TO_CHAR(SQ_BCI_SALES_ORDLNS.INVENTORY_
```

```
ITEM_ID)||'~'||to_char(SQ_BCI_SALES_ORDLNS.SHIP_FROM_ORG_ID)
```

Any changes in these fields results in another row in the `W_SALES_BOOKING_LINE_F` table. However, changes in any other fields does not result in a new row; instead, the existing information are overwritten with the changed information. No history is kept for changes to these other field values. If you want to track other changes you can do so. For example, you may want to track changes to the sales representative who is handling the order. The ELT processes are prepackaged to overwrite sales representative changes; however, if you want to retain them, you must add the attribute to the Booking ID definition in the Booking ID expression in the Source Adapter Interface (`mplt_SA_ORA_SalesOrderLinesFact`). The following section describes what happens if you modify the Booking ID to include the sales representative.

8.2.11.25 About Viewing the Data Warehouse Changes by Salesperson ID

Assume you want to track changes to the sales representative for bookings and de-bookings. You decide to do this to better evaluate each representative's sales performance. To track changes by Salesperson ID, you have to change the expression of BOOKING_ID to:

```
TO_CHAR(SQ_BCI_SALES_ORDLNS.LINE_ID)||'~'||TO_CHAR(SQ_BCI_SALES_ORDLNS.INVENTORY_
ITEM_ID)||'~'||to_char(SQ_BCI_SALES_ORDLNS.SHIP_FROM_ORG_ID)
```

For example, to edit the BOOKING_ID value, do the following:

1. In ODI Designer, open the interface SDE_ORA_SalesOrderLinesFact_Full.SALES_ORDER_LINE_FS_FULL.
2. Edit the column BOOKING_ID on target datastore W_SALES_ORDER_LINE_FS.
3. In the implementation tab, edit the expression.
4. Click OK and save the changes.
5. Repeat the steps 2-4 for the Interface SDE_ORA_SalesOrderLinesFact.SALES_ORDER_LINE_FS.

   If you capture booking lines on schedule line level (10.2.5.2.5), please repeat the steps 2-4 for the following interfaces:

   - SDE_ORA_SalesScheduleLinesFact_Full.SALES_SCHEDULE_LINE_FS_FULL
   - SDE_ORA_SalesScheduleLinesFact.SALES_SCHEDULE_LINE_FS

The following paragraphs and tables describe what happens in the source system and the `W_SALES_BOOKING_LINE_F` table when you change sales representatives under this scenario.

Day 1: One order is placed with Salesperson 1001. The source system displays the information as shown in Table 8–11.

Table 8–11 Oracle 11i and Oracle R12: Source System Table Row After Day One Activity

Sales Order Number	Sales Order Line Number	Salesperson ID	Quantity	Selling Price	Date
1	1	1001	100	25	1-June-2000

The row in Table 8–11 is entered into the IA Bookings table (`W_SALES_BOOKING_LINE_F`) as shown in Table 8–12.

Table 8–12 Oracle 11i and Oracle R12: W_SALES_BOOKING_LINE_F Table Row After Day One Activity

SALES_ ORDER_NUM	SALES_ ORDER_ ITEM	SALESREP _ID	SALES _QTY	NET_ DOC_AMT	BOOKED_ ON_DT
1	1	1001	100	2500	1-June-2000

Day 2: Salesperson 1002 takes over this order, replacing Salesperson 1001. Thus, the salesperson associated with the order is changed from 1001 to 1002 in the source system. The row in the source system looks like the row shown in Table 8–13.

Table 8–13 Oracle 11i and Oracle R12: Source System Table Row After Day Two Activity

Sales Order Number	Sales Order Line Number	Salesperson ID	Quantity	Selling Price	Date
1	1	*1002*	100	25	2-June-2000

The packages SIL_SalesBookingLinesFact_Load_OrderLine_Credit or SIL_SalesBookingLinesFact_Load_ScheduleLine_Credit, which also writes to the booking line table, now does a de-booking for the old line, and the packages SIL_SalesBookingLinesFact_Load_OrderLine_Debt or SIL_SalesBookingLinesFact_Load_ScheduleLine_Debt inserts a new row into the W_SALES_BOOKING_LINE_F booking table. On day two, the row in the W_SALES_BOOKING_LINE_F table looks like the row shown in the Table 8–14.

Table 8–14 Oracle 11i and Oracle R12: W_SALES_BOOKING_LINE_F Table Row After Day Two Activity

SALES_ ORDER_NUM	SALES_ ORDER_ ITEM	SALESREP_ ID	SALES _QTY	NET_ DOC_ AMT	BOOKED_ ON_DT
1	1	1001	100	2500	1-June-2000
1	1	1001	-100	-2500	2-June-2000
1	1	1002	100	2500	2-June-2000

8.2.11.26 How to Configure Different Types of Backlog Calculations

Backlog information is stored in the `W_SALES_BACKLOG_LINE_F` and W_SALES_BACKLOG_HISTORY_F tables. This topic describes how to modify the type of information stored in these tables. Many types of backlog exist in the Oracle Supply Chain and Order Management Analytics application—financial backlog, operational backlog, delinquent backlog, scheduled backlog, unscheduled backlog, and blocked backlog. Each type of backlog is defined by two particular dates in the sales process; therefore, calculations of backlog hits multiple fact tables.

For example, financial backlog records which items have been ordered but payment has not been received. Thus, to calculate the number of financial backlog items, you use the Sales Order Lines table (to determine which items have been ordered) and the Sales Invoice Lines table (to see which orders have been paid for). Using these two tables, you can determine the number of items and the value of those items that are on financial backlog.

8.2.11.27 Adding Closed Orders to Backlog Calculations

By default, the Oracle Supply Chain and Order Management Analytics application only extracts open sales orders from the Sales Order Lines (W_SALES_ORDER_LINE_F) table and Sales Schedule Lines table (`W_SALES_SCHEDULE_LINE_F`) for backlog calculations to populate the Backlog tables. *Open sales orders* are defined as orders that are not canceled or not complete. The purpose in extracting only open orders is that in most organizations those orders that are closed are no longer a part of backlog. However, if you want to extract sales orders that are marked as closed, you may remove the default filter condition from the extract mapping.

For example, assume your customer orders ten items. Six items are invoiced and shipped, but four items are placed on operational and financial backlog. This backlog status continues until one of two things happens:

- The items are eventually shipped and invoiced.
- The remainder of the order is canceled.

If you choose to extract sales orders that are flagged as closed, you must remove the condition in the Backlog flag. To do so, use the following procedure.

The `BACKLOG_FLAG` in the `W_SALES_ORDER_LINE_F` table is also used to identify which sales orders are eligible for backlog calculations. By default, all sales order types have their Backlog flag set to `Y`. As a result, all sales orders are included in backlog calculations.

To remove open order extract filters:

1. In ODI Designer, open the SDE_ORA11510_Adaptor folder.
2. Open the Interface SDE_ORA_SalesOrderLinesFact_Full.SALES_ORDER_LINE_FS_FULL.
3. Click the column on target datastore, and display the implementation tab.
4. Edit the expression of OPR_BACKLOG_FLG and remove the ‘SQ_BCI_SALES_ORDLNS.OPEN_FLAG = 'Y' AND '.
5. Edit the expression of FIN_BACKLOG_FLG and remove the ‘SQ_BCI_SALES_ORDLNS.OPEN_FLAG = 'Y' AND '.
6. Click OK and save your changes to the repository.
7. Repeat step 2-5 for the following Interfaces:
   - SDE_ORA_SalesOrderLinesFact.SALES_ORDER_LINE_FS
   - SDE_ORA_SalesScheduleLinesFact_Full.SALES_SCHEDULE_LINE_FS_FULL
   - SDE_ORA_SalesScheduleLinesFact.SALES_SCHEDULE_LINE_FS
8. Open the PLP folder.
9. Open the Interface PLP_SalesBacklogLinesFact_Load_OrderLines.SQ_SALES_ORER_LINES_BACKLOG
10. Remove the filter having expression of W_STATUS_CODE <> 'Closed'.
11. Click OK and save your changes to the repository.
12. Repeat steps 2-5 for following Interfaces:
    - PLP_SalesBacklogLinesFact_Load_OrderLines.SALES_BACKLOG_LINE_F1
    - PLP_SalesBacklogLinesFact_Load_ScheduleLines.SQ_W_SALES_SCHEDULE_LINE_F

- PLP_SalesBacklogLinesFact_Load_ScheduleLines.SALES_BACKLOG_LINE_F1

9 Configuring Oracle Human Resources Analytics

This chapter describes how to configure Oracle Human Resources Analytics for particular sources to meet your business needs, and contains the following topics:

- Section 9.1, "Overview of Oracle Human Resources Analytics"
- Section 9.2, "Configuration Required Before A Full Load for Oracle HR Analytics"

To find out about other possible tasks required to deploy Oracle Business Intelligence Applications, see Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle Business Intelligence Applications With ODI".

9.1 Overview of Oracle Human Resources Analytics

Oracle Human Resources contains information for HR operations, employee events, and payroll.

The Oracle HR Analytics application has the following functional areas:

- **Compensation**. HR Compensation allows you to analyze the salaries, benefits, and rewards that comprise your employee compensation plan. The metrics provided as part of the application allow you to measure several areas of performance and perform a variety of comparative analyses at various levels of granularity.

  It provides your company with employee payroll information that can be vital to success in today's economy. Over or under-compensating employees can both have serious effects on your company's ability to maintain a competitive edge. The HR Compensation area provides the information your HR Management department needs to manage compensation costs, such as identifying emerging trends within the organization, or within specific areas of compensation, and evaluating the effectiveness of the level of compensation as an incentive.

- **Human Resource Performance.** The information stored in the Human Resource Performance area allows you to measure several areas of performance, including contribution and productivity, HR effectiveness, and trends analytics.

- **Retention.** Under the Retention functional area you can find the events that are the hallmarks of employees' professional life cycle. These events include their hiring information, their promotional opportunities realized and not realized, the quality of the employees' job performance as measured by performance ranking, their length of service, and the reasons for termination, both voluntary and involuntary. Monitoring retention rates within departments is useful in determining potential problem areas that may want to be addressed by senior management.

- **U.S. Statutory Compliance**. The U.S. Statutory Compliance functional area stores information that helps Human Resources departments prepare government-required reports.
- **Workforce Profile.** The Workforce Profile functional area provides you with the tools to separate sensitive from non-sensitive information, and to restrict access to sensitive data. Sensitive information includes such data as ethnicity, age, native language, marital status, and performance ratings. Non-sensitive information includes information such as job title, work location, and position status.

9.2 Configuration Required Before A Full Load for Oracle HR Analytics

This section contains configuration steps that you need to perform on Oracle HR Analytics before you do a full data load, and contains the following topics:

- Section 9.2.1, "About Domain Values and CSV Worksheet Files for Oracle HR Analytics"
- Section 9.2.2, "How to Configure the Employee Ethnic Group Codes"
- Section 9.2.3, "How to Configure the Employee Sex Codes"
- Section 9.2.4, "How to Configure Employee Veteran Status Codes"
- Section 9.2.5, "How to Configure the Employment Category Codes"
- Section 9.2.6, "How to Configure Employment Exempt Status"
- Section 9.2.7, "How to Configure Employment Full Time Status"
- Section 9.2.8, "How to Configure Employment Status"
- Section 9.2.9, "How to Configure Event Types"
- Section 9.2.10, "How to Configure HR Active Position Status"
- Section 9.2.11, "How to Configure HR Position Exempt Status"
- Section 9.2.12, "How to Configure Job EEO Category Codes"
- Section 9.2.13, "How to Configure Job FLSA Status Codes"
- Section 9.2.14, "How to Configure Pay Type Group Codes"
- Section 9.2.15, "How to Configure the Pay Type Flag"
- Section 9.2.16, "How to Configure Address Types for HR Profile"
- Section 9.2.17, "How to Configure Phone Types for HR Profile"
- Section 9.2.18, "How to Configure Education Degree Codes for Employee Dimension"
- Section 9.2.19, "About Configuring Key Flexfields"
- Section 9.2.20, "How to Configure the Key Flexfields for the Job Dimension"
- Section 9.2.21, "How to Configure the Key Flexfields for the HR Position Dimension"
- Section 9.2.22, "How to Configure the Key Flexfields for the Pay Grade Dimension"
- Section 9.2.23, "How to Configure multi-segmented Flexfields"
- Section 9.2.24, "How to Configure Flags for the Pay Type Dimension"
- Section 9.2.25, "How to Configure Classification Names for Payroll"

- Section 9.2.26, "Configuration Steps for Controlling Your Data Set"

9.2.1 About Domain Values and CSV Worksheet Files for Oracle HR Analytics

You configure Oracle HR Analytics by mapping domain values to columns in the CSV files located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles\ directory.

For more information on domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

> **Note:** Incorrect mappings result in inaccurate calculations of Oracle Business Intelligence metrics.

Table 9–1 lists the CSV worksheet files and the domain values for Oracle HR Analytics in the `$ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles\` directory.

Table 9–1 Domain Values and CSV Worksheet Files for Oracle HR Analytics

Worksheet File Name	Domain Value Table - Column	Description	Package/Interface
domainValues_Employee_Ethnic_Group_Code_ora<ver>.csv	W_EMPLOYEE_D.W_ETHNIC_GRP_CODE	Lists the Ethnic codes and their corresponding domain values of 'Ethnic Group Code' for Oracle EBS.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employee_Sex_MF_ora<ver>.csv	W_EMPLOYEE_D.W_SEX_MF_CODE	Lists the Sex codes and their corresponding domain values of 'Sex Code' for Oracle EBS.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employee_Veteran_Status_Code_ora<ver>.csv	W_EMPLOYEE_D.W_VETERAN_STATUS_CODE	Lists the Veteran codes and their corresponding domain values of 'Veteran Status Code' for Oracle EBS.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employment_Cat_ora<ver>.csv	W_EMPLOYMENT_D.W_EMPLOYEE_CAT_CODE	Lists the User Person Types and their corresponding domain values of 'Employment Category Code' for Oracle EBS.	SDE_ORA_Employment Dimension, SDE_ORA_Employment Dimension_Full
domainValues_Employment_Exempt_Flg_ora<ver>.csv	W_EMPLOYMENT_D.W_EXEMPT_FLG	Lists the FLSA Statuses and their corresponding domain values of 'Exempt Flag' for Oracle EBS.	SDE_ORA_Employment Dimension, SDE_ORA_Employment Dimension_Full
domainValues_Employment_Full_Time_Flg_ora<ver>.csv	W_EMPLOYMENT_D.W_FULL_TIME_FLG	Lists the Employment Categories and their corresponding domain values of 'Full Time Flag' for Oracle EBS.	SDE_ORA_Employment Dimension, SDE_ORA_Employment Dimension_Full

Table 9–1 (Cont.) Domain Values and CSV Worksheet Files for Oracle HR Analytics

Worksheet File Name	Domain Value Table - Column	Description	Package/Interface
domainValues_Employment_Status_ora<ver>.csv	W_EMPLOYMENT_D.W_EMPLOYMENT_STAT_CODE	Lists the Per System Statuses and their corresponding domain values of 'Employment Status' for Oracle EBS.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full
domainValues_EventTypes_ora<ver>.csv	W_EVENT_TYPE_D.W_EVENT_CLASS, W_EVENT_TYPE_D.W_EVENT_GRP_CODE, W_EVENT_TYPE_D.W_EVENT_REASON_CODE, W_EVENT_SUBG_CODE	Lists the Event Types, Event Codes and Meanings and their corresponding domain values of 'Event Group', 'Event Sub-Group' and 'Event Reason' for Oracle EBS.	SDE_ORA_EventTypeDimension_AbsenceAttendance, SDE_ORA_EventTypeDimension_AbsenceAttendance_Full, SDE_ORA_EventTypeDimension_AdditionalEvents_FromFile, SDE_ORA_EventTypeDimension_OtherHREvents, SDE_ORA_EventTypeDimension_OtherHREvents_Full
domainValues_HRPosition_Active_Pos_Flg_ora<ver>.csv	W_HR_POSITION_D.W_ACTIVE_POSITION_FLG	Lists the Position Statuses and their corresponding domain values of 'Active Position Flag' for Oracle EBS.	SDE_ORA_HRPositionDimension, SDE_ORA_HRPositionDimension
domainValues_HRPosition_Exempt_Flg_ora<ver>.csv	W_HR_POSITION_D.W_EXEMPT_FLG	Lists the FLSA Statuses and their corresponding domain values of 'Exempt Flag' for Oracle EBS.	SDE_ORA_HRPositionDimension, SDE_ORA_HRPositionDimension
domainValues_Job_Eeo_Cat_Code_ora<ver>.csv	W_JOB_D.W_EEO_JOB_CAT_CODE	Lists the EEO Job Categories and their corresponding domain values of 'EEO Job Category' for Oracle EBS.	SDE_ORA_JobDimension, SDE_ORA_JobDimension_Full
domainValues_Job_Flsa_Stat_Code_ora<ver>.csv	W_JOB_D.W_FLSA_STAT_CODE	Lists the FLSA Statuses and their corresponding domain values of 'FLSA Status Code' for Oracle EBS.	SDE_ORA_JobDimension, SDE_ORA_JobDimension_Full
domainValues_Pay_Type_Grp_Code_ora<ver>.csv	W_PAY_TYPE_D.W_PAY_TYPE_GRP_CODE	Lists the Classification Names, Element Names and their corresponding domain values of 'Pay Type Group Code' for Oracle EBS.	SDE_ORA_PayTypeDimension, SDE_ORA_PayTypeDimension_Full

Table 9–1 (Cont.) Domain Values and CSV Worksheet Files for Oracle HR Analytics

Worksheet File Name	Domain Value Table - Column	Description	Package/Interface
domainValues_Pay_Type_Flg_ora<ver>.csv	W_PAY_TYPE_D.W_PAY_TYPE_FLG	Lists the Costing Debit or Credit values and their corresponding domain values of 'Pay type Flag' for Oracle EBS.	SDE_ORA_PayTypeDimension, SDE_ORA_PayTypeDimension_Full
domainValues_Employee_Sex_MF_ora<ver>.csv	W_EMPLOYEE_D.W_SEX_MF_CODE	Lists the Sex codes and their corresponding domain values of 'Sex Code' for Oracle EBS.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employee_Veteran_Status_Code_ora<ver>.csv	W_EMPLOYEE_D.W_VETERAN_STATUS_CODE	Lists the Veteran codes and their corresponding domain values of 'Veteran Status Code' for Oracle EBS.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employment_Cat_ora<ver>.csv	W_EMPLOYMENT_D.W_EMPLOYEE_CAT_CODE	Lists the User Person Types and their corresponding domain values of 'Employment Category Code' for Oracle EBS.	SDE_ORA_Employment Dimension, SDE_ORA_Employment Dimension_Full
domainValues_Employment_Exempt_Flg_ora<ver>.csv	W_EMPLOYMENT_D.W_EXEMPT_FLG	Lists the FLSA Statuses and their corresponding domain values of 'Exempt Flag' for Oracle EBS.	SDE_ORA_Employment Dimension, SDE_ORA_Employment Dimension_Full
domainValues_Employment_Full_Time_Flg_ora<ver>.csv	W_EMPLOYMENT_D.W_FULL_TIME_FLG	Lists the Employment Categories and their corresponding domain values of 'Full Time Flag' for Oracle EBS.	SDE_ORA_Employment Dimension, SDE_ORA_Employment Dimension_Full
domainValues_Employment_Status_ora<ver>.csv	W_EMPLOYMENT_D.W_EMPLOYMENT_STAT_CODE	Lists the Per System Statuses and their corresponding domain values of 'Employment Status' for Oracle EBS.	SDE_ORA_Employment Dimension, SDE_ORA_Employment Dimension_Full

Table 9–1 (Cont.) Domain Values and CSV Worksheet Files for Oracle HR Analytics

Worksheet File Name	Domain Value Table - Column	Description	Package/Interface
domainValues_EventTypes_ora<ver>.csv	W_EVENT_TYPE_D.W_EVENT_CLASS, W_EVENT_TYPE_D.W_EVENT_GRP_CODE, W_EVENT_TYPE_D.W_EVENT_REASON_CODE, W_EVENT_SUBG_CODE	Lists the Event Types, Event Codes and Meanings and their corresponding domain values of 'Event Group', 'Event Sub-Group' and 'Event Reason' for Oracle EBS.	SDE_ORA_EventTypeDimension_AbsenceAttendance, SDE_ORA_EventTypeDimension_AbsenceAttendance_Full, SDE_ORA_EventTypeDimension_AdditionalEvents_FromFile, SDE_ORA_EventTypeDimension_OtherHREvents, SDE_ORA_EventTypeDimension_OtherHREvents_Full
domainValues_HRPosition_Active_Pos_Flg_ora<ver>.csv	W_HR_POSITION_D.W_ACTIVE_POSITION_FLG	Lists the Position Statuses and their corresponding domain values of 'Active Position Flag' for Oracle EBS.	SDE_ORA_HRPositionDimension, SDE_ORA_HRPositionDimension
domainValues_HRPosition_Exempt_Flg_ora<ver>.csv	W_HR_POSITION_D.W_EXEMPT_FLG	Lists the FLSA Statuses and their corresponding domain values of 'Exempt Flag' for Oracle EBS.	SDE_ORA_HRPositionDimension, SDE_ORA_HRPositionDimension
domainValues_Job_Eeo_Cat_Code_ora<ver>.csv	W_JOB_D.W_EEO_JOB_CAT_CODE	Lists the EEO Job Categories and their corresponding domain values of 'EEO Job Category' for Oracle EBS.	SDE_ORA_JobDimension, SDE_ORA_JobDimension_Full
domainValues_Job_Flsa_Stat_Code_ora<ver>.csv	W_JOB_D.W_FLSA_STAT_CODE	Lists the FLSA Statuses and their corresponding domain values of 'FLSA Status Code' for Oracle EBS.	SDE_ORA_JobDimension, SDE_ORA_JobDimension_Full
domainValues_Pay_Type_Grp_Code_ora<ver>.csv	W_PAY_TYPE_D.W_PAY_TYPE_GRP_CODE	Lists the Classification Names, Element Names and their corresponding domain values of 'Pay Type Group Code' for Oracle EBS.	SDE_ORA_PayTypeDimension, SDE_ORA_PayTypeDimension_Full
domainValues_Pay_Type_Flg_ora<ver>.csv	W_PAY_TYPE_D.W_PAY_TYPE_FLG	Lists the Costing Debit or Credit values and their corresponding domain values of 'Pay type Flag' for Oracle EBS.	SDE_ORA_PayTypeDimension, SDE_ORA_PayTypeDimension_Full

---

Note: When editing CSV files, make sure that you:

- Do no change the case of values in the CSV file.

  For example, do not change 'CONTRACTOR' to 'Contractor'.

- Do not add new values to the W_ columns, which are not already included in the CSV file.

  In other words, you can add new rows to the spreadsheet, but the W_ values must map to those in the out-of-the-box spreadsheet.

9.2.2 How to Configure the Employee Ethnic Group Codes

This section explains how to configure the domainValues_Employee_Ethnic_Group_Code_ora<ver>.csv file.

1. Identify the Ethnic Group Codes in your Oracle source system by using the following SQL (executed using APPS schema credentials):

   ```
   SELECT DISTINCT PER_INFORMATION1 FROM PER_ALL_PEOPLE_F
   WHERE PER_INFORMATION1 in
   ('1','2','3','4','5','6','7','8','9','10','11','12','BA','BC','BO','C','I','O',
   'P','W')
   ORDER BY 1
   ```

2. Open the domainValues_Employee_Ethnic_Group_Code_ora<ver>.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles\ directory.

3. Copy the PER_INFORMATION1 to the ETHNIC_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each Ethnic Code to one domain value.
5. Save and close the file.

9.2.3 How to Configure the Employee Sex Codes

This section explains how to configure Employee Sex Codes using the file domainValues_Employee_Sex_MF_ora<ver>.csv.

1. Identify the Sex Codes in your Oracle source system by using the following SQL:

   ```
   SELECT DISTINCT SEX FROM PER_ALL_PEOPLE_F ORDER BY 1
   ```

2. Open the domainValues_Employee_Sex_MF_ora<ver>.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles\ directory.

3. Copy the SEX column to the SEX column in the file. The data must be copied starting from the 6th line.
4. Map each Sex Code to one domain value.
5. Save and close the file.

9.2.4 How to Configure Employee Veteran Status Codes

This section explains how to configure Employee Veteran Status codes using the file domainValues_Employee_Veteran_Status_Code_ora<ver>.csv

1. Identify the Veteran Status Codes in your Oracle source system by using the following SQL:

   ```
   SELECT DISTINCT PER_INFORMATION5 FROM PER_ALL_PEOPLE_F
   WHERE PER_INFORMATION5 in ('NOTVET', 'OTEDV', 'VET', 'VETDIS', 'VIETVET',
   'VIETVETDIS')
   ORDER BY 1
   ```

2. Open the domainValues_Employee_Veteran_Status_Code_ora<ver>.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles\ directory.

3. Copy the PER_INFORMATION5 column to the VETERAN_STATUS_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each Veteran Status Code to one domain value.
5. Save and close the file.

9.2.5 How to Configure the Employment Category Codes

This section explains how to configure Employment Category Codes using the file domainValues_Employment_Cat_ora<ver>.csv.

1. Identify the User Person Types in your Oracle source system by using the following SQL:

   ```
   SELECT DISTINCT SYSTEM_PERSON_TYPE, USER_PERSON_TYPE FROM
   PER_PERSON_TYPES
   WHERE SYSTEM_PERSON_TYPE IN
   ('EMP','OTHER','EMP_APL','EX_EMP','EX_EMP_APL','RETIREE','PRTN')
   ORDER BY 1,2
   ```

2. Open the domainValues_Employment_Cat_ora<ver>.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles\ directory.

3. Copy the User Person Types to the USER_PERSON_TYPE column in the file. The data must be copied starting from the 6th line.
4. Map each User Person Type (results of the SQL query) to one of the delivered W_EMPLOYEE_CATEGORY_CODE domain values in the delivered example csv file.

   You can map more than one User Person Type to the same W_EMPLOYEE_CATEGORY_CODE by adding in new rows. For example:

   ```
   Contractor          CONTRACTOR  CONTRACTOR
   Contingent Worker   CONTRACTOR  CONTRACTOR
   ```

   System Person Types are also extracted with User Person Type to help you map the domain values. Do not copy the System Person types in the CSV file.

5. Save and close the file.

9.2.6 How to Configure Employment Exempt Status

This section explains how to configure Employment Exempt Status using the file domainValues_Employment_Exempt_Flg_ora<ver>.csv

1. Identify the FLSA Statuses in your Oracle source system by using the following SQL:

   ```
   SELECT DISTINCT JOB_INFORMATION3 FROM PER_JOBS  ORDER BY 1
   ```

2. Open the domainValues_Employment_Exempt_Flg_ora<ver>.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles\ directory.

3. Copy the JOB_INFORMATION3 to the FLSA_STATUS_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each FLSA_STATUS_CODE to one domain value.
5. Save and close the file.

9.2.7 How to Configure Employment Full Time Status

This section explains how to configure Employment Full Time Status using the file domainValues_Employment_Full_Time_Flg_ora<ver>.csv.

1. Identify the Employment Categories in your Oracle source system by using the following SQL:

   ```
   SELECT DISTINCT EMPLOYMENT_CATEGORY FROM PER_ALL_ASSIGNMENTS_F ORDER BY 1
   ```

2. Open the domainValues_Employment_Full_Time_Flg_ora<ver>.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles\ directory.

3. Copy the Employment Category to the EMPLOYMENT_CATEGORY column in the file. The data must be copied starting from the 6th line.
4. Map each EMPLOYMENT_CATEGORY to one domain value.
5. Save and close the file.

9.2.8 How to Configure Employment Status

This section explains how to configure Employment Status using the file domainValues_Employment_Status_ora<ver>.csv

1. Identify the Per System Statuses in your Oracle source system by using the following SQL:

   ```
   SELECT DISTINCT PER_SYSTEM_STATUS FROM PER_ASSIGNMENT_STATUS_TYPES
   ORDER BY 1
   ```

2. Open the domainValues_Employment_Status_ora<ver>.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles\ directory.

3. Copy the PER_SYSTEM_STATUS to the PER_SYSTEM_STATUS column in the file. The data must be copied starting from the 6th line.

4. Map each PER_SYSTEM_STATUS to one domain value.
5. Save and close the file.

9.2.9 How to Configure Event Types

This section explains how to configure Event Types using the file domainValues_EventTypes_ora<ver>.csv

1. Identify the Event Types in your Oracle source system by using the following SQL:

   ```
   SELECT DISTINCT LOOKUP_TYPE, LOOKUP_CODE, MEANING
   FROM FND_LOOKUP_VALUES
   WHERE LOOKUP_TYPE IN
   ('EMP_ASSIGN_REASON',
   'LEAV_REAS',
   'PROPOSAL_REASON')
   ORDER BY 1, 2, 3
   ```

2. Open the domainValues_EventTypes_ora<ver>.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles\ directory.

3. Copy the Lookup Type, Lookup Code and Meaning to the LOOKUP_TYPE, LOOKUP_CODE, and MEANING columns in the file respectively. The data must be copied starting from the 6th line. Use commas to separate the entries.
4. Map each Event Type (LOOKUP_CODE) to one domain value for each of the 3 domain columns — W_EVENT_GRP_CODE, W_EVENT_SUBG_CODE, and W_EVENT_REASON_CODE. Event Category (LOOKUP_TYPE) and Event Description (MEANING) are also extracted with Event Type to help you map the domain values.
5. Save and close the file.

9.2.10 How to Configure HR Active Position Status

This section explains how to configure HR Active Position Status using the file domainValues_HRPosition_Active_Pos_Flg_ora<ver>.csv.

1. Identify the Position Statuses in your Oracle source system by using the following SQL:

   ```
   SELECT DISTINCT STATUS FROM HR_ALL_POSITIONS_F ORDER BY 1
   ```

2. Open the domainValues_HRPosition_Active_Pos_Flg_ora<ver>.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles\ directory.

3. Copy the STATUS to the STATUS column in the file. The data must be copied starting from the 6th line.
4. Map each position STATUS to one domain value.
5. Save and close the file.

9.2.11 How to Configure HR Position Exempt Status

This section explains how to configure HR Position Exempt Status using the file domainValues_HRPosition_Exempt_Flg_ora<ver>.csv.

1. Identify the FLSA Statuses in your Oracle source system by using the following SQL:

   ```
   SELECT DISTINCT JOB_INFORMATION3 FROM PER_JOBS
   ORDER BY 1
   ```

2. Open the domainValues_HRPosition_Exempt_Flg_ora<ver>.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles\ directory.

3. Copy the JOB_INFORMATION3 to the FLSA_STATUS_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each FLSA_STATUS_CODE to one domain value.
5. Save and close the file.

9.2.12 How to Configure Job EEO Category Codes

This section explains how to configure Job EEO Category Codes using the file domainValues_Job_Eeo_Cat_Code_ora<ver>.csv

1. Identify the EEO Job Categories in your Oracle source system by using the following SQL:

   ```
   SELECT DISTINCT JOB_INFORMATION1 FROM PER_JOBS
   ORDER BY 1
   ```

2. Open the domainValues_Job_Eeo_Cat_Code_ora<ver>.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles\ directory.

3. Copy the JOB_INFORMATION1 to the EEO_JOB_CAT_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each EEO_JOB_CAT_CODE to one domain value.
5. Save and close the file.

9.2.13 How to Configure Job FLSA Status Codes

This section explains how to configure Job FLSA Status Codes using the file domainValues_Job_Flsa_Stat_Code_ora<ver>.csv.

1. Identify the FLSA Statuses in your Oracle source system by using the following SQL:

   ```
   SELECT DISTINCT JOB_INFORMATION3 FROM PER_JOBS
   ORDER BY 1
   ```

2. Open the domainValues_Job_Flsa_Stat_Code_ora<ver>.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles\ directory.

3. Copy the JOB_INFORMATION3 to the FLSA_STAT_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each FLSA_STAT_CODE to one domain value.
5. Save and close the file.

9.2.14 How to Configure Pay Type Group Codes

This section explains how to configure Pay Type Group Codes using the file domainValues_Pay_Type_Grp_Code_ora<ver>.csv.

1. Identify the Pay Elements in your Oracle source system by using the following SQL:

```
SELECT DISTINCT CLASSIFICATION_NAME, ELEMENT_NAME
FROM
PAY_ELEMENT_TYPES_F,
PAY_ELEMENT_CLASSIFICATIONS
WHERE
PAY_ELEMENT_CLASSIFICATIONS.CLASSIFICATION_ID = PAY_ELEMENT_TYPES_
F.CLASSIFICATION_ID AND
CLASSIFICATION_NAME NOT LIKE '%Information%' AND
CLASSIFICATION_NAME NOT LIKE '%Employer%' AND
CLASSIFICATION_NAME NOT LIKE '%Balance%'
ORDER BY 1, 2
```

2. Open the domainValues_Pay_Type_Grp_Code_ora<ver>.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles\ directory.

3. Copy the Classification Name and Element Name to the CLASSIFICATION_NAME and ELEMENT_NAME columns in the file respectively. The data must be copied starting from the 6th line. Use commas to separate the entries.

   Note: Do not change lines one to four in the domainValues_Pay_Type_Grp_Code_ora<ver>.csv file.

4. Map each Element Name to one domain value. Classification Names are also extracted with Element Names to help you map the domain values. If the element is not related to Payroll Pay Check, you can map the element to 'OTHER'.
5. Save and close the file.

9.2.15 How to Configure the Pay Type Flag

This section explains how to configure Pay Type Flags using the file domainValues_Pay_Type_Flg_ora<ver>.csv.

1. Identify the Costing (Debit or Credit) in your Oracle source system by using the following SQL:

```
SELECT DISTINCT COSTING_DEBIT_OR_CREDIT FROM PAY_ELEMENT_CLASSIFICATIONS
ORDER BY 1
```

2. Open the domainValues_Pay_Type_Flg_ora<ver>.csv file in a text editor.

   This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles\ directory.

3. Copy the COSTING_DEBIT_OR_CREDIT to the COSTING_DEBIT_OR_CREDIT column in the file. The data must be copied starting from the 6th line.
4. Map each _DEBIT_OR_CREDIT to one domain value.
5. Save and close the file.

9.2.16 How to Configure Address Types for HR Profile

This section explains how to configure Address Types for HR Profile. There are three address fields in the Employee dimension table:

- Permanent address
- Mail address
- Work address

For each of these, we use only the primary ones. In addition, the following logic is used to determine the various types of addresses:

- Permanent: Address Type = 'H' (This is also the home address, in other words)
- Mail: Address Type = 'M'. If this is not available, use Permanent address (which can be the best alternate for mailing address).
- Work: Address Type = 'Default'. If this is not available, use Permanent address (which can be the best alternate for mailing address).

You can modify this logic if required. For example, if your system tracks work address with a special address type 'W', then you should be able to modify the existing logic. Or, if you do not want to assume that the mail address (M) is same as the home address (H), you may want to remove the null evaluation check there.

To configure Address Type:

1. In ODI Designer, display the Projects view, and expand the 'Oracle BI Applications 7.9.5.2' folder.
2. Expand the Interfaces node.
3. Double-click the Interface SDE_ORA_EmployeeDimension_Addresses.EMPLOYEE_D_ADDRESSES_TML to display the Interface: <*Name*> dialog.
4. Display the Diagram tab.
5. In the Target Datastore area, select the column ADDRESS_TYPE.
6. In the Mapping pane below, modify the value in the **Implementation** box.

   For example, if you have a specific address type for work addresses, (for example 'W'), you would change the expression for the ADDRESS_TYPE column, from:

   ```
   COALESCE(PER_ADDRESSES.ADDRESS_TYPE,'Default')
   ```

   To:

   ```
   COALESCE(PER_ADDRESSES.ADDRESS_TYPE,'W')
   ```

7. Repeat steps 3 to 6 for the Interface SDE_ORA_EmployeeDimension_Addresses.EMPLOYEE_D_ADDRESSES_TML_FULL.
8. Save the changes.
9. Double-click the Interlace Open Interface SDE_ORA_Employee_Dimension.EMPLOYEE_DS to display the Interface: <*Name*> dialog.

10. Display the Diagram tab.
11. Locate the source tables SQ_Employees and LKP_ADDRESSES_WORK.
12. Select the join between these source tables to display the 'Join between <*Name*>' pane.
13. In the **Implementation** box, update the WHERE clause predicate as follows, from:

    ```
    LKP_ADDRESSES_WORK.ADDRESS_TYPE='Default'
    ```

    To:

    ```
    LKP_ADDRESSES_WORK.ADDRESS_TYPE='W'
    ```

14. Save the changes.

If you do not want to assume that the mail address (M) is the same as the home address (H) in the event of mail address not being available, then you would modify the logic in the Target Datastore for the mappings MAIL_ADDR_EFF_DATE, MAIL_CITY, MAIL_COUNTRY_CODE, MAIL_COUNTRY_NAME, MAIL_COUNTRY_REGION, MAIL_STATE_CODE, MAIL_STATE_NAME, MAIL_STATE_REGION, MAIL_ST_ADDRESS, MAIL_ZIPCODE.

For example, for MAIL_CITY, you might change the value from:

```
IIF(LKP_ADDRESSES_MAIL.ADDRESS_ID IS NOT NULL,
LKP_ADDRESSES_MAIL.TOWN_OR_CITY,
LKP_ADDRESSES_DEFAULT.TOWN_OR_CITY
)
```

To:

```
LKP_ADDRESSES_MAIL.TOWN_OR_CITY
```

9.2.17 How to Configure Phone Types for HR Profile

This section explains how to configure Phone Types for HR Profile. There are four phone related fields in the Employee dimension table, as follows:

- Fax
- Work phone
- Pager
- Mobile phone

The following logic is used to arrive at the various types of addresses:

- Fax: Phone Type = 'WF' (Work Fax)
- Work Phone: Phone Type = 'W1' (First work phone, if there are more than one)
- Pager: Phone Type = 'P' (Pager)
- Mobile: Phone Type = 'M' (Mobile)

You can modify this logic if required. For example, if your system tracks the primary work phone with a special phone type 'WP', instead of W1, then you should be able to modify the existing logic. The same applies for other phone types as well.

To configure Phone Type:

1. In ODI Designer, display the Projects view, and expand the 'Oracle BI Applications 7.9.5.2' folder.

2. Expand the Interfaces node.
3. Double-click the Interface SDE_ORA_Employee_Dimension.EMPLOYEE_DS to display the Interface: <*Name*> dialog.
4. Display the Diagram tab.
5. Locate the source tables Sq_employees and LKP_PHONES_WORK.
6. Select the join between these source tables to display the 'Join between <*Name*>' pane, and use the **Implementation** box to modify the behavior.

   For example, if you have a specific phone type for the primary work phone, 'WP', you might modify the predicate in the expression as follows, from:

   ```
   LKP_PHONES_WORK.PHONE_TYPE='W1'
   ```

   To:

   ```
   LKP_PHONES_WORK.PHONE_TYPE='WP'
   ```

7. Save the changes.

9.2.18 How to Configure Education Degree Codes for Employee Dimension

This section explains how to configure the categories that are used to evaluate the highest education degree code for an employee.

The package SDE_ORA_ EmployeeDimension uses the ODI parameter QUALIFICATION_CATEGORY_LIST to get the list of categories for the defined Education Degrees in Oracle E-Business Suite. The package uses this parameter to identify and populate the Highest Education Degree attribute for the Employee Dimension. Incorrect setup of this parameter might cause data quality issues with the Highest Education Degree attribute in the Employee Dimension.

To configure the categories that are used to evaluate the highest education degree code for an employee:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Common tab.
4. Specify a value for the QUALIFICATION_CATEGORY_LIST parameter.

   For example, QUALIFICATION_CATEGORY_LIST = 'DEGREE', 'DT'.
5. Save your changes.

To obtain the list of the valid values for the education degree codes please login to Oracle E-Business Suite Instance using SQL*Plus and execute the following SQL (executed using APPS schema credentials):

```
SELECT lookup_code, meaning FROM hr_lookups WHERE lookup_type =
'PER_CATEGORIES'
```

Select from the returned lookup_code column values and decide which ones are used to identify Education Degrees.

9.2.19 About Configuring Key Flexfields

This section explains how to configure Key Flexfields in Oracle EBS applications.

In Oracle EBS Application module, a Flexfield is a field that allows a user to customize the Oracle applications. Each Flexfield in Oracle in turn consists of sub-fields called segments. Each segment is assigned a value set. The value set consists of values.

The main purpose of using Flexfields in Oracle EBS Applications is to customize the applications according to the business environment and practices. Oracle EBS Application allows a user to validate values that are entered in these flexfields. A user can change the structure of a Flexfield depending on the data in the application.

Since Flexfield configurations vary from implementation to implementation, in order to accommodate all kinds of flexfield configurations done at the Oracle EBS Applications end, the attributes sourced from Flexfield columns is parameterized.

To configure Oracle HR Analytics to support flexfields, do the following:

- Section 9.2.20, "How to Configure the Key Flexfields for the Job Dimension"
- Section 9.2.21, "How to Configure the Key Flexfields for the HR Position Dimension"
- Section 9.2.22, "How to Configure the Key Flexfields for the Pay Grade Dimension"
- Section 9.2.23, "How to Configure multi-segmented Flexfields"

9.2.20 How to Configure the Key Flexfields for the Job Dimension

This section explains how to configure the Key Flexfields Job Code, Job Name and Job Family Code.

To configure the Key Flexfields for the Job Dimension

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Application Specific tab, and select Human Resources Analytics from the **Select BI Application** field
4. Specify a value for the following parameters:
   - JOBCODE_FLXFLD_SEGMENT_COL = the Column Name of Job Code.
   - JOBFAMILYCODE_FLXFLD_SEGMENT_COL = the Column name of JobGroup Code.
5. Display the Common tab.
6. Specify a value for the following parameters:
   - JOBCODE_FLXFLD_SEGMENT_COL = the Column Name of Job Code.
   - JOBNAME_FLXFLD_SEGMENT_COL = the Column Name of the JobName.

To find out what columns are defined in the Flexfield, do the following:

1. Login with Application Developer Responsibility.
2. Under the Key Flexfield, click on Segments.
3. Query for the Application Human Resources and locate the Flexfield Title relevant to Job.
4. Click on the Segments corresponding to the Flexfield that is being used in order to find the columns being used for the Job Code and Job Name.

9.2.21 How to Configure the Key Flexfields for the HR Position Dimension

This section explains how to configure the Position Number column.

To configure the Key Flexfields for the HR Position Dimension

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Parameters** link.
3. Display the Common tab.
4. Specify a value for the following parameter:
   - POSITION_NUM_FLXFLD_SEGMENT_COL = the Column Name of Position Num
5. Save your changes.

To find out what columns are defined in the Flexfield, do the following:

1. Login with Application Developer Responsibility.
2. Under the Key Flexfield, click on Segments.
3. Query for the Application Human Resources and locate the Flexfield Title relevant to Position.
4. Click on the Segments corresponding to the Flexfield that is being used in order to find the columns being used for the Position Num.

9.2.22 How to Configure the Key Flexfields for the Pay Grade Dimension

This section explains how to configure the Pay Level Name column.

To configure the Key Flexfields for the Pay Grade Dimension

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Common tab.
4. Specify a value for the following parameter:
   - PAYLVLNAME_FLXFLD_SEGMENT_COL = the Column Name for the Paylevel Name

To find out what columns are defined in the Flexfield, do the following:

1. Login with Application Developer Responsibility.
2. Under the Key Flexfield, click on Segments.
3. Query for the Application Human Resources and locate the Flexfield Title relevant to Job.
4. Click on the Segments corresponding to the Flexfield that is being used in order to find the columns being used for the Job Code and Job Name.

9.2.23 How to Configure multi-segmented Flexfields

This section explains how to configure HR Analytics for multi-segmented Flexfields for any column.

In case your Flexfield configuration for a given business column involves more that one segment column, or in case you want to report a business column as a concatenation of more than one segment column separated by a character (for example, a period '.'), then the corresponding parameter can be used to provide the information correctly. The example of configuring Job Name flexfield is provided below to illustrate the process.

When Oracle Business Intelligence Applications is installed out-of-the-box, the ODI parameter for the package 'SDE_ORA_CodeDimension_Job' is set to the following:

```
JOBNAME_FLXFLD_SEGMENT_COL=SEGMENT6
```

If the Job Name in your deployment is configured using SEGMENT3 together with SEGMENT6 (or if you want to report Job Name as a concatenation of these two with 'dot' as the separator), you should set the ODI parameter value as:

```
JOBNAME_FLXFLD_SEGMENT_COL=SEGMENT3 || SEGMENT6
```

Alternatively, if you want a period (that is '.') in between, set the ODI parameter value as:

```
JOBNAME_FLXFLD_SEGMENT_COL=SEGMENT3 || '.' || SEGMENT6
```

If your OLTP table already stores the concatenated value in some other column (like NAME), you could use it directly as follows:

```
JOBNAME_FLXFLD_SEGMENT_COL=NAME
```

Note: You must make sure you take a look at the SQL code (in ODI Designer) before setting the parameter values. For example, before configuring the values for the ODI parameter JOBNAME_FLXFLD_SEGMENT_COL (as in the above example), you should know which table the data will be extracted from. If necessary, use Oracle BI Applications Configuration Manager to lookup the package name (for example, SDE_ORA_CODEDIMENSION_JOB), then use ODI Designer to obtain the OLTP table name for this package (using the Projects\Mappings\SDE_ORA11510_Adaptor folder).

Once you know the OLTP table names, you can work out the column names to use in the parameter file (SEGMENT3 or SEGMENT6 or NAME and so on) that really exist in the table PER_JOB_DEFINITIONS.

9.2.24 How to Configure Flags for the Pay Type Dimension

This section explains how to configure Pay Types. The Pay Type dimension W_PAY_TYPE_D has three Flag columns namely COMP_FLG, TAXABLE_FLG and PENSION_COMP_FLG, apart from others. Each of these three flags can be either 'Y' or 'N'. These flag columns indicate whether the Pay Type specified in the current record is a Taxable Earning type, or Regular Compensation Earning type or a Pension Compensation Earning type. These Flag columns are chosen to contribute towards slowly changing dimension of type 2, and hence should be configured correctly.

Each of these three Flag columns are evaluated based on the Incoming Classification Name using a Flat File lookup. This Flat file needs to be configured to see the accurate values of the Pay Type Flags in the Data Warehouse.

To Configure Flags for the Pay Type Dimension

1. Open the file_comp_pension_taxable_flg_ora<ver>.csv in a text editor.

This file is located in the $ODI_HOME\biapps_odi\odifiles\odidatafiles\lkpfiles\ directory.

2. Add all possible Classification Names that are being used in your business.
3. Corresponding to the Classification Name, add the relevant values for the Pay Type Flags. Make sure to use either 'Y' or 'N'.

In Oracle HR Analytics, these flags apply to Earnings only. Therefore, you should only need to configure Classification Names related to Earnings. For other areas, the default ODI mapping values should be acceptable.

9.2.25 How to Configure Classification Names for Payroll

This section explains how to configure classification names for Payroll.

The Aggregated items are loaded into the Payroll Fact table with a DETAIL_FLG = 'N', apart from the line items. The Aggregated Line items that are loaded are: TOTAL_GROSS, NET_PAY, TOTAL_DEDUCTIONS and TOTAL_TAXES.

The following parameters need to be configured to get accurate results. Each Aggregated Line item is computed by grouping by elements that belong to a certain Classification name set.

Table 9–2 Classification Name Parameters

Parameter Name	Description
GROSS_PAY_CLASSIFICATION_NAME	Add all the Classification Names that add up to the Total Gross Pay.
TOTAL_DEDUCTION_CLASSIFICATION_NAME	Add all the Classification Names that correspond to any deduction.
TOTAL_TAXES_CLASSIFICATION_NAME	Add all the Classification Names that correspond to various Taxes.
TOTAL_GROSS_FOR_NETPAY_EARNINGS	Add all the Earnings Classification Names that are realized in Net pay. Note that some Classification Names are considered for Gross pay, but do not feature in the Net Pay (for example, Imputed Income). This is the major difference between this parameter and the GROSS_PAY_CLASSIFICATION_NAME parameter.
TOTAL_GROSS_FOR_NETPAY_DEDUCTIONS	This is same as the parameter TOTAL_DEDUCTION_CLASSIFICATION_NAME. However, in case some additional Classification Names are required to be added or removed for Net Pay, this parameter should be used for that.

The parameters are assigned default values when Oracle Business Intelligence Applications is installed out of the box, but you can modify the values by following the steps below.

To configure the classification name parameters:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.

3. Display the Application Specific tab, and select Human Resources Analytics from the **Select BI Application** field
4. Specify an appropriate value for the following parameters:
   - CLASSIFICATION_NAMES_FOR_GROSS_PAY='PTOAccruals','Earnings','Imputed Earnings'
   - CLASSIFICATION_NAMES_FOR_TOTAL_DEDUCTION='InvoluntaryDeductions','Pre-Tax Deductions','Voluntary Deductions','Tax Deductions'
   - CLASSIFICATION_NAMES_FOR_TOTAL_TAXES='Tax Deductions'
   - CLASSIFICATION_NAMES_FOR_NET_PAY_EARNINGS='PTOAccruals','Earnings'
   - CLASSIFICATION_NAMES_FOR_NET_PAY_DEDUCTIONS='InvoluntaryDeductions','Pre-Tax Deductions','Voluntary Deductions','Tax Deductions'
5. Save your changes.

To find out the possible Classification Names existing in your system, please run the following SQL against the OLTP Database:

```
SELECT CLASSIFICATION_NAME FROM PAY_ELEMENT_CLASSIFICATIONS
WHERE LEGISLATION_CODE LIKE 'US'
```

The Classification Name of Elements can be obtained using the following SQL:

Note: Enter the Element Type IDs of the elements for which you need the Classification Name within the quotes. If there are multiple elements, separate them using commas:

```
SELECT E.ELEMENT_TYPE_ID,E.ELEMENT_NAME,C.CLASSIFICATION_NAME
FROM PAY_ELEMENT_TYPES_F E, PAY_ELEMENT_CLASSIFICATIONS C

WHERE E.CLASSIFICATION_ID = C.CLASSIFICATION_ID AND E.ELEMENT_
TYPE_ID IN ()
```

9.2.26 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle HR Analytics.

- Section 9.2.26.1, "How to Aggregate the Payroll Table for Oracle HR Analytics"
- Section 9.2.26.3, "How to control the rolling period for storing Employee Daily Snapshot data"
- Section 9.2.26.4, "How to control the granularity of the Employee Monthly Snapshot data"

9.2.26.1 How to Aggregate the Payroll Table for Oracle HR Analytics

You can aggregate the Payroll table to a different time levels, and aggregate levels of Employees, Jobs, and Payment Types dimensions. There are two time grain parameters to configure for this aggregate table and these parameters need to have the same value.

The GRAIN parameter has a preconfigured value of Month. The possible values for the GRAIN parameter are:

- DAY

- WEEK
- MONTH
- QUARTER
- YEAR

The Payroll aggregate table is fully loaded from the base table in the initial E-LT run by the mapping 'PLP_PayrollAggregate_Load_Full'. The table can grow to millions of records. The Payroll aggregate table is not fully reloaded from the base table after an incremental E-LT run. Oracle HR Analytics minimizes the incremental aggregation effort, by modifying the aggregate table incrementally as the base table is updated. Oracle Business Intelligence Applications looks for new records in the base table during the incremental E-LT. This process is done in two steps:

1. There are new records in the W_PAYROLL_A table, which are inserted after the last E-LT run. These new records are inserted into the W_PAYROLL_A_TMP table. This step is part of the post load-processing workflow, and the mapping is called 'PLP_PayrollAggregate_Extract'.
2. Oracle HR Analytics aggregates the W_PAYROLL_A_TMP table and joins it with the W_PAYROLL_A aggregate table to insert new or update existing buckets to the aggregate table. This step is part of the post load-processing workflow, and the mapping is called 'PLP_PayrollAggregate_Load'.

To load the Payroll aggregate table (W_PAYROLL_A), you need to configure the post-load processing parameters in Oracle Business Intelligence Applications Configuration Manager.

To load the Payroll aggregate table (W_PAYROLL_A):

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Application Specific tab, and select Human Resources Analytics from the **Select BI Application** field.
4. Edit the value of GRAIN parameter for the PLP_PayrollAggregate_Load scenario, and specify an appropriate value ('DAY', 'WEEK', 'MONTH', 'QUARTER' or 'YEAR').
5. Save the Task.

9.2.26.2 About Configuring the Employee Snapshot Fact tables

The Employee Daily Snapshot Fact table stores employee-related metrics for all employees against their primary job function. You can configure the two fact tables that store Employee Snapshot data, namely W_EMPLOYEE_DAILY_SNP_F and W_EMPLOYEE_MONTHLY_SNP_F.

You can configure the Rolling Period and the Granularity for the Employee Snapshot data process.

Rolling Period

The Employee Daily Snapshot Fact table is loaded everyday, the date being identified by the snapshot date column. For example, if your organization has 1000 employees, this table would load 1000 records everyday. If an employee’s contract is terminated on a particular day, the following day's snapshot data would not have that record (that is, 999 records would be uploaded). Continuing this example, if you run E-LT for 60 days, you will have 60 snapshot data sets. However, the number of days worth of snapshot data you want to store can be configured. In fact, you can configure to store only for a 'rolling period of' some days, example 30. This is the out of box setting. If you start your E-LT on January 1 and keep running for 40 days, you will have snapshot data sets from Jan 20 until Feb 10. If you configure to store only 7 rolling days, you will have snapshot data sets from Feb 3 until Feb 10, and so on.

The ODI package 'PLP_EmployeeDailySnapshot_Trim' maintains the rolling period choice, and uses two ODI parameters:

- KEEP_PERIOD

  Indicates your choice of the period type based on which you will be providing the rolling periods. The Valid values for this parameter are 'DAY', 'WEEK', 'MONTH', 'QUARTER', and 'YEAR' (values should include single quotes as indicated).

- NUM_OF_PERIOD

  Indicates your choice of how many days/weeks/months/years (set by KEEP_PERIOD) data you want to preserve in a rolling fashion. The valid value for this parameter is any integer.

For example, if you choose to preserve a rolling period of 45 Days, you would set KEEP_PERIOD='DAY' and NUM_OF_PERIOD=45.

Granularity

The Employee Monthly Snapshot Fact table stores the end-of-month snapshot data for all employees in your organization. Out of the box, this table stores end-of-month snapshot data, but this is configurable to store end-of-week, end-of-quarter or even end-of-year data. If you configure it to store end-of-month, as it is installed out-of-the-box, then you will have one snapshot data set for a given month, the set being refreshed everyday.

The out-of-box setting for the W_EMPLOYEE_MONTHLY_SNP_F table is to store end-of-month snapshot data set, on set for each calendar month. However, this can be configured. If you like to have this table store end-of-week, or end-of-quarter or even end-of-year data, you need to configure the GRAIN parameter in Oracle Business Intelligence Applications Configuration Manager. The GRAIN parameter indicates the chosen granularity of the snapshot fact table that stores data at a interval higher than the base snapshot fact table. Valid values for this parameter are 'WEEK', 'MONTH', 'QUARTER', 'YEAR' (values should include single quotes as indicated.

Notes:

- Do not set the value of GRAIN to 'DAY'.
- In the reporting logical metadata (RPD), the table W_EMPLOYEE_MONTHLY_SNP_F is joined with the Calendar Month dimension table - both in the physical layer as well as the logical layer. If you are change the GRAIN out-of-box value from 'MONTH' to anything else (like 'QUARTER'), you would need to modify the join in the RPD as well to the appropriate date dimension.
- The physical name of the table can still remain as W_EMPLOYEE_MONTHLY_SNP_F. You can change the logical name (or alias) appropriately. For example, if your GRAIN is 'QUARTER', you might change it to quarterly snapshot.

To configure the Employee Snapshot Fact tables, perform the following tasks:

- Section 9.2.26.3, "How to control the rolling period for storing Employee Daily Snapshot data"

- Section 9.2.26.4, "How to control the granularity of the Employee Monthly Snapshot data"

9.2.26.3 How to control the rolling period for storing Employee Daily Snapshot data

For more information about the rolling period for the Employee Snapshot data, see Section 9.2.26.2, "About Configuring the Employee Snapshot Fact tables".

To configure the rolling period for storing Employee Daily Snapshot data:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Application Specific tab, and select Human Resources Analytics from the **Select BI Application** field
4. Set the values of the two parameters KEEP_PERIOD and NUM_OF_PERIOD for the PLP_EmployeeDailySnapshot_Trim scenario.

   Note: Although the NUM_OF_PERIOD requires an integer value, do not change the data type from 'text'.

9.2.26.4 How to control the granularity of the Employee Monthly Snapshot data

For more information about granularity in the Employee Snapshot data, see Section 9.2.26.2, "About Configuring the Employee Snapshot Fact tables".

To configure the granularity of the Employee Snapshot data:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the **Administer ELT Parameters** link.
3. Display the Application Specific tab, and select Human Resources Analytics from the **Select BI Application** field.
4. Specify an appropriate value for the GRAIN parameter for the PLP_EmployeeMonthlySnapshot scenario.

10 Configuring the Oracle Business Intelligence Applications Repository

This chapter describes how to configure the Oracle BI Repository for the Oracle Business Analytics Warehouse, and contains the following topics:

- Section 10.1, "Configuring the Oracle BI Repository Connections"
- Section 10.2, "Setting up Date Specific Metrics"
- Section 10.3, "Setting Up Additional Time Series Metrics for Oracle Business Analytics Warehouse"
- Section 10.4, "Setting Up Additional Dimension Tables for Oracle Business Analytics Warehouse"
- Section 10.6, "About Oracle BI Time Repository Variables"
- Section 10.7, "About User Authentication"
- Section 10.8, "About the Security or Visibility Configuration"
- Section 10.9, "About the Group Variable"
- Section 10.5, "About the Period Ago Keys for Oracle Business Analytics Warehouse"
- Section 10.10, "About Configuring Usage Tracking for Oracle Business Analytics Warehouse"
- Section 10.11, "About the Incremental Deployment of the Oracle BI Applications Repository"

Note: When you access the OracleBIAnalyticsApps.rpd repository in Oracle BI Administration Tool, you must log on as the username and password Administrator\SADMIN.

10.1 Configuring the Oracle BI Repository Connections

The Oracle Business Intelligence Applications repository uses two databases defined in the Physical layer, as follows:

- Oracle Data Warehouse
- Oracle EBS OLTP

You need to configure the Oracle BI repository variables and connection pools to connect to your databases, as follows:

- For information about configuring connection pools, see Section 10.1.2, "How to Configure Oracle Business Intelligence Applications Repository Connections".
- For information about configuring repository variables, see Section 10.1.3, "How to Configure Oracle Business Intelligence Applications Repository Variables".

10.1.1 About The Predefined Connection Pools In The Oracle Business Analytics Warehouse

The Oracle Business Analytics Warehouse physical database has two predefined connection pools:

- **Oracle Business Analytics Warehouse Connection Pool.** The Oracle Business Analytics Warehouse Connection Pool is the main connection pool in the Oracle Business Intelligence Applications Repository. You need to configure this connection pool to connect to your physical data warehouse. The connection is used by the session initialization blocks. You can use this connection pool to set up a dynamic data source name.
- **Oracle Business Analytics Warehouse Repository Initblocks Connection Pool.** Yo u need to configure the Oracle Business Analytics Warehouse Repository Initblocks Connection Pool to connect to the your physical data warehouse. The connection is used by the repository level initialization blocks. Repository level initialization blocks cannot be configured to use the dynamic data source name.

  You can also set up dynamic data source names, which allow an Administrator to set one instance of Analytics server to connect to different data warehouses depending on the user. For more information about how to set up dynamic data source names, see Section 10.1.1.1, "How to Configure Dynamic Data Source Names".

You also need to configure the following Static variables:

- **OLAP_DSN.** The value of the OLAP_DSN static variable is set to the data source name for the warehouse database.
- **OLAP_USER.** The value of the OLAP_USER static variable is set to the database user name for the warehouse database.
- **OLAPTBO.** The value of the OLAPTBO static variable is set to the database table owner for the warehouse database.

The Oracle EBS OLTP database has two predefined connection pools. The actual databases in the RPD will depend on the modules licensed by the customer. The connection pools and their functions within each database are listed below.

- Oracle EBS OLTP:
  - **Oracle EBS OLTP DBAuth Connection Pool.** The Oracle EBS OLTP DBAuth Connection Pool is used if database authentication is required.
  - **Oracle EBS OLTP Connection Pool.** The Oracle EBS OLTP Connection Pool is used to connect to the Oracle EBS OLTP system.

  You also need to configure the following Static variables:

  - ORA_EBS_OLTP_DSN. The value of the ORA_EBS_OLTP_DSN static variable is set to the data source name for the Oracle EBS OLTP database.
  - ORA_EBS_OLTP_USER. The value of the ORA_EBS_OLTP_USER static variable is set to the database user name for the Oracle EBS OLTP database.

10.1.1.1 How to Configure Dynamic Data Source Names

This sections explains how to create and configure dynamic data source names.

Dynamic data source names allow the Administrator to set one instance of Analytics server to connect to different data warehouses depending on the user. For this you need to have your user authentication based on an external system (like LDAP), and add the following to your repository:

1. Create new session variables: Session_OLAP_DSN and Session_OLAP_USER
2. Create a Session Init Block which uses 'Oracle Business Analytics Warehouse Repository Initblocks Connection Pool' to populate these session variables based on the user login.
3. Add this Initialization Block to the Execution Precedence list of the Authorization Initialization block.
4. Modify the values of Data Source Name and User Name fields in 'Oracle Data Warehouse Connection Pool' to be VALUEOF(Session_OLAP_DSN) and VALUEOF(Session_OLAP_USER) respectively.

   For the Password field, you should have the user password the same as the user name.
5. Update the field password with the same value as of User Name.

10.1.2 How to Configure Oracle Business Intelligence Applications Repository Connections

The section explains how to configure the Oracle BI connection physical databases and connection pools to connect to your database by doing the following:

- Configuring the Oracle Data Warehouse and, Oracle EBS OLTP database.
- Configuring the Oracle Business Intelligence Applications Repository connection pools.

To configure the Oracle Data Warehouse or Oracle EBS OLTP database:

1. Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd file in the $SAHome\OracleBI\Server\Repository folder.
2. In the Physical pane, double-click the Oracle Data Warehouse object.
3. In the Database list, click your database type.
4. Save the repository.
5. Click Yes to Check Global Consistency.
6. Repeat steps 1-5 for the Oracle EBS OLTP data warehouse objects
7. Click OK when the Warnings are displayed.

To configure the Oracle Business Intelligence Applications Repository connection pools:

1. Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd file in the $SAHome\OracleBI\Server\Repository folder.
2. In the Physical pane:
   a. Double-click the Oracle Data Warehouse Connection Pool within the Oracle Data Warehouse.
   b. Type the database source name in the **Data source name** box.

   c. Type your database User ID in the **User name** box.
   d. Type your password in the **Password** box.
3. Repeat Steps a. to d. above for the other connection pools listed above.
4. Save the repository.
5. Click Yes to Check Global Consistency.
6. Click OK when the Warnings are displayed.

10.1.3 How to Configure Oracle Business Intelligence Applications Repository Variables

The section explains how to configure the Oracle BI repository variables.

To configure the Oracle Business Intelligence Applications Repository variables:

1. Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd file in the $SAHome\OracleBI\Server\Repository folder.
2. On the Manage menu, click Variables.
3. In the Variables Manager dialog, click Static.
4. Edit the OLAP_DSN, OLAP_USER, OLAPTBO, OLTP_DSN, OLTP_USER, ORA_EBS_OLTP_DSN, ORA_EBS_OLTP_USER variables, and close the Variables Manager dialog.
5. On the Manage menu, click Security
6. In the Security Manager dialog, click Users, and then:
   a. Double-click on Administrator user, and type a new password.
   b. Double-click on SADMIN, and type a new password.
   c. Save and close the Security Manager.
7. Save the repository.
8. Click Yes to Check Global Consistency.
9. Click OK when the Warnings are displayed.

10.2 Setting up Date Specific Metrics

The time dimension in the Oracle BI repository for Oracle Business Analytics Warehouse is a standard or canonical time dimension that links to the important time role in each star schema. The Physical table alias used as a canonical time dimension is W_DAY_D_Common.

If a fact table contains a distinct set of metrics that needs to be reported by different dates, the metadata is organized so that each metric is reported by its causal date.

For example, the Invoice fact table has three metrics called Invoice Amount, Fulfill Amount, and Paid Amount, and each of these metrics need to be reported by the corresponding date—Invoice Date, Fulfill Date, and Payment Date. Additional dates in a fact table that a metric could be queried by are known as Secondary dates. These are available to the end users inside a detailed presentation folder. The detailed presentation folder is typically called the Details folder.

In Table 10–1 each of the metrics reflect the activity related to that event for the entire period, for example, Invoice Amount by Invoice Date, Fulfill Amount by Fulfill date, and Payment Amount by Payment Date.

Table 10–1 Invoice Fact Table Example

Date	Invoice Amount	Fulfill Amount	Payment Amount
January	4000	5000	4500

To implement date specific metrics:

1. Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd.

   The OracleBIAnalyticsApps.rpd file is located in the \OracleBI\server\Repository folder.
2. Right-click on Oracle Business Analytics Warehouse in the Physical layer, and create a new physical alias for the fact table.
3. Create Joins for the physical alias which are similar to the base fact table.

   The Join to the date dimension is changed to use the date role in question.
4. Create a new logical table source in the logical fact table that maps the metrics for the physical fact alias.

   The grain of the fact table is the same as the base fact table.

   > **Note:** You need to map each metric to one logical table source at the Detail Level.

10.3 Setting Up Additional Time Series Metrics for Oracle Business Analytics Warehouse

The Oracle BI repository provides a framework to add Period Ago metrics. The repository for Oracle Business Analytics Warehouse is preconfigured with pre-mapped period ago metrics, however you can map other metrics by using the following procedure.

To set up additional time series metrics:

1. Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd.
2. Right-click on Oracle Business Analytics Warehouse in the Physical layer, and create a new Period Ago physical alias table.
3. Create additional tables in the Physical Layer for each Period Ago alias required.

   For example, Quarter Ago, Year Ago, and so on.

   These aliases need to have the same joins as the base fact table, except for the date join, which you can change in the next step. Setting up this alias is easier to accomplish by copying the base table.
4. Change the join to the date dimension (W_DAY_D) to use the appropriate Period Ago Key.
5. Map the Period Ago metrics in the logical table using the new fact alias by creating a new logical table source under the fact table.
6. Set the content pane levels for the period ago logical table source, to specify the level of the source data.

These settings are the same as the base fact table.

7. Save and close the OracleBIAnalyticsApps.rpd file.

10.4 Setting Up Additional Dimension Tables for Oracle Business Analytics Warehouse

Oracle Business Analytics Warehouse is preconfigured to map dimension tables required for analysis. The physical layer in the Oracle BI repository provides several other dimensional table keys that can be used for certain specific analysis. If you need to set up any of the additional dimensions tables to the physical layer, perform the following procedure.

To set up additional dimension tables:

1. Validate that the dimension table key is resolved appropriately for the data source that you are using.

> **Note:** Dimension tables do not apply to every source system.

2. Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd.
3. Add a dimension table alias in the physical layer.
4. Join the dimension table alias to the fact table alias using the appropriate keys.
5. Save and close the OracleBIAnalyticsApps.rpd file.

10.5 About the Period Ago Keys for Oracle Business Analytics Warehouse

The Period Ago Key fields are used to set up the time series metrics like Year Ago, Quarter Ago, and so on. The Period Ago Key fields represent metrics for a prior period, for example, Quarter Ago Revenue, Year Ago Revenue, and so on. Oracle Business Analytics Warehouse is preconfigured with a set of fields in the W_DAY_D table. These fields are:

- MONTH_AGO_WID
- QUARTER_AGO_WID
- TRIMESTER_AGO_WID
- WEEK_AGO_WID
- YEAR_AGO_WID

These fields are used in joins to Oracle Business Analytics Warehouse fact tables to achieve the period ago metrics. The joins in Oracle Business Analytics Warehouse uses the Period Ago fields in the W_DAY_D table.

10.6 About Oracle BI Time Repository Variables

The Oracle BI repository is preconfigured with variables that are used for both reporting and internal usage.

Table 10–2 lists the Oracle BI repository date variables and their descriptions.

Table 10–2 Oracle BI Repository Date Variables

Variable Name	Description
CAL_MONTH_YEAR_AGO	Returns the value of Previous Year Month in the YYYY/MM format.
CURRENT_BALANCE_DK_AP	Returns the value of the last date key for the available Accounts Payable balance. It is used in Accounts Payable Account Balance Computation.
CURRENT_BALANCE_DK_AR	Returns the value of the last date key for the available Accounts Receivables balance. It is used in Accounts Receivable Account Balance Computation.
CURRENT_BALANCE_DK_GL	Returns the value of the last date key for the available General Ledger balance. It is used in General Ledger Account Balance Computation.
CURRENT_DAY	Returns the value of Current Date in the MM/DD/YYYY format.
CURRENT_FSCL_MONTH	Returns the value of Current Fiscal Month in the YYYY/MM format.
CURRENT_FSCL_QUARTER	Returns the value of Current Quarter in the YYYY Q n format.
CURRENT_FSCL_WEEK	Returns the value of Current Fiscal Week in the YYYY Week nn format.
CURRENT_FSCL_YEAR	Returns the value of Current Fiscal Year in the FYYYYY format.
CURRENT_JULIAN_DAY_NUM	Returns the value of Current Julian Date Number.
CURRENT_MONTH	Returns the value of Current Month in the YYYY/MM format.
CURRENT_QTR	Returns the value of Current Quarter in YYYY Q n format.
CURRENT_WEEK	Returns the value of Current Week in the YYYY Week nn format.
CURRENT_YEAR	Returns the value of Current Year in the YYYY format.
FSCL_MONTH_YEAR_AGO	Returns the value of Previous Year Fiscal Month in YYYY/MM format.
FSCL_QTR_YEAR_AGO	Returns the value of Previous Year Quarter in YYYY Q n format.
NEXT_FSCL_MONTH	Returns the value of Next Fiscal Month in the YYYY / MM format.
NEXT_FSCL_QUARTER	Returns the value of Next Quarter in the YYYY Q n.
NEXT_FSCL_WEEK	Returns the value of Next Fiscal Week in the YYYY Weeknn format.
NEXT_FSCL_YEAR	Returns the value of Next Fiscal Year in the FYYYYY format.
NEXT_MONTH	Returns the value of Next Month in the YYYY / MM format.
NEXT_QUARTER	Returns the value of Next Quarter in the YYYY Q n.
NEXT_WEEK	Returns the value of Next Week in the YYYY Weeknn format.
NEXT_YEAR	Returns the value of Next Year in the YYYY format.
PREVIOUS_FSCL_MONTH	Returns the value of Previous Fiscal Month in the YYYY/MM format.
PREVIOUS_FSCL_QUARTER	Returns the value of Previous Quarter in the YYYY Q n format.

Table 10–2 (Cont.) Oracle BI Repository Date Variables

Variable Name	Description
PREVIOUS_FSCL_WEEK	Returns the value of Previous Fiscal Week in the YYYY Weeknn format.
PREVIOUS_FSCL_YEAR	Returns the value of Previous Fiscal Year in the FYYYYY format.
PREVIOUS_MONTH	Returns the value of Previous Month in the YYYY/MM format.
PREVIOUS_QUARTER	Returns the value of Previous Quarter in the YYYY Q n.
PREVIOUS_WEEK	Returns the value of Previous Week in the YYYY Weeknn format.
PREVIOUS_YEAR	Returns the value of Previous Year in the YYYY format.
REF_JULIAN_DATE	Stores the start date of the Julian calendar and should not be changed.
REF_JULIAN_DATE_NUM	Stores the Julian number for the start of the Julian calendar and should not be changed.
TIME_OFFSET	Returns the difference between the current date and a given number of days value. It is primarily used for testing to simulate an earlier or later date. You could set the variable to the number of days you want the preceding date variables to be moved back.
YEAR_AGO_DAY	Returns the value of year ago date in the mm/dd/yyyy format.

10.7 About User Authentication

You need to configure the user authentication of the Oracle Business Analytics Warehouse repository depending on your requirements. The Oracle Business Analytics Warehouse supports various authentication modes, for example, Repository authentication, Database authentication, and LDAP.

For more information about configuring user authentication, see Oracle Business Intelligence Server Administration Guide.

10.8 About the Security or Visibility Configuration

The Oracle Business Analytics Warehouse repository is preconfigured with a set of user groups. These groups control the visibility of catalogs in the presentation layer.

For more information on adding a user to repository user group, see Oracle Business Intelligence Server Administration Guide.

Table 10–3 lists the groups in the Oracle Business Analytics Warehouse repository.

Table 10–3 Repository User Groups

Repository User Group	Description
Administrators	The Administrators user group has all rights and privileges. It cannot be removed.
Agent Scorecard User	This user group is able to view Agent Scorecard application content.
AP Analyst	This user group is able to view application content for Oracle Payables Analytics.
AP Manager	This user group is able to view high-level application content for Oracle Payables Analytics.

Table 10–3 (Cont.) Repository User Groups

Repository User Group	Description
AR Analyst	This user group is able to view application content for Oracle Receivables Analytics.
AR Manager	This user group is able to view high-level application content for Oracle Receivables Analytics.
CFO	This user group is able to view most of the Oracle Financial Analytics application content.
Contact Center and Agent Performance Analyst	This user group is able to view Contact Center Telephony Analytics and Agent Performance application content.
Contact Center and Agent Performance User	This user group is able to view a subset of Contact Center Telephony Analytics and Agent Performance application content.
Contact Center Sales Analyst	This user group is able to view Contact Center Telephony Analytics and Order Management Analytics application content.
Contact Center Sales User	This user group is able to view a subset of Contact Center Telephony Analytics and Order Management Analytics application content.
Controller	This user group is able to view application content for Oracle General Ledger and Profitability Analytics and Siebel Profitability Analytics.
Customer Service Analyst	This user group is able to view Customer Service for Oracle Contact Center Telephony Analytics application content.
Customer Service User	This user group is able to view a subset of Customer Service for Oracle BI Contact Center Telephony Analytics application content.
Contact Center Telephony Analytics User	This user group is able to view Oracle BI Contact Center Telephony Analytics application content.
Financial Analyst	This user group is able to view Oracle Financial Analytics application content.
Human Resources Analyst	This user group is able to view Oracle HR Analytics application content.
Human Resources Vice President	This user group is able to view high-level application content for Oracle HR Analytics application.
Inventory Analyst	This user group is able to view application content for Oracle's Procurement and Spend Analytics Family of Products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics).
Inventory Manager	This user group is able to view high-level application content for Oracle's Procurement and Spend Analytics Family of Products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics).
Primary Owner-Based Security	Used for securing owner-based data elements that come from the transactional system.
Primary Position-Based Security	Used for securing position-based data elements that come from the transactional system.
Purchasing Buyer	This user group is able to view Oracle's Procurement and Spend Analytics Family of Products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) content pertaining to purchasing.

Table 10–3 (Cont.) Repository User Groups

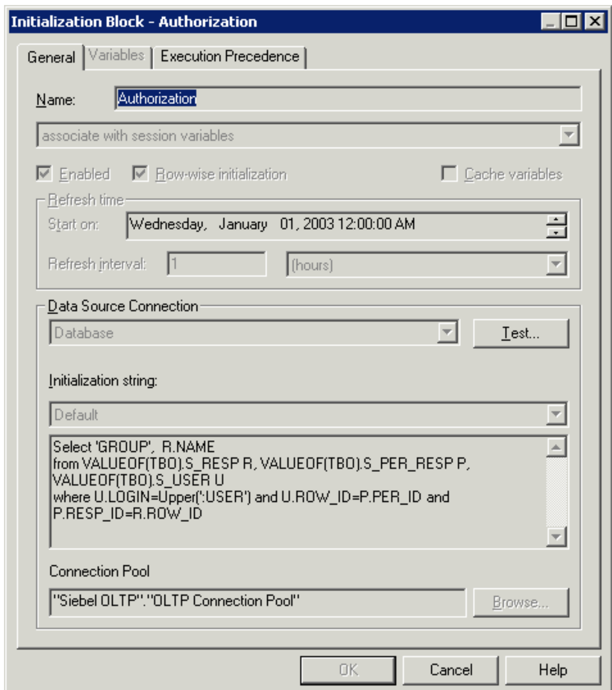
Repository User Group	Description
Sales Executive Analytics	This user group is able to view high-level application content for the Order Management Analytics application.
Sales Manager	This user group is able to view most of the high-level application content for Oracle BI Contact Center Telephony Analytics application.
Sales Manager Analytics	This user group is able to view most of the high-level application content for Oracle BI Contact Center Telephony Analytics application.
Sales Operations Analytics	This user group is able to view operational application content for Oracle BI Contact Center Telephony Analytics application.
Sales Representative Analytics	This user group is able to view low-level application content for Oracle BI Contact Center Telephony Analytics application.
Sales Rev and Fulfill Analyst	This user group is able to view the content for Oracle BI Contact Center Telephony Analytics Revenue and Fulfillment application.
Sales Rev and Fulfill Exec	This user group is able to view the high-level application content for Oracle BI Contact Center Telephony Analytics Revenue and Fulfillment application.
Sales Rev and Fulfill Mgr	This user group is able to view most of the high-level application content for Oracle BI Contact Center Telephony Analytics Revenue and Fulfillment application.
Sales Rev and Fulfill Rep	This user group is able to view low-level application content for Order Management Analytics Revenue and Fulfillment application.
Sales Revenue Analyst	This user group is able to view the content for Oracle BI Contact Center Telephony Analytics Revenue application.
Sales Revenue Exec	This user group is able to view the high-level application content for Oracle BI Contact Center Telephony Analytics Revenue application.
Sales Revenue Mgr	This user group is able to view most of the high-level application content for Oracle BI Contact Center Telephony Analytics Revenue application.
Sales Revenue Rep	This user group is able to view low-level application content for Oracle BI Contact Center Telephony Analytics Revenue application.
Service Delivery and Costs Analyst	This user group is able to view Service Delivery and Costs for Oracle BI Contact Center Telephony Analytics application content.
Service Delivery and Costs User	This user group is able to view a subset of Service Delivery and Costs for Oracle BI Contact Center Telephony Analytics application content.
Supplier Performance Analyst	This user group is able to view Oracle's Procurement and Spend Analytics Family of Products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) content pertaining to supplier performance.
Supplier Performance Manager	This user group is able to view high-level content for Oracle's Procurement and Spend Analytics Family of Products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) content pertaining to supplier performance.
Supply Chain Executive	This user group is able to view Oracle's Procurement and Spend Analytics Family of Products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) content.

10.9 About the Group Variable

The Group variable determines the membership of a user in the various security groups. You need to associate users to the appropriate groups defined in the OracleBIAnalyticsApps.rpd for the security filters to take effect.

If you are using the Oracle Business Intelligence Applications with the Siebel CRM system, then you can leverage the Authorization session init block to populate the Group variable with the appropriate Siebel Responsibilities, which associates users to what they are allowed to see in the Oracle BI Application by using their Siebel Responsibility. For more information about configuring the Group variable, see *Oracle Business Intelligence Web Administration Guide*. The screen shot below shows an example of an initialization block that associates a user to a Group membership.

Figure 10–1 Screenshot of the Initialization Block - Authorization screen



10.10 About Configuring Usage Tracking for Oracle Business Analytics Warehouse

Oracle Business Analytics Warehouse supports the accumulation of usage tracking statistics. The Oracle BI repository for Oracle Business Analytics Warehouse is preconfigured with a connection pool to enable the population of the Usage Tracking log table.

You need to configure this connection pool to connect to the S_NQ_ACCT table. For more information the Usage Tracking application administering Usage Tracking, see the Oracle Business Intelligence Server Administration Guide.

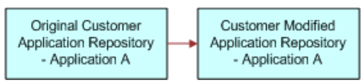
10.11 About the Incremental Deployment of the Oracle BI Applications Repository

Oracle Business Analytics Warehouse consist of various families of Oracle Business Intelligence Applications, for example, Supplier Performance Analytics, Contact Center Telephony Analytics, General Ledger & Profitability Analytics, and so on. You can purchase these applications at different times. You can customize functionality and incrementally add new application or applications.

This section describes the procedure for deploying multiple applications. You can repeat the procedure to add applications incrementally.

The figure below shows a single Oracle Business Analytics Warehouse application environment. During installation, you will be asked to specify the application module(s) you have licensed, and the installer will extract the metadata project(s) corresponding to this module(s) into one repository file. You can then modify the Oracle BI repository to suit your business needs.

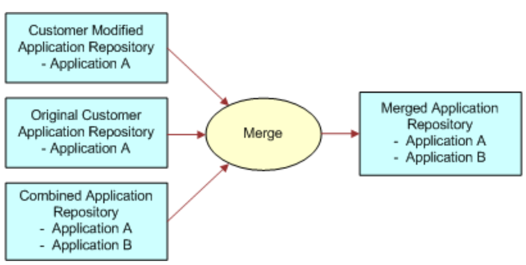
Figure 10–2 Oracle Business Analytics Warehouse environment



When you purchase another Oracle Business Analytics Warehouse application, you need to extract a new application repository, containing the metadata for all the modules that you have licensed. Use the Administration merge utility to perform a three-way merge of the original repository, the modified repository, and the combined repository. For more information on merging repositories, see *Oracle Business Intelligence Server Administration Guide*.

The merged repository preserves your modifications from the original Oracle BI repository and appends the information with the new Oracle BI repository, as shown in the figure below.

Figure 10–3 Merging with an Oracle Business Intelligence Applications Repository



You can repeat this merging procedure to add more Oracle Business Analytics Warehouse applications to the Oracle BI repository

Part IV

Customizing Oracle Business Intelligence Applications

Part IV explains how to customize the out-of-the-box behavior in Oracle Business Intelligence Applications, and contains the following chapters:

- Chapter 11, "Customizing the Oracle Business Analytics Warehouse"

Note: For a high level road map for installation, configuration, and customization steps for Oracle Business Intelligence Applications, see Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle Business Intelligence Applications With ODI".

11

Customizing the Oracle Business Analytics Warehouse

This chapter describes concepts and techniques for customizing the E-LT functionality in Oracle Business Intelligence Applications, and contains the following topics:

- Section 11.1, "Overview of Customization in Oracle Business Intelligence Applications"
- Section 11.2, "Category 1 Customizations: Adding Columns to Existing Fact or Dimension Tables"
- Section 11.3, "Category 2 Customizations: Adding Additional Tables"
- Section 11.4, "Category 3 Customizations: Adding New Data as a Whole Row into a Standard Dimension Table"
- Section 11.5, "Customizing Stored Lookups, Domain Values, and Adding Indexes"

11.1 Overview of Customization in Oracle Business Intelligence Applications

This section provides an overview of customization in Oracle Business Intelligence Applications, and contains the following topics:

- Section 11.1.1, "What is Customization in Oracle Business Intelligence Applications?"
- Section 11.1.3, "About the Impact of Patch Installation on Customizations"

11.1.1 What is Customization in Oracle Business Intelligence Applications?

In Oracle Business Intelligence Applications, customization is defined as changing the out-of-the-box behavior to enable you to analyze new information in your business intelligence dashboards. For example, you might want to add a column to a dashboard by extracting data from the field HZ_CUST_ACCOUNTS.ATTRIBUTE1 and storing it in the Oracle Business Analytics Warehouse in the X_ACCOUNT_LOG field.

The type of data source that you have determines the type of customization that you can do. Data sources can be one of the following types:

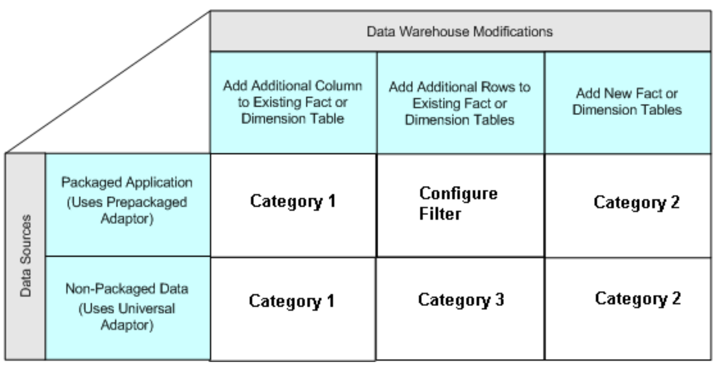
- Packaged applications (for example, Oracle EBS), which use prepackaged adapters.
- Non-packaged data sources, which use the Universal adapter.

Customizations are grouped into the following categories:

- **Category 1.** In a Category 1 customization, you add additional columns from source systems that have pre-packaged adapters and load the data into existing Oracle Business Analytics Warehouse tables. For more information about performing Category 1 customizations, see Section 11.2, "Category 1 Customizations: Adding Columns to Existing Fact or Dimension Tables".
- **Category 2.** In a Category 2 customization, you use pre-packaged adapters to add new fact or dimension tables to the Oracle Business Analytics Warehouse. Category 2 customizations normally require that you build new SDE and SIL mappings. For more information about performing Category 2 customizations, see Section 11.3, "Category 2 Customizations: Adding Additional Tables".
- **Category 3.** In a Category 3 customization, you use the Universal adapter to load data from sources that do not have pre-packaged adapters. For more information about performing Category 3 customizations, see Section 11.4, "Category 3 Customizations: Adding New Data as a Whole Row into a Standard Dimension Table".

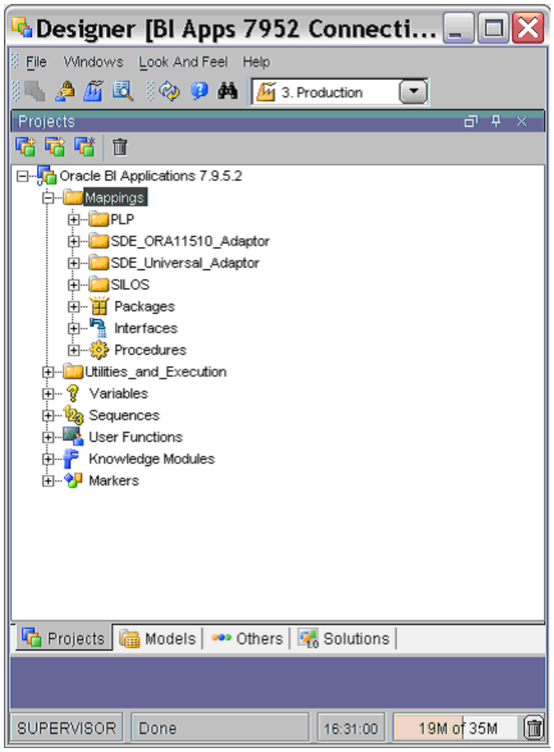
The figure below summarizes the category of customization that you can perform for each type of data source and type of modification.

Figure 11–1 Supported customizations based on data source



For detailed information about tables and naming conventions, see *Oracle Business Analytics Warehouse Data Model Reference*.

When you customize E-LT Packages and Interfaces, you usually work in the \Oracle BI Applications 7.9.5.2\Mappings folder in the Models view in ODI Designer.



Note: When you make customizations to any object, create a version of that object before and after the modifications. These versions will enable you to revert to previous functionality if required, and also to manage customizations after a patch upgrade.

11.1.2 About the Customization Process

This chapter explains how to customize your E-LT functionality, after you have performed a Business Analysis and Technical Analysis. This chapter does not cover the other typical tasks that you need to perform, as follows:

- Business Analysis - before you start customization, you typically analyze your current BI dashboards to determine the changes you need to support your business or organization.
- Technical Analysis - when you have agreed your business requirements, you need to determine the technical changes you need to make, by identifying source tables, staging tables, target tables, and ODI Packages and Interfaces that you need to modify.
- RPD Modification - having made the customizations in the E-LT functionality, you need to modify your RPD to expose the new data in your dashboards. For more information about RPD modification, refer to the Oracle Business Intelligence Enterprise Edition documentation library.

11.1.3 About the Impact of Patch Installation on Customizations

This section explains what you must do to re-apply a customization that you have made if you apply an Oracle Business Intelligence Applications patch that overwrites that customization. For example, if you install a patch that modifies the Supply Chain and Order Management application, you might need to manually re-apply customizations that you have made to the Supply Chain and Order Management application.

Before you apply a patch, you need to 'version' your existing Work Repository (for example, in ODI Designer so that you can detect and re-apply customizations after the patch installation.

A patch only installs changed repository objects, not the whole Work Repository. Therefore, you only need to re-apply customizations to mappings that have been changed by the patch. For example, if a patch only modifies the Supply Chain and Order Management application, you only need to manually re-apply customizations that you have made to the Supply Chain and Order Management application. Customizations in other applications are not affected by the patch.

To minimize the amount of effort required to re-apply customizations after a patch installation, Oracle recommends that you follow the customization methodology that is described in this chapter.

To maintain customizations after a patch installation:

1. Before you apply a patch, version your existing Work Repository and customized objects by following the steps in Section 11.1.3.1, "How to Version A Work Repository And Work Repository Objects".
2. Apply the patch.
3. After you apply a patch, re-apply customizations that were overwritten during the patch installation by following the steps in Section 11.1.3.2, "How to Re-apply Customizations After a Patch Installation".
4. Create a version of the customized objects.

Notes

- When you customize objects, you must evaluate the options and determine the best approach for your environment. If you find that the custom object approach allows the ELT to run in an acceptable amount of time, then this is the preferred approach. If the custom object causes the ELT process to take too long, you might want to consider incorporating the extension into an existing object package or interface.
- When you add custom columns to the Oracle Business Analytics Warehouse, you must make the change in all Oracle Business Analytics Warehouse objects (for example, the W_XXX_D/_F tables as well as the staging tables).

11.1.3.1 How to Version A Work Repository And Work Repository Objects

This section explains how to 'version' your Work Repository in ODI Designer so that you can detect and re-apply customizations after a patch installation.

After you apply a patch, you must re-apply customizations by following the steps in Section 11.1.3.2, "How to Re-apply Customizations After a Patch Installation"

After you 'version' an object in ODI Designer, use the Version Browser to manage versions (that is, select File, then Version Browser).

To re-apply customizations after a patch installation:

1. In ODI Designer, display the Projects view.
2. Right-click the Oracle BI Applications 7.9.5.2 project and choose Version, then Create to display the Create: <*Project Name*> dialog.
3. Use the Create: <*Project Name*> dialog to specify a unique version number and optional description, and click OK.

4. Run the package 'Export Changed Objects' in the folder 'Oracle BI Applications 7.9.5.2\Utilities_and_Execution\Utilities\User.

   This package will create a file called 'List_Of_Modified_Objects.txt' in the folder location that you specify. This TXT file contains a list of customized objects. This package will also create XML export files for the changed objects in the folder location that you specify, within the date range.

11.1.3.2 How to Re-apply Customizations After a Patch Installation

This section explains how to re-apply customizations after you have applied a patch. Initially, any customized objects in the area affected by a patch are over-written with a new version of the object, which overwrites the customization. You use the Version Comparison utility in ODI Designer on the customized objects to compare the new version of an object and the original customized version of that object. When Version Comparison utility has performed the comparison, you can do one of the following:

- Manually re-apply the customizations to the object.
- Retain the new non-customized version object provided by Oracle (in the patch).
- Restore the previous customized version of the object.

For example, before a patch installation, you might have modified the expression of the DUE_DAY_OF_MONTH column in the Interface SDE_ORA_APTermsDimension.SQ_RA_TERMS. After the patch installation, the expression of the DUE_DAY_OF_MONTH column might be reverted to AP_TERMS_LINES.DUE_DAY_OF_MONTH. To reapply the customization, check the expression difference using the comparison dialog, and then replace the current version column expression with the previous version of the column expression.

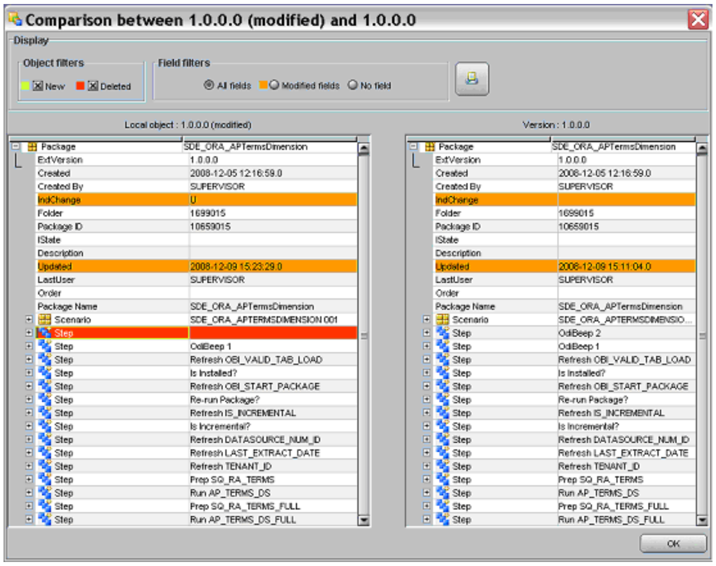
Before you can re-apply customizations, you must have versioned your Work Repository and customized objects by following the steps in Section 11.1.3.1, "How to Version A Work Repository And Work Repository Objects".

To re-apply a customization after a patch installation:

1. In ODI Designer, locate the object that you want to compare with the original (customized version).

   For example, if you customized the Package SDE_ORA_APTermsDimension in the SDE_ORA_APTermsDimension project folder, locate this object in the Project tab.

2. Right-click the object (for example, a package), and choose Version, then 'Compare with version' to display the Compare With dialog.
3. Use the Compare With dialog to select the original version, then click OK to display the Comparison dialog.



4. Use the Comparison dialog to determine what customizations need to be re-applied.

5. Manually re-apply the customizations on the respective objects.

   For example, to restore a previous (customized) version of an object, right click on the object and select Version, then Restore, and select the version of the object that contains the customization.

   Alternatively, you could manually apply a customization to the new version of the object by editing the new object in ODI Designer and making the changes again. For example, you edit an Interface, and change the expression for a column in the Target Datastore by editing the value in the **Implementation** field.

6. Repeat steps 1 to 5 for every customization that you need to re-apply.

7. Create a new version of each customized object.

11.2 Category 1 Customizations: Adding Columns to Existing Fact or Dimension Tables

Category 1 customizations add additional columns from source systems that have pre-packaged adapters and load the data into existing Oracle Business Analytics Warehouse tables.

This section contains the following topics:

- Section 11.2.1, "About Extending Mappings"
- Section 11.2.2, "Typical Steps to Extend Mappings in the Oracle Business Analytics Warehouse"
- Section 11.2.3, "Example of Extending the Oracle Business Analytics Warehouse"
- Section 11.2.4, "Other Types of Customizations Requiring Special Handling"

11.2.1 About Extending Mappings

Category 1 customizations involve extracting additional columns from source systems for which pre-packaged adapters are included (for example, Oracle) and loading the data into existing Oracle Business Analytics Warehouse tables. For Category 1 customizations, data can also come from non-packaged sources, but this section assumes that the sources have already been mapped with a Universal adapter and only need to be extended to capture additional columns. (The initial mapping of a Universal adapter is considered a Category 3 customization. For information, see Section 11.4, "Category 3 Customizations: Adding New Data as a Whole Row into a Standard Dimension Table".)

In order to see additional columns in the Oracle Business Analytics Warehouse, the columns must first be passed through the ELT process. The existing mappings and tables are extensible. Oracle Business Intelligence Applications provides a methodology to extend preconfigured mappings to include these additional columns and load the data into existing tables.

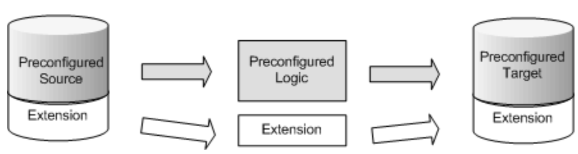
Oracle Business Intelligence Applications recognizes two types of customization: extension and modification. The supported extension logic allows you to add to existing objects. For example, you can extract additional columns from a source, pass them through existing mappings, and populate new columns added to an existing table. Generally, Oracle Business Intelligence Applications does not allow you to modify existing logic or columns. You should not change existing calculations to use different columns, and you should not remap existing columns to be loaded from different sources.

For example, if you want to calculate revenue differently from the existing logic, you should create a new transformation and connect that calculation to a new column, for example, X_REVENUE. You can then remap the Oracle Business Intelligence repository to point to the new X_REVENUE column.

Most mappings have a single placeholder column, named X_CUSTOM, that marks a *safe path* through the ODI interfaces. If you add transformations to the mapping, they should follow the same route through the mapping as X_CUSTOM.

In the figure below, the preconfigured logic is shaded in gray. You should not modify anything contained within these objects. You should add customizations to existing objects rather than creating new packages and interfaces, which allows them to run parallel to the existing logic.

Figure 11–2 Preconfigured logic and customizations



11.2.2 Typical Steps to Extend Mappings in the Oracle Business Analytics Warehouse

The most common reason for extending the Oracle Business Analytics Warehouse is to extract existing columns from a source system and map them to an existing Oracle Business Analytics Warehouse table (either fact or dimension). This type of change typically requires you to extend the interfaces within a SIL package. If the data is coming from a packaged source, then you will also need to extend the interfaces within an appropriate SDE adapter package. If the data is coming from a non-packaged source, then you must use a Universal adapter package. If an appropriate package does not already exist, you will need to create a Universal adapter package with interfaces.

To extend an ODI package in the Oracle Business Analytics Warehouse:

1. Create a version of the folder that contains the ODI Packages and Interfaces that you need to customize.

   For example, in ODI Designer, display the Projects view, right-click on a folder, and choose Version, then Create.

2. Extend the source and target tables by making changes to the tables in the database.

   You then can use ODI to reverse-engineer the additive changes on the source and target definitions into models in ODI (which replaces the existing definitions), or manually edit the existing definition.

   As a best practice, Oracle recommends that you name custom columns with a X_ prefix to make it easier to distinguish custom columns that have been added to an existing table and to ensure there are no name conflicts with any columns Oracle might add later to that table.

3. Extend the SDE Package and Interfaces by mapping the additional columns, as follows:

   a. In the ODI Package editor, display the Projects view, and open the package.

      For example, you might open the Oracle BI Applications 7.9.5.2\Mappings\SDE_ORA11510_Adaptor\SDE_ORA_OrganizationDimension_Customer folder, and edit the SDE_ORA_OrganizationDimension_Customer Package.

   b. Display the Diagram tab.

   c. Edit either the SQ_BCI_ interface or the LKP_ interface to display the Interface: <*Name*> dialog.

      For example, the SDE_ORA_OrganizationDimension_Customer Package, you might edit the SQ_BCI_CUSTOMERS Interface.

   d. Display the Diagram tab.

   e. Map the new column in the Source table to the Target Datastore.

   f. Click OK to save the changes.

   g. In the ODI Package editor, edit the last interface in the sequence (for example, named RUN <target table name>).

   h. Display the Diagram tab.

   i. Map the new column in the Source table to the Target Datastore.

   j. Click OK to save the changes.

   k. Repeat steps a. to j. for both branches within the package (both full load and incremental load).

      For example, in the SDE_ORA_OrganizationDimension_Customer Package, you might also edit the SQ_BCI_CUSTOMERS_FULL Interface.

      For a detailed example, see Section 11.2.3.2, "Example of Extracting Data from an Oracle EBS 11.5.10 Data Packaged Source into the ODI Staging Area"

4. Repeat step 3 for the SIL package and interfaces.

   For example, you might open the Oracle BI Applications 7.9.5.2\Mappings\SILOS\SIL_OrganizationDimension folder, and edit the SIL_OrganizationDimension Package.

For a detailed example, see Section 11.2.3.3, "Example of Loading Data from the Staging Area into an Existing Target Table"

5. Regenerate the scenarios for the packages that you have modified.

   You are now ready to perform E-LT with the modified scenarios.

11.2.3 Example of Extending the Oracle Business Analytics Warehouse

This section contains a worked example of adding additional columns from source systems that have pre-packaged adapters and loading the data into existing Oracle Business Analytics Warehouse tables (known as a Category 1 customization).

This section contains the following topics:

- Section 11.2.3.1, "Overview to the Example"
- Section 11.2.3.2, "Example of Extracting Data from an Oracle EBS 11.5.10 Data Packaged Source into the ODI Staging Area"
- Section 11.2.3.3, "Example of Loading Data from the Staging Area into an Existing Target Table"
- Section 11.2.3.4, "Tips for Modifying the SQ_BCI_ Interface"
- Section 11.2.3.5, "Including a Source Table for the Change Capture Process"

11.2.3.1 Overview to the Example

In this example, a company has identified additional fields in a source system table HZ_CUST_ACCOUNTS that need to be added to the Oracle Business Analytics Warehouse table W_ORG_D. Data is passed from an existing source table to an existing target table, known as a category 1 customization. The company uses an extension field to capture information related to organizations referred to as 'ACCOUNT_LOG.' In addition, the company wants to include the name of the person who last updated the record as an attribute of the organization.

In this example, you want to extract information from the following two fields that are not extracted by the out-of-the-box application:

- HZ_CUST_ACCOUNTS.ATTRIBUTE1

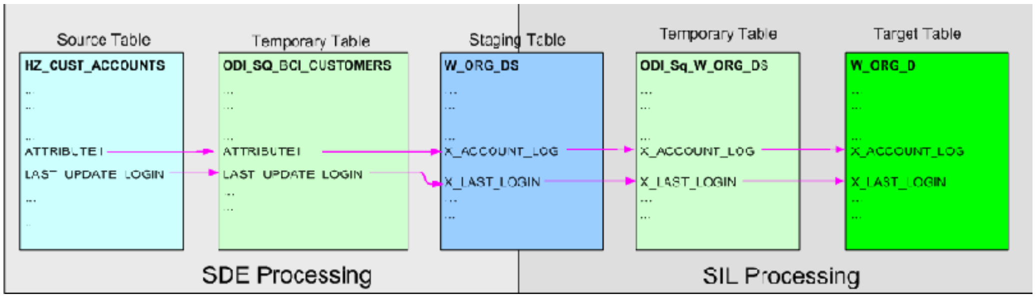
  ATTRIBUTE1 is currently not extracted from the source table HZ_CUST_ACCOUNTS into the temporary table ODI_SQ_BCI_CUSTOMERS.

- HZ_CUST_ACCOUNTS.LAST_UPDATE_LOGIN

  LAST_UPDATE_LOGIN is currently extracted from the source table HZ_CUST_ACCOUNTS into the temporary table ODI_SQ_BCI_CUSTOMERS, but is not loaded into the staging table W_ORG_DS.

The diagram below shows the two fields ATTRIBUTE1 and LAST_UPDATE_LOGIN as they pass from the source system table to the target table via the tables: HZ_CUST_ACCOUNTS to ODI_SQ_BCI_CUSTOMERS to W_ORG_DS to ODI_Sq_W_ORG_DS to W_ORG_D.

Figure 11–3 Passing two new fields from the source table to the target table



The customization is done in two parts, as follows:

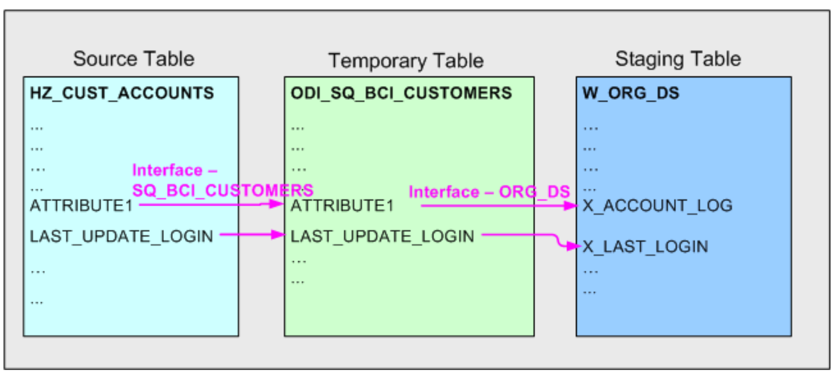
- SDE Processing, which extracts data from the source system and loads it into the staging area (for more information, see Section 11.2.3.2, "Example of Extracting Data from an Oracle EBS 11.5.10 Data Packaged Source into the ODI Staging Area").
- SIL Processing, which extracts data from the staging area and loads it into the target table (for more information, see Section 11.2.3.3, "Example of Loading Data from the Staging Area into an Existing Target Table").

11.2.3.2 Example of Extracting Data from an Oracle EBS 11.5.10 Data Packaged Source into the ODI Staging Area

This section shows how data is extracted from an existing source table into the staging area.

The diagram below shows the new E-LT mappings that you need to load the new data into the staging area, and the ODI Interfaces that you need to modify.

Figure 11–4 Required new mappings for loading data into the staging area



Note: The diagram above only shows the incremental interfaces.

To customize the E-LT process to load these two fields into the staging area, you need to:

- Extract the HZ_CUST_ACCOUNTS.ATTRIBUTE1 value from the source table HZ_CUST_ACCOUNTS into the temporary table ODI_SQ_BCI_CUSTOMERS using the Interfaces SQ_BCI_CUSTOMERS and SQ_BCI_CUSTOMERS_FULL.

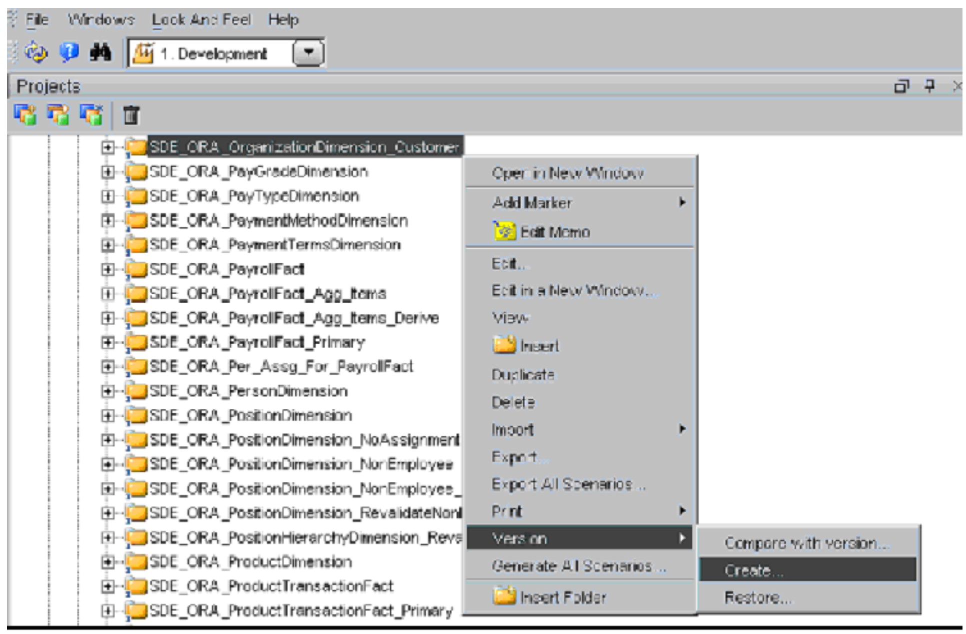
  Then, load the ODI_SQ_BCI_CUSTOMERS.ATTRIBUTE1 value from the temporary table ODI_SQ_BCI_CUSTOMERS into the X_ACCOUNT_LOG field in the staging table W_ORG_DS using the Interfaces ORG_DS and ORG_DS_FULL.

- Load the SQ_BCI_CUSTOMERS.LAST_UPDATE_LOGIN value from the temporary table ODI_SQ_BCI_CUSTOMERS into the X_LAST_LOGIN field in the staging table W_ORG_DS using the Interfaces ORG_DS and ORG_DS_FULL.

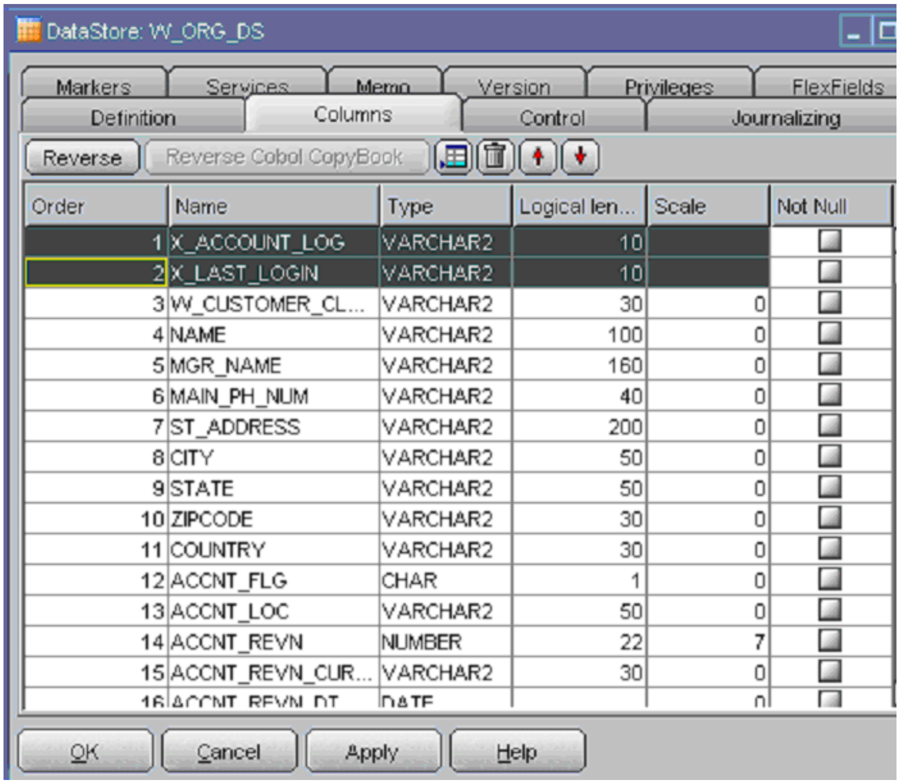
  Note: Remember that LAST_UPDATE_LOGIN value is already extracted from the source table HZ_CUST_ACCOUNTS into the temporary table ODI_SQ_BCI_CUSTOMERS, but is not loaded into the staging table W_ORG_DS.

To extract data from an Oracle EBS 11.5.10 Data Packaged Source:

1. In ODI Designer, display the Projects view, expand the 'Oracle BI Applications 7.9.5.2'\Mappings\SDE_ORA11510_Adaptor folder.
2. Right-click on the SDE_ORA_OrganizationDimension_Customer folder, and choose Version, then Create to display the Create: *<Object>* dialog, and specify a unique version number and optional version description.



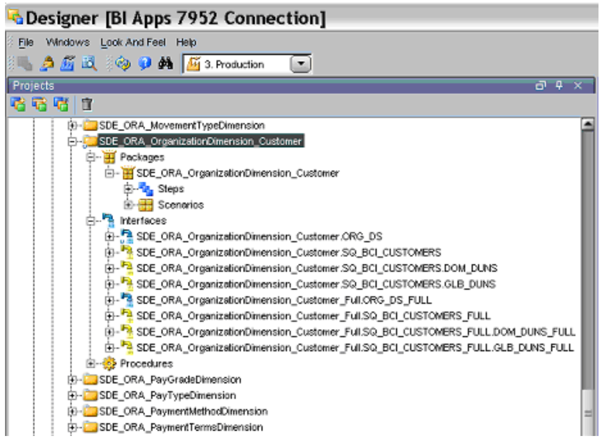
3. Display the Models view, expand the Dimension Stage folder, and edit the W_ORG_DS data store to display the DataStore: *<Name>* dialog, and display the Columns tab.



4. Create the following columns:
   - X_ACCOUNT_LOG (VARCHAR2(10))
   - X_LAST_LOGIN (VARCHAR2(10))
5. In the Models view, right click on the model 'Oracle BI Applications 7.9.5.2' and select Generate DDL to display the Generate DDL dialog.

   The Generate DDL option deploys the changes in the database.
6. Select the check-box in the Synchronize column next for the W_ORG_DS table.
7. Click the (...) button to the right of the **Generation Folder** field to display the **Select a folder** dialog, and select the \Utilities\System folder, and click OK.
8. When the Procedure: DDL <*Name*> dialog is displayed, click Execute.

   Display ODI Operator and make sure that the procedure executes successfully.
9. Display the Projects view, expand the Mappings folder, and expand the SDE_ORA_OrganizationDimension_Customer folder.



10. Edit the Interface SDE_ORA_OrganizationDimension_Customer.SQ_BCI_CUSTOMERS to display the Interface: <*Name*> dialog, and do the following:

    a. Display the Diagram tab, and select the ATTRIBUTE1 field in the Target Datastore area.

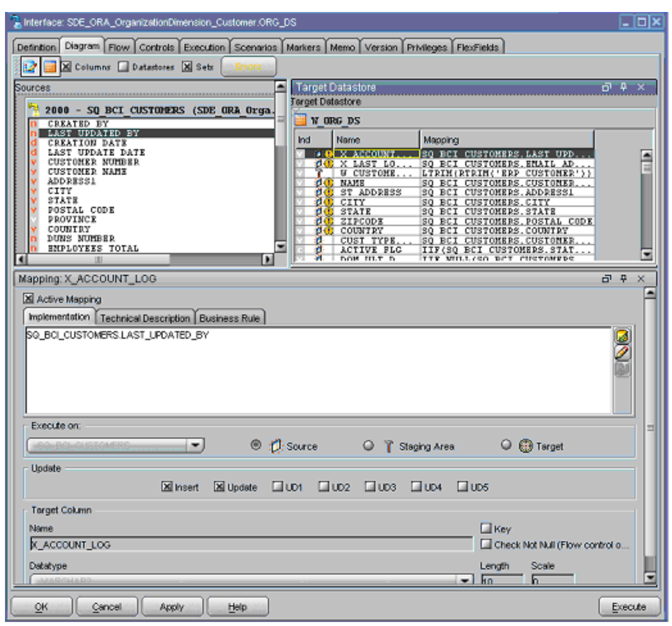
    b. Use the Launch Expression Editor icon to display the Expression Editor dialog, and use this dialog to select HZ_CUST_ACCOUNTS.ATTRIBUTE1 as the value in the **Implementation** field.

    c. Click OK to save the details.

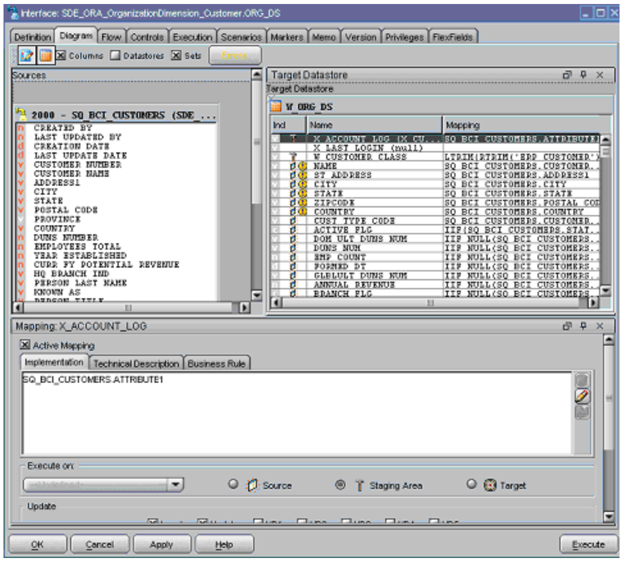
11. Repeat Step 10 for the Interface SDE_ORA_OrganizationDimension_Customer.SQ_BCI_CUSTOMERS_FULL.

12. Edit the Interface SDE_ORA_OrganizationDimension_Customer.ORG_DS to display the Interface: <*Name*> dialog, and do the following:

    a. Display the Diagram tab, and select the X_ACCOUNT_LOG field in the Target Datastore area.

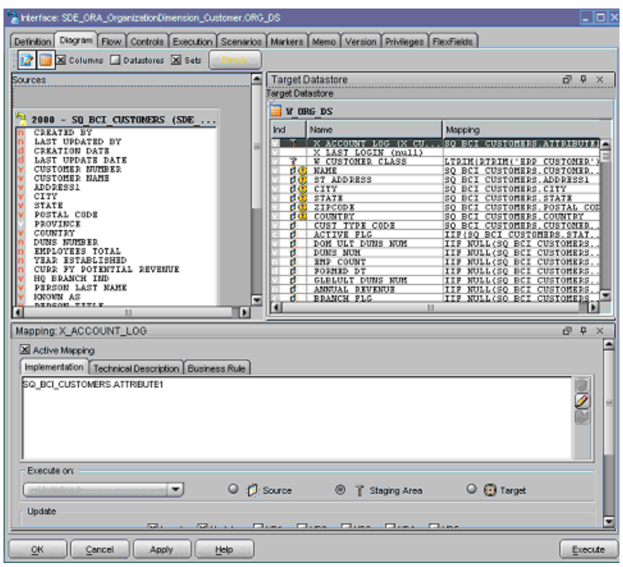


    b. Use the Launch Expression Editor icon to display the Expression Editor dialog, and use this dialog to select SQ_BCI_CUSTOMERS.ATTRIBUTE1 as the value in the **Implementation** field.



c. Select the X_LAST_LOGIN field in the Target Datastore area

d. Use the Launch Expression Editor icon to display the Expression Editor dialog, and use this dialog to select SQ_BCI_CUSTOMERS.LAST_UPDATE_LOGIN as the value in the **Implementation** field.



e. Click OK to save the details.

13. Repeat Step 12 for the Interface SDE_ORA_OrganizationDimension_Customer.ORG_DS_FULL.

14. Regenerate scenario SDE_ORA_OrganizationDimension_Customer (that is, right click on the scenario and select Regenerate).

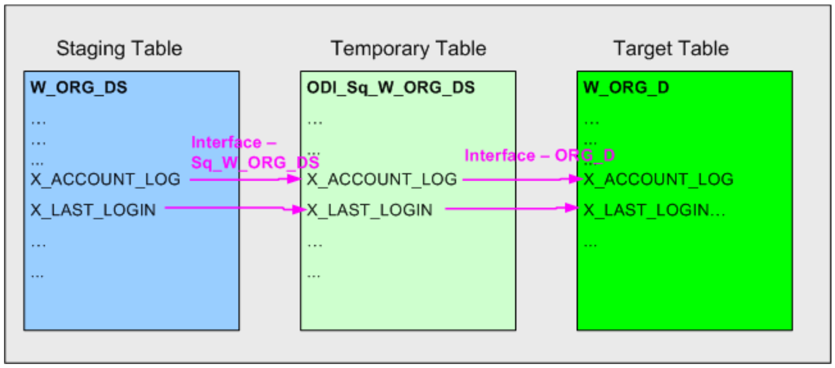
Now that you have set up the E-LT process for extracting and staging the data, you need to load the new data into the data warehouse (for more information, see Section 11.2.3.3, "Example of Loading Data from the Staging Area into an Existing Target Table"),

11.2.3.3 Example of Loading Data from the Staging Area into an Existing Target Table

This section shows how data is loaded from the staging area into an existing target table.

The diagram below shows the new E-LT mappings that you need to load the new data from the staging area into the target table, and the ODI Interfaces that you need to modify.

Figure 11–5 Required new mappings for loading data into the target table



Note: The diagram above only shows the incremental interfaces.

To customize the E-LT process to load these two fields into the staging area, you need to:

- Load the X_ACCOUNT_LOG value and X_LAST_LOGIN value from the staging table W_ORG_DS into the temporary table SQ_W_ORG_DS using the Interfaces Sq_W_ORG_DS and Sq_W_ORG_DS_FULL.
- Load the X_ACCOUNT_LOG value and X_LAST_LOGIN value from the temporary table ODI_Sq_W_ORG_DS into the Target table W_ORG_D using the Interfaces ORG_D, ORG_D_FULL, and ORG_D_UNSPC.

To extract data from an Oracle EBS 11.5.10 Data Packaged Source:

1. In ODI Designer, display the Projects view, expand the 'Oracle BI Applications 7.9.5.2'\Mappings\SILOS folder.
2. Right-click on the SIL_OrganizationDimension folder, and choose Version, then Create to display the Create: <*Object*> dialog, and specify a unique version number and optional version description.
3. Display the Models view, expand the Dimension Stage folder, and edit the W_ORG_DS data store to display the DataStore: <*Name*> dialog, and display the Columns tab.
4. Make sure that the following columns are setup:
   - X_ACCOUNT_LOG (VARCHAR2(10))
   - X_LAST_LOGIN (VARCHAR2(10))
5. Repeat steps 3 and 4 for the W_ORG_D data store.
6. In the Models view, right click on the model 'Oracle BI Applications 7.9.5.2' and select Generate DDL to display the Generate DDL dialog.

   The Generate DDL option deploys the changes in the database.

7. Select the check-box in the Synchronize column next for the W_ORG_DS table.
8. Click the (...) button to the right of the **Generation Folder** field to display the **Select a folder** dialog, and select the \Utilities\System folder, and click OK.
9. When the Procedure: DDL <*Name*> dialog is displayed, click Execute.

   Display ODI Operator and make sure that the procedure executes successfully.
10. Display the Projects view, expand the Mappings folder, and expand the SIL_OrganizationDimension folder.
11. Edit the Interface Sq_W_ORG_DS to display the Interface: <*Name*> dialog, and do the following:
    a. Display the Diagram tab, and select the X_ACCOUNT_LOG field in the Target Datastore area.
    b. Use the Launch Expression Editor icon to display the Expression Editor dialog, and use this dialog to select W_ORG_DS.X_ACCOUNT_LOG as the value in the **Implementation** field.
    c. Select the X_LAST_LOGIN field in the Target Datastore area
    d. Use the Launch Expression Editor icon to display the Expression Editor dialog, and use this dialog to select W_ORG_DS.X_LAST_LOGIN as the value in the **Implementation** field.
    e. Click OK to save the details.
12. Repeat Step 11 for the Interface Sq_W_ORG_DS_FULL.
13. Edit the Interface ORG_D to display the Interface: <*Name*> dialog, and do the following:
    a. Display the Diagram tab, and select the X_ACCOUNT_LOG field in the Target Datastore area.
    b. Use the Launch Expression Editor icon to display the Expression Editor dialog, and use this dialog to select Sq_W_ORG_DS.X_ACCOUNT_LOG as the value in the **Implementation** field.
    c. Select the X_LAST_LOGIN field in the Target Datastore area
    d. Use the Launch Expression Editor icon to display the Expression Editor dialog, and use this dialog to select Sq_W_ORG_DS.X_LAST_LOGIN as the value in the **Implementation** field.
    e. Click OK to save the details.
14. Repeat Step 13 for the Interface ORG_D_FULL and ORG_D_UNSPC.
15. Regenerate scenario SILOS\SIL_OrganizationDimension (that is, right click on the scenario and select Regenerate).

11.2.3.4 Tips for Modifying the SQ_BCI_ Interface

- A new source table should always be defined on right side of a LEFT OUTER join syntax with existing source tables. Using an INNER join or a RIGHT OUTER join can result in loss of records.
- Make sure that you define joins to match on a unique set of values. If you do not define a join that ensures a unique relationship, you might get a Cartesian product, which changes the granularity and results in duplicate errors. If it is not possible to define a unique join, then do the following:

1. Create an inline view interface sourcing from the new table, outputting necessary columns plus a column LKP_ACTIVE.

   For example, you might specify the expression for LKP_ACTIVE as:

   ```
   IS_FIRST(
   ARG_GROUP(columns to be partitioned by),
   ARG_GROUP(columns to be ordered by))
   ```

   Note: In the above example, the IS_FIRST command and the matching filter are only needed if multiple records might be returned.

2. Bring the inline view interface into an existing interface with a filter LKP_ACTIVE=1, which guarantees that at most one record will be returned.

   As a best practice, you should comment custom code that you introduce. Comments should include the developer's name and the date that the code was added.

11.2.3.5 Including a Source Table for the Change Capture Process

If you are bringing in data from a new table that was not previously included in an existing SDE package, you might need to create an auxiliary change capture mapping. When a row changes in the new table, the auxiliary change capture mapping marks the corresponding row in the main table as changed. Auxiliary change capture processes can degrade ELT performance. Therefore, auxiliary change capture processes should only be implemented if required.

11.2.4 Other Types of Customizations Requiring Special Handling

This section contains the following topics:

- Section 11.2.4.1, "How to Modify Category 2 SCD Triggers"
- Section 11.2.4.2, "How to Add A Dimension to an Existing Fact"
- Section 11.2.4.3, "How to Add a Date Dimension to an Existing Fact"

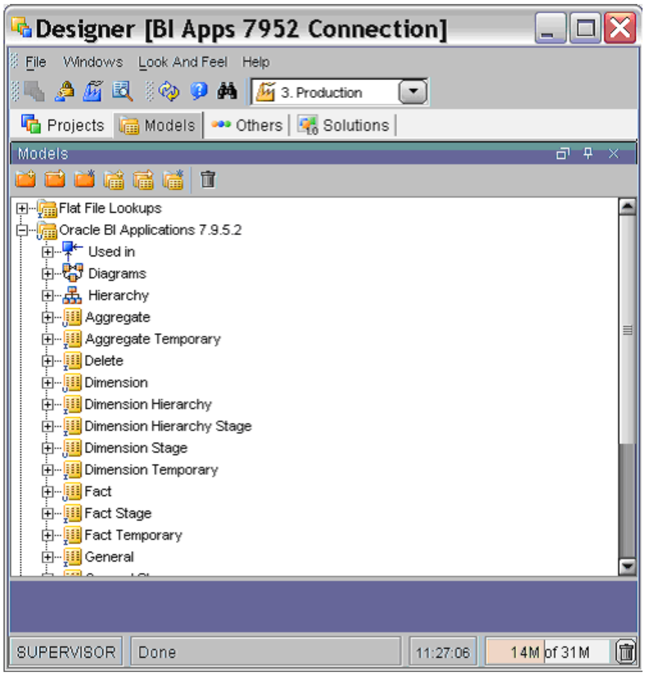
11.2.4.1 How to Modify Category 2 SCD Triggers

If a dimension is enabled to capture Type-II Change, you can modify the criteria that trigger a Type-II change in a dimension. Most changes in a dimension are treated as Type-I changes in that the existing column is simply overwritten with the new value. Once enabled, there are only a small number of columns that will trigger a Type-II change. You can extend the logic that triggers a Type-II change by adding additional columns to the logic that tracks Type-II changes. In addition, you can remove columns from this logic in case you do not want these types of changes to trigger a Type-II change. The Logic that tracks Type-II changes is contained in the data model.

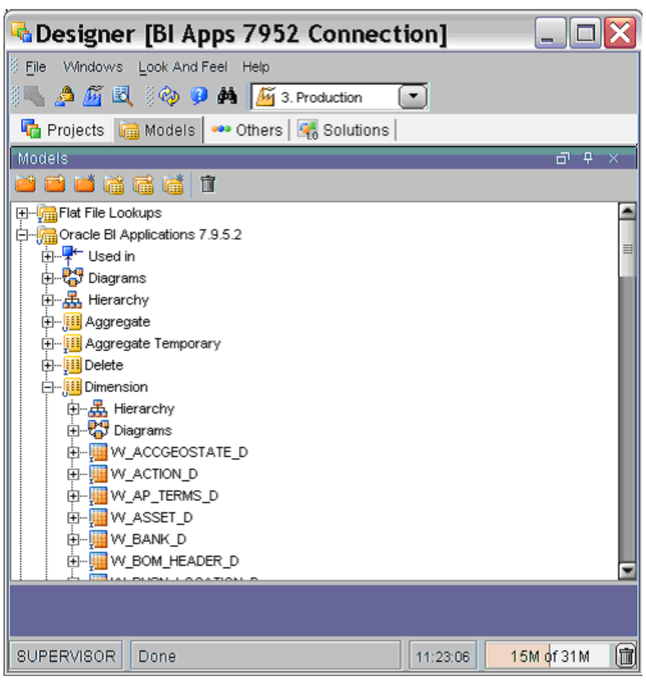
Note: Modifying the Type-II tracking logic is the only change that you should make to shipped logic.

To modify a Category 2 SCD Trigger:

1. In ODI Designer, display the Models view, and expand the 'Oracle BI Applications 7.9.5.2' folder.

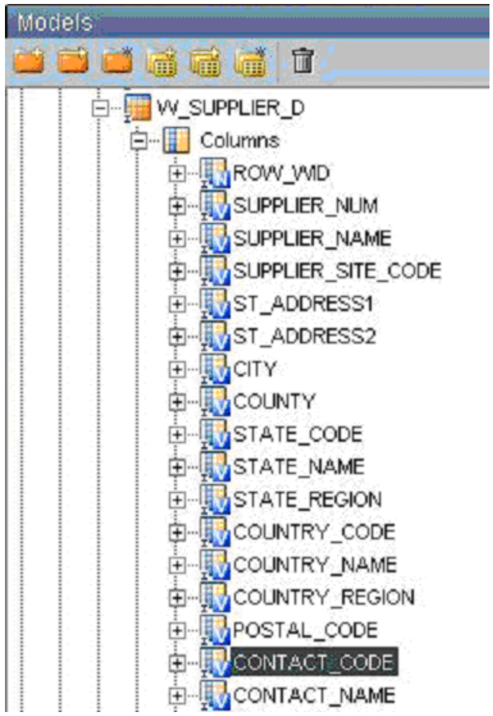


2. Expand the Dimension node.

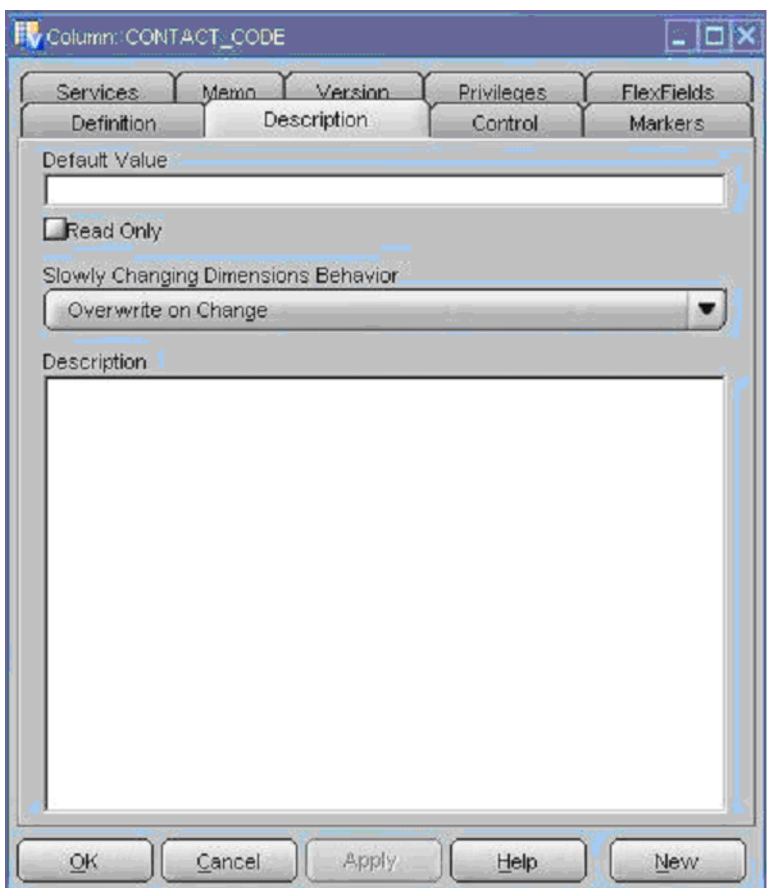


3. Select the Dimension and column on which you want to implement the Category 2 trigger.

   For example, you might select the W_SUPPLIER_D dimension and the CONTACT_CODE column.



4. Double click the column name to display the Column: <*Name*> dialog.
5. Display the Description tab.

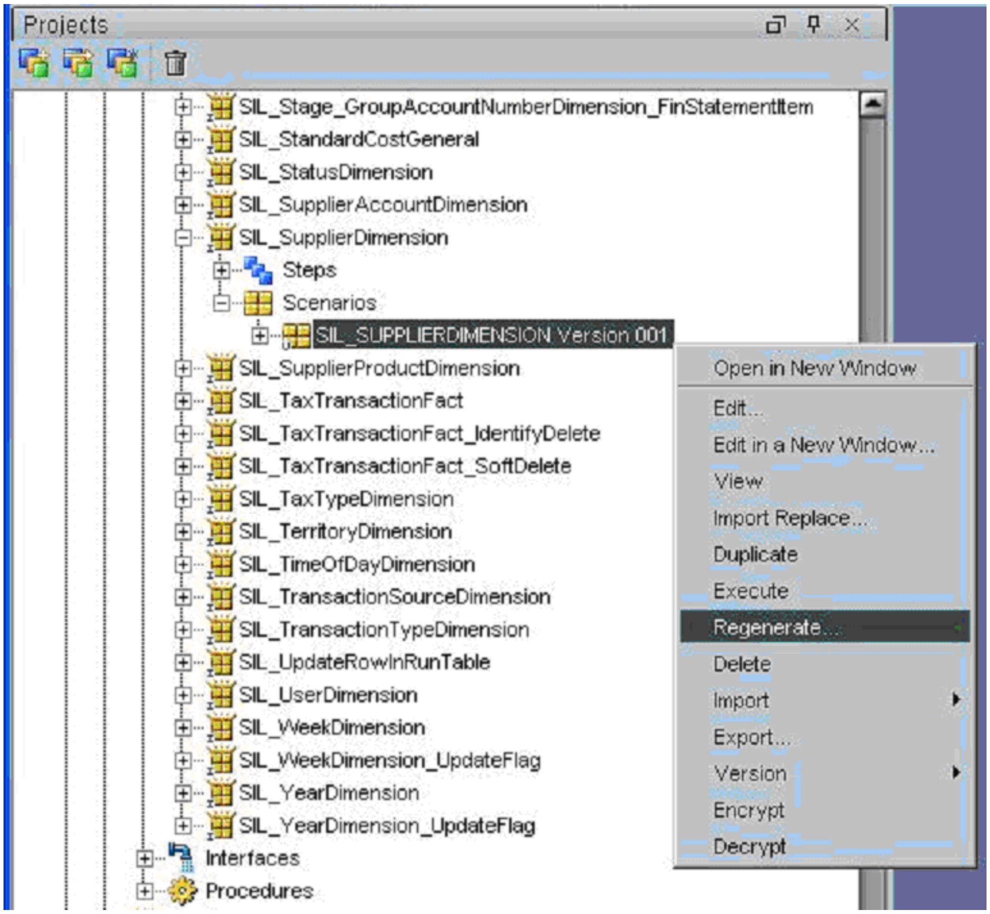


6. Use the **Slowly Changing Dimensions Behavior** drop down list to specify the behavior that you want, as follows:

   - If you want to trigger a Category 2 change, select 'Add Row on Change'.
   - If you want to trigger a Category 1 change, select 'Overwrite on Change'.

7. Re-generate the Scenario for the SIL Dimension mapping to reflect the data model change.

   This is, right-click the SIL Dimension, then select Regenerate.



For more information about customizing Slowly Changing Dimensions, see Section 11.4.4, "Configuring Slowly Changing Dimensions".

11.2.4.2 How to Add A Dimension to an Existing Fact

This section explains how to add a dimension (pre-existing or custom) to an existing fact. It assumes that you have already built the required process to populate this dimension.

1. In ODI Designer, define the staging table column as a varchar2(80) field and named with in _ID suffix.
2. Define the Oracle Business Analytics Warehouse table column as an integer and named with a _WID suffix.
3. Modify the SDE fact package to pass through the unique identifier of the dimension key.

There must be a relationship between the base table and this unique identifier. It might already be stored in the base table or stored by joining to a related table. This identifier can be based on a single column or derived from multiple columns.

The table below depicts various formats used to derive the INTEGRATION_ID, which is used to identify a dimension key. The INTEGRATION_ID value should be passed to the fact staging table.

Table 11–1 Formats to Derive INTEGRATION_ID

Dimension	Foreign Key	When Source is Oracle Application
W_AP_TERMS_D		TO_CHAR(TERM_ID)
W_BUSN_LOCATION_D	ASSET_LOC_WID	ASSET_LOC~' || LOCATION_ID
W_BUSN_LOCATION_D	EMP_LOC_WID	EMP_LOC~' || LOCATION_ID
W_BUSN_LOCATION_D	INVENTORY_LOC_WID	STORAGE_LOC' || '~' || ORGANIZATION_ID || '~' || SUBINVENTORY_CODE || '~' || INVENTORY_LOCATION_ID
W_BUSN_LOCATION_D	PLANT_LOC_WID	'PLANT' || '~' || TO_CHAR(ORGANIZATION_ID)
W_BUSN_LOCATION_D	RECEIVING_LOC_WID	'RECIPIENT_LOC' || '~' || TO_CHAR(LOCATION_ID)
W_BUSN_LOCATION_D	STORAGE_LOC_WID	'STORAGE_LOC' || '~' || ORGANIZATION_ID || '~' || SECONDARY_INVENTORY_NAME || '~'
W_CUSTOMER_FIN_PROFL_D	CUSTOMER_FIN_PROFL_WID	P'||'~'||TO_CHAR(CUSTOMER_ID) ||'~' || TO_CHAR(SITE_USE_ID) ||'~' || CURRENCY_CODE - CUSTOMER_ID is CUST_ACCOUNT_ID from HZ_CUST_ACCOUNTS and CURRENCY_CODE is from HZ_CUST_PROF_CLASS_AMTS
W_CUSTOMER_LOC_D		To get the customer location key, look up W_CUSTOMER_LOC_USE_D as below
W_CUSTOMER_LOC_USE_D		TO_CHAR(SITE_USE_ID) - Get Site Use Id from HZ_CUST_ACCOUNT_ROLES
W_FREIGHT_TERMS_D		LOOKUP_CODE
W_GL_ACCOUNT_D		to_char(ccid)
W_INT_ORG_D	COMPANY_ORG_KEY	COMPANY'||'~'||TO_CHAR(SET_OF_BOOKS_ID)
W_INT_ORG_D	*_ORG_KEY	Remove any prefixes and use TO_CHAR()
W_ORG_D	CUSTOMER_WID	TO_CHAR(CUSTOMER_ID) - CUSTOMER_ID is CUST_ACCOUNT_ID from HZ_CUST_ACCOUNTS
W_PAYMENT_METHOD_D		LOOKUP_CODE
W_PAYMENT_METHOD_D		TO_CHAR(TERM_ID)
W_PERSON_D	CUST_CONTCT_WID	TO_CHAR(PARTY_ID) - PARTY_ID from HZ_PARTY_RELATIONS
W_PRODUCT_D	PRODUCT_WID	TO_CHAR(INVENTORY_ITEM_ID)

Table 11–1 (Cont.) Formats to Derive INTEGRATION_ID

Dimension	Foreign Key	When Source is Oracle Application
W_SALES_PRODUCT_D		TO_CHAR(INVENTORY_ITEM_ID) ||'~'||TO_CHAR(ORGANIZATION_ID)

4. To resolve the value of the new WID column in the SIL package, do the following:
   a. Create an Inline View Interface X_LKP_W_Dim_D for the dimension table.
   b. In the 'SQ_xxxxx' interfaces, use an outer join to join the Fact table to the Inline View of the dimension table using INTEGRATION_ID, DATASOURCE_NUM_ID.

      If the dimension is a slowly changing dimension, the fact table's standard or 'canonical' date should be used as well as the join condition, even if the dimension has not been enabled to capture Category 2 changes.
5. Use ROW_WID of the dimension table as the WID value of the fact table.

   Add logic to default the WID value to 0 if no record is returned from the join.
6. Save the changes to the SDE and SIL packages.
7. Make sure that the new dimension extract and load packages are already part of the Master packages, and that it is loaded before the Fact load.

11.2.4.3 How to Add a Date Dimension to an Existing Fact

If you want to add a date dimension to a fact table, you pass the date through the SIL Interfaces to the target table.

To add a date dimension to an existing fact:

1. Add a new field called DT_WID to the fact table, as follows:
   a. In ODI Designer, display the Models view, and open the Oracle BI Applications 7.9.5.2 folder.
   b. Expand the Fact folder, and double-click the fact table to display the DataStore:<*Name*> dialog.

      For example, you might edit the W_GL_REVN_F table.
   c. Click the Add Column icon to add a new column to the list, and change the default column name to DT_WID, and change the Type to DATE.
   d. In the Models view, right click on the model 'Oracle BI Applications 7.9.5.2' and select Generate DDL to display the Generate DDL dialog.

      The Generate DDL option deploys the changes in the database.
   e. Select the check-box in the Synchronize column next for the table that you added.
   f. Click the (...) button to the right of the **Generation Folder** field to display the **Select a folder** dialog, and select the \Utilities\System folder, and click OK.
   g. When the Procedure: DDL <*Name*> dialog is displayed, click Execute.

      Display ODI Operator and make sure that the procedure executes successfully.
2. Modify the new DT_WID column in the SIL Interface, as follows:
   a. In ODI Designer, display the Projects view, and open the SIL package.

For example, you added a new date field to the W_GL_REVN_F table, you might want to edit the SIL_GLRevenueFact\SIL_GLRevenueFact Package.

b. Edit the last Interface located at the end of the flow.

For example, if you edit the SIL_GLRevenueFact\SIL_GLRevenueFact Package, you edit the 'Run GL_REVN_F_FULL' Interface.

c. Display the Diagram tab.

d. In the Target Datastore area, select the DT_WID column that you created in step 1.

e. In the Mapping pane below, add the following expression in the **Implementation** field:

```
COALESCE(IIF(
ISNULL(Sq_Fact_Table.Newly_added_DT),
NULL,
TO_INTEGER(TO_CHAR_FORMAT(Sq_Fact_Table.Newly_added_DT,'YYYYMMDD'))),0)
```

f. Save the details.

g. Regenerate the interface.

11.3 Category 2 Customizations: Adding Additional Tables

Category 2 customizations use pre-packaged adapters to add new fact or dimension tables to the Oracle Business Analytics Warehouse.

This section contains the following topics:

- Section 11.3.1, "About Creating New Dimension or Fact Tables"
- Section 11.3.3, "Creating Custom ODI Master Packages"
- Section 11.3.4, "Adding a New Dimension to the Oracle Business Analytics Warehouse"
- Section 11.3.5, "Adding a New Fact Table to the Oracle Business Analytics Warehouse"
- Section 11.3.6, "Adding a New Dimension Table for a New Fact Table in the Oracle Business Analytics Warehouse"

11.3.1 About Creating New Dimension or Fact Tables

This section relates to building entirely new tables that will be loaded with data from a source table that is not already extracted from. For example, you might want to create a new Project dimension table. In this case, you create new dimension and staging tables as well as new extract and load ELT mappings.

When creating a new custom table, use the prefix WC_ to help distinguish custom tables from tables provided by Oracle as well as to avoid naming conflicts in case Oracle later releases a table with a similar name. For example, for your Project dimension you might create a WC_PROJECT_DS and a WC_PROJECT_D table.

When you create a new dimension or fact table, use the required system columns that are part of each of the Oracle Business Analytics Warehouse tables to maintain consistency and enable you to reference existing table structures. When you create a new table, you need to define the table and indices in ODI Designer Models area first. The destination model for the Oracle Business Analytics Warehouse is 'Oracle BI Applications 7.9.5.2'.

11.3.1.1 About the Main Required Columns

For custom staging tables, the following columns are required:

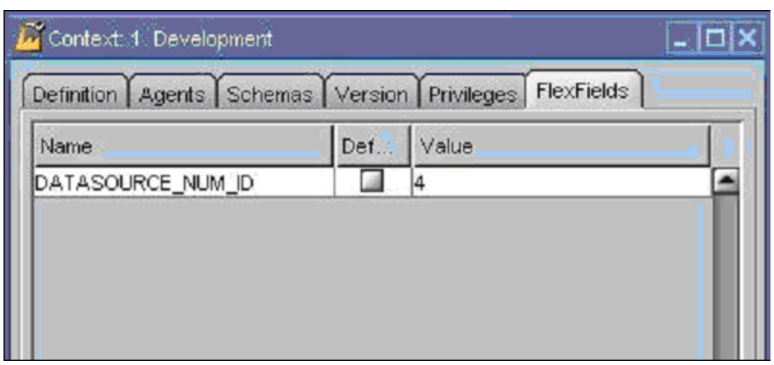
- **INTEGRATION_ID.** Stores the primary key or the unique identifier of a record as in the source table.
- **DATASOURCE_NUM_ID.** Stores the data source from which the data is extracted.

For dimension and fact tables, the required columns are the INTEGRATION_ID and DATASOURCE_NUM_ID columns as well as the following:

- **ROW_WID.** A sequence number generated during the ELT process, which is used as a unique identifier for the Oracle Business Analytics Warehouse.
- **ELT_PROC_WID.** Stores the ID of the ELT process information. The details of the ELT process are stored in the W_ELT_RUN_S table on the Oracle Business Analytics Warehouse side.

11.3.2 About the DATASOURCE_NUM_ID Column

The tables in the Oracle Business Analytics Warehouse schema have DATASOURCE_NUM_ID as part of their unique user key. While the transactional application normally ensures that a primary key is unique, it is possible that a primary key is duplicated between transactional systems. To avoid problems when loading this data into the data warehouse, uniqueness is ensured by including the DATASOURCE_NUM_ID as part of the user key. This means that the rows can be loaded in the same data warehouse tables from different sources if this column is given a different value for each data source. The DATASOURCE_NUM_ID is maintained by a Flex Field value that you can specify using ODI Topology Manager. The out-of-the-box Flex Field value is '1', but you change this to '4' (for Oracle EBS 11.5.10 applications) as part of the installation and setup process (for more information, see Section 4.5.5.3, "How to set up the Data Source Number").



11.3.3 Creating Custom ODI Master Packages

Creating a custom ODI Master Package enables you to create a custom Subject Area.

Note: If you customize an existing package or interface, then you do not need to modify the master packages since the modified package is already being executed.

If you create a new customized package, you need to include the customized package in the master packages to be executed during a regular load. To include a package in the master package, it should be called in either a level 3 subject area package or a level 4 task group package. To create one of these level 3 or level 4 packages, you can use of the package templates:

- 3_Master_PLP_<App>_<Subj>
- 3_Master_SDE_Facts_<App>_<Subj>
- 3_Master_SIL_Facts_<App>_<Subj>
- 4_Master_PLP_TG_<Table>
- 4_Master_SDE_Dimensions_TG_<Dim>
- 4_Master_SIL_Dimensions_TG_<Dim>
- 4_Master_SDE_Facts_TG_<Fact>
- 4_Master_SIL_Facts_TG_<Fact>

11.3.3.1 Notes on Using the Package Templates

- You need to make a copy of the appropriate template that you wish to use for your customized package. If your customized package is for a dimension table and is for an existing category, you just need to append it to their respective level 3 SDE/SIL dimension category package.
- If your customized package is for a dimension table and needs a new category, create 3_Master_SDE_Dimensions_Custom and 3_Master_SIL_Dimensions_Custom. There is no template package in the master package folder for these packages. This is because there are no other steps in these packages aside from the package that you need to execute. Use the 'Insert package' option in ODI for creating these packages. Put the step before the 'wait for child session' step in these packages.
- If the customized package is a new PLP dimensions package, you need to append this in 3_Master_PLP_Dimensions package. There is no need to create a new level 3 PLP dimension master package.
- If your customized package is for a Fact or PLP table and is for an existing subject area, you just need to append this package to their appropriate level 3 packages.
- If your customized package is for a Fact or PLP table and is for a new subject area, use the other appropriate level 3 templates. There are steps in this template for refresh and evaluate of the variable OBI_EXEC_PACKAGE. You need to append the steps you need to execute after the evaluate variable step. Use the ‘ok’ connection from the ‘Execute Package?’ step to your customized step. Change <App> with CUSTOM and <Subj> with your own customized subject area name. For new subject areas, you need to use Oracle BI Applications Configuration Manager to add the subject area and enable it for execution in its appropriate subject area. Before you can use Oracle BI Applications Configuration Manager to add a new subject area, use the 'Add Custom Subject Area' and 'Refresh Package Module Data' packages located in the Projects folder in ODI Designer (in \Utilities_and_Execution\Utilities\User\Packages\).
- After creating the level 3 package, you need to generate the scenario for this package. Add a scenario execution step in their appropriate level 2 master package, as follows.
  - After creating the level 3 package, you need to generate the scenario for this package. Create the following level 2 master packages to execute these customized packages: 2_Master_PLP_CUSTOM, 2_Master_SDE_Facts_CUSTOM, and 2_Master_SIL_Facts_CUSTOM.

    Add a scenario execution step in their appropriate customized level 2 master packages. Set the agent code in these steps to "WORKFLOW". Set the appropriate "Synchronous / Asynchronous" setting for the steps. Add an

"OdiWaitForChildSession" step if necessary. Generate the scenarios for these customized level 2 packages, add a scenario step to: 1_Master_PLP, 1_Master_SDE_Facts, and 1_Master_SIL_Facts.

  - For level 4 task group packages, this is required if there are multiple customized package loading the same target table or you are using temporary table.

- If your customized package is for an existing task group, just append it to their appropriate level 4 task group package.
- If your customized package is for a dimension table and require a new task group package, use either 4_Master_SDE_Dimensions_TG_<Dim> or 4_Master_SIL_Dimensions_TG_<Dim> template.
- In this template, there are steps for refresh and evaluate of OBI_START_TASK_GROUP, a refresh of IS_INCREMENTAL variable and an Update Load Date procedure call. These steps are mandatory for this type of task group. The other steps are Drop Indexes, Truncate Table, and Create Indexes. These steps are optional if you want such process to be handled in the task group level. The parameters for these steps should be set as follows:
  - For drop index step, set the value in additional variable tab for OBI_MANAGE_TABLE_MASK and OBI_INDEX_DROP_ELT.
  - For truncate table step, set the value in additional variable tab for OBI_MANAGE_TABLE_MASK, OBI_TRUNCATE_TABLE, and OBI_TRUNCATE_TASK_GROUP.
  - For create index step, set the value in additional variable tab for OBI_MANAGE_TABLE_MASK and OBI_INDEX_CREATE_ELT.
- Your customized package should be added after the truncate table step and before create indexes step. Use an OdiWaitForChildSession step if necessary for asynchronous jobs.
- If your customized package is for a Fact or PLP table and require a new task group package, then use 4_Master_SDE_Facts_TG_<Fact>, 4_Master_SIL_Facts_<Fact>, or 4_Master_PLP_TG_<Table> template.
- The steps in these templates are similar to those in the dimensions task group. The only difference is it has steps for refresh and evaluate of OBI_START_PACKAGE instead of OBI_START_TASK_GROUP variable. Set the other steps the same way as the dimensions task group. These customized level 4 task group packages should be added to their appropriate level 3 master packages.
- **Note**: For all packages modified, you need to regenerate the scenario.

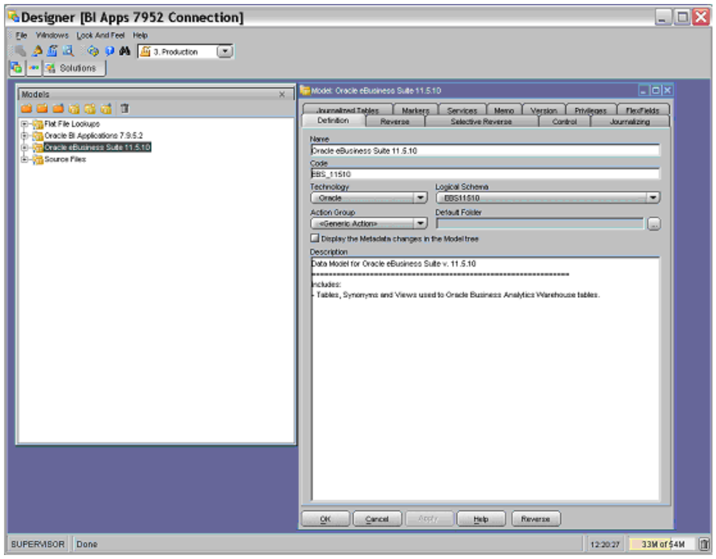
11.3.3.2 Additional Information About Customizing

This section contains additional miscellaneous information about customization in Oracle Business Intelligence Applications.

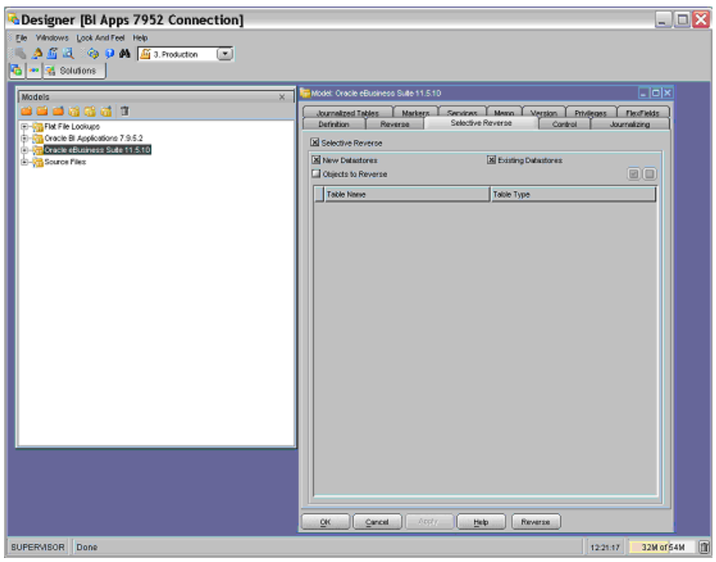
11.3.3.2.1 About Table Definitions in ODI When you import table definitions from external data sources to an ODI work repository, make sure that the 'Technology' is set to Oracle.

To import table definitions into an ODI Work Repository:

1. In ODI Designer, display the Models view, and double-click the folder that holds the table definitions (for example, 'Oracle eBusiness Suite 11.5.10') to display the Model: <**Name**> dialog.



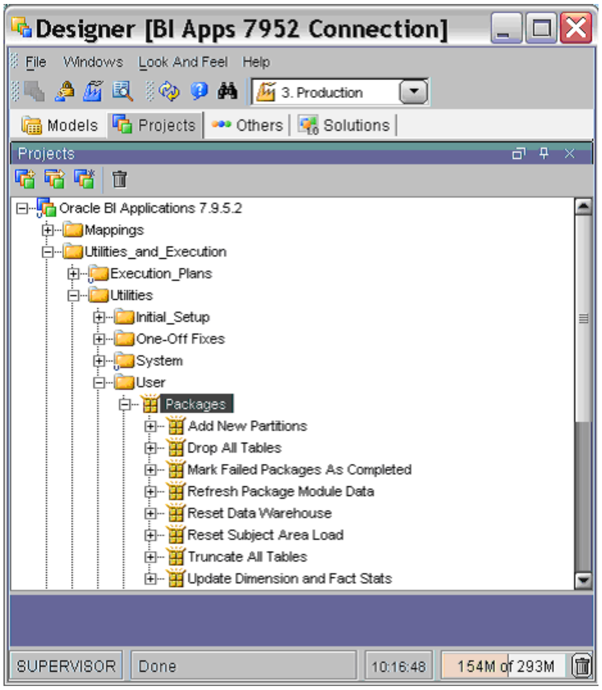
2. Display the Selective Reverse tab.



3. Select the **Objects To Reverse** check box.
4. Select the objects (Table/View/Synonym) that you want to import into the ODI Repository.
5. Click OK to reverse-engineer the objects.

11.3.3.2.2 About the Update Strategy For loading new fact and dimension tables, design a custom process on the source side to detect the new and modified records. The SDE process should be designed to load only the changed data (new and modified). If the data is loaded without the incremental process, the data that was previously loaded will be erroneously updated again. For example, the logic in the preconfigured SIL mappings looks up the destination tables based on the INTEGRATION_ID and DATASOURCE_NUM_ID and returns the ROW_WID if the combination exists, in which case it updates the record. If the lookup returns null, it inserts the record instead. In some cases, last update date(s) stored in target tables are also compared in addition to the columns specified above to determine insert or update. Look at the similar mappings in the preconfigured folder for more details.

11.3.3.2.3 About Truncating Target Tables Truncation of Tables should be done using the 'Truncate All Tables' package, which is located in the Projects\Oracle BI Applications 7.9.5.2\Utilities_and_Execution\Utilities\User\Packages folder. This utility truncates all the tables in the data warehouse prior to a forced full load.



11.3.3.2.4 About the ETL_PROC_WID Setting Use the ETL_PROC_WID setting in the W_PARAM_G table in custom mappings. ETL_PROC_WID is a reference key to the Run History of the ODI Master Package.

11.3.3.2.5 About Indices and Naming Conventions Staging tables typically do not require any indices. Use care to determine if indices are required on staging tables. Create indices on all the columns that the E-LT will use for dimensions and facts (for example, ROW_WIDs of Dimensions and Facts, INTEGRATION_ID and DATASOURCE_NUM_ID and flags). Carefully consider which columns or combination of columns filter conditions should exist, and define indices to improve query performance. Inspect the OTB objects for guidance. Name all the newly created tables as WC_. This helps visually isolate the new tables from the out-of-the-box tables. Keep good documentation of the customizations done; this helps when upgrading your data warehouse. Once the indices are decided upon, they should be registered in the ODI Model (for more information, see Section 11.5.3, "How to add an index to an existing fact or dimension table").

11.3.4 Adding a New Dimension to the Oracle Business Analytics Warehouse

Follow this procedure to add a new dimension table to the Oracle Business Analytics Warehouse.

To add a new dimension and use it with an existing fact table:

1. In ODI Designer, log in as SUPERVISOR, and display the Models view.
2. In the Oracle BI Applications 7.9.5.2 folder, create a new dimension table structure (with appropriate system columns),

   Use the naming convention W_<*Dimension Name*>_D.

3. In the Oracle BI Applications 7.9.5.2 folder, create a new staging table structure (with appropriate system columns), using the naming convention W_<*Dimensions Name*>_DS.
4. In the definition of the fact tables related to this new dimension, insert a reference constraint to this dimension table.

   The dimension table must be the parent table in the reference constraint definition.
5. Display the Projects view, and select the Oracle BI Applications 7.9.5.2 folder.
6. Create a new interface called SDE_<*XYZ*>.<*Interface name*>(_Full) to populate the dimension stage.

   Create a custom inline view interface if necessary. This inline view interface will be the source of the main interface for loading the dimension stage. Assign the appropriate knowledge module for these interfaces. Set the appropriate value for the options in the assigned knowledge module. If required, create separate interfaces for full load and incremental load. Refer to existing interfaces as examples.
7. Create a new package called SDE_<*Package_name*> to contain the interfaces for loading the dimension stage.

   Create the steps for refresh and evaluate of variables OBI_START_PACKAGE and IS_INCREMENTAL (if full and incremental is necessary). The 'Refresh OBI_START_PACKAGE' step is always the first step. Add other variables or objects as required by the package. Invoke the interfaces created in the previous step in this package. A package can have two branches, for full or incremental load. Refer to existing SDE packages as a base sample.
8. Create new custom interfaces and packages for the SILO jobs for loading the dimension table from dimension stage.

   To set truncate options for the interfaces, assign a value to the KM option OBI_TRUNCATE_TABLE. A value of F is for truncate on full load, while a value of Y is for truncate always. For SDE it is usually truncate always, while for SIL it is truncate on full load.
9. Insert the new packages into the master package.

11.3.5 Adding a New Fact Table to the Oracle Business Analytics Warehouse

Follow this procedure to add a new fact table to the Oracle Business Analytics Warehouse.

To add a new fact table:

1. In ODI Designer, log in as SUPERVISOR, and display the Models view.
2. In the Oracle BI Applications 7.9.5.2 folder, create a new fact table structure (with appropriate system columns), using the naming convention W_<*Dimension Name*>_F.

   In the constraints tree of this new fact table, insert a reference constraint to all dimension tables related to this fact. The dimension table has to be the parent table in the reference constraint definition. In the flexfields tab of this new fact table, change the value of the OBI_MODULE flexfield to the appropriate module of this new fact table such as OM,SCA,HR, FIN or CUSTOM. In addition, set the value of the OBI_SUBJECT_AREA flexfield to your customized subject area name.

3. In the Oracle BI Applications 7.9.5.2 folder, create a new fact staging table structure (with appropriate system columns), using the naming convention W_<*Dimensions Name*>_FS.
4. Display the Projects view, and select the Oracle BI Applications 7.9.5.2 folder.
5. Create a new interface called SDE_<*XYZ*>.<*Interface name*>(_Full) to populate the fact stage.

   Create a custom inline view interface if necessary. This inline view interface will be the source of the main interface for loading the dimension stage. Assign the appropriate knowledge module for these interfaces. Set the appropriate value for the options in the assigned knowledge module. If required, create separate interfaces for full load and incremental load. Refer to existing interfaces as examples.
6. Create a new package called SDE_<*Package_name*> to contain the interfaces for loading the fact stage.

   Create the steps for refresh and evaluate of variables OBI_START_PACKAGE and IS_INCREMENTAL (if full and incremental is necessary). The 'Refresh OBI_START_PACKAGE' step is always the first step. Add other variables or objects as required by the package. Invoke the interfaces created in the previous step in this package. A package can have two branches, for full or incremental load. Refer to existing SDE packages as a base sample.
7. Create new custom interfaces and packages for the SILO jobs for loading the fact table from the fact stage.

   To set truncate options for the interfaces, assign a value to the KM option OBI_TRUNCATE_TABLE. A value of F is for truncate on full load, while a value of Y is for truncate always. For SDE it is usually truncate always, while for SIL it is truncate on full load.
8. Insert the new packages into the master package.

11.3.6 Adding a New Dimension Table for a New Fact Table in the Oracle Business Analytics Warehouse

The steps for creating a new dimension table are similar to the steps for incremental change capture.

To add a new dimension table for a new fact table:

1. In the new custom fact loading interface, drag and drop the new dimension table into the source pane of the interface.
2. Create a join between the dimension table and the fact staging table.
3. Extract the ROW_WID column from the dimension table and assign it to the corresponding column in the target fact table.

11.4 Category 3 Customizations: Adding New Data as a Whole Row into a Standard Dimension Table

Category 3 customizations use the Universal adapter to load data from sources that do not have pre-packaged adapters.

This section contains the following topics:

11.4.1 How to Add New Data as a Whole Row Into a Standard Dimension Table

Follow this procedure to add new data as a whole row into a standard dimension table in the Oracle Business Analytics Warehouse.

To add new data as a whole row into the standard dimension table:

1. Identify and understand the existing structure of staging tables. Refer to *Oracle Business Analytics Warehouse Data Model Reference* for the table structures. Non-system columns can include the null value.

2. Create a custom SDE interface to load the data into the staging table in the custom folder for this purpose. The staging table needs to be populated with incremental data (rows that have been added or changed since the last Refresh ELT process), for performance reasons.

3. Populate the INTEGRATION_ID column with the unique identifier for the record.

   The combination of INTEGRATION_ID and DATASOURCE_NUM_ID is unique. When importing the data, make sure that a unique identifier for the external data source is inserted in the DATASOURCE_NUM_ID column. Populate the INTEGRATION_ID column with the unique identifier for the record. The combination of INTEGRATION_ID and DATASOURCE_NUM_ID is unique. When importing the data, make sure that a unique identifier for the external data source is inserted in the DATASOURCE_NUM_ID column. The DATASOURCE_NUM_ID should be set to 4 for interfaces that source data from the Oracle Applications 11.5.10. This is a reserved value and is used in all standard interfaces.

   For example, a value of 2 can be defined for DATASOURCE_NUM_ID in the custom SDE interfaces. The standard SDE interfaces populate the INTEGRATION_ID column of the dimension staging table (used for resolving the dimension's Oracle Applications 11.5.10 ROW_ID value). The custom process must be used to populate the same column with a unique identifier from the external data source.

4. After the data is populated in the staging table, use the standard SIL interfaces to populate the dimension target tables.

5. Modify the SDE and SIL interfaces of all the related fact tables (fact tables that need to be linked to this dimension).

   The custom fact SDE interfaces must populate the foreign key column of the changed dimension (using a custom map table process to convert from Oracle EBS 11.5.10 ROW IDs to the external data source row IDs). The custom SIL interface should be modified to use the appropriate DATASOURCE_NUM_ID, because the standard SIL interfaces assume that the DATASOURCE_NUM_ID for the dimensions are the same as the fact table's DATASOURCE_NUM_ID.

   It is important to decide when the data is going to be loaded. If it is going to be loaded along with the Oracle source data, make sure that failure recovery is configured correctly. The preconfigured Master Packages truncate the target staging table prior to loading. Upon failure, when the Master Package restarts the task, the data is truncated and all the data is loaded again.

If the data from the external source gets loaded into the same staging table, be careful with how you handle this situation, since you cannot use the truncate table functionality. The data migrating into the staging table is not incrementally loaded, and, therefore, should be cleaned up prior to attempting to load this table again. In such a case, it is recommended that you encapsulate the extract part from both the sources inside an Master Package. Note that the data from both the sources should be run at the same time, all the time.

If it is decided that the data is going to be loaded at different time frequencies, then the new SDE packages need not depend on the preconfigured SDE packages and can use the Truncate Table option for failure recovery. Make sure the shared SIL process depends on the SDE processes from both sources.

11.4.2 Configuring Extracts

Each application has prepackaged logic to extract particular data from a particular source. This section discusses how to capture all data relevant to your reports and ad hoc queries by addressing what type of records you want and do not want to load into the Oracle Business Analytics Warehouse, and contains the following topics:

- Section 11.4.2.1, "Extracting Additional Data"
- Section 11.4.2.2, "Setting Up the Delimiter for a Source File"

11.4.2.1 Extracting Additional Data

You can configure extract mappings and Interfaces in the Oracle Business Analytics Warehouse to accommodate additional source data. For example, if your business divides customer information into separate tables based on region, then you would have to set up the extract interface to include data from these tables.

11.4.2.1.1 Extracting New Data Using an Existing Source Table Extract interfaces generally consist of source tables, expressions used in the target columns, and a staging table. If you want to extract new data using the existing interface, you have to modify the extract interface to include the new data by performing the following tasks:

To modify an existing interface to include new data:

1. Modify the existing interface to extract information from the source, and add it to an appropriate extension column.
2. Modify the Expressions in the target table to perform any necessary transformations.
3. Save the changes.
4. Regenerate the scenario.

You have to determine which type of extension column to map the data to in the staging table. After you modified the extract interface, you would also have to modify the corresponding load interfaces (SDE and SIL) to make sure that the extension columns that you added are connected all the way from the staging table to the target data warehouse table.

11.4.2.1.2 Extracting Data from a New Source Table Extract interfaces (which have the SQ_* naming convention) reside in source-specific folders within the repository. Extract interfaces are used to extract data from the source system. You can configure these extract interfaces to perform the following:

- Extract data from a new source table.

- Set incremental extraction logic.

11.4.2.2 Setting Up the Delimiter for a Source File

When you load data from a Comma Separated Values (CSV) formatted source file, if the data contains a comma character (,), you must enclose the source data with a suitable enclosing character known as a delimiter that does not exist in the source data.

Note: Alternatively, you could configure your data extraction program to enclose the data with a suitable enclosing character automatically.

For example, you might have a CSV source data file with the following data:

```
Months, Status
January, February, March, Active
April, May, June, Active
```

If you loaded this data without modification, ODI would load 'January' as the Months value, and 'February' as the Status value. The remaining data for the first record (that is, March, Active) would not be loaded.

To enable ODI to load this data correctly, you might enclose the data in the Months field within the double-quotation mark enclosing character (" ") as follows:

```
Months, Status
"January, February, March", Active
"April, May, June", Active
```

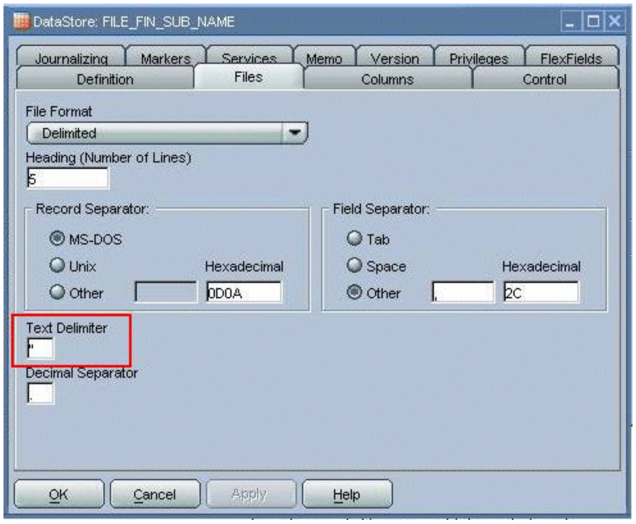
After modification, ODI would load the data correctly. In this example, for the first record ODI would load 'January, February, March' as the Months value, and 'Active' as the Status value.

To set up the delimiter for a source file:

1. Open the CSV file containing the source data.
2. Enclose the data fields with the enclosing character that you have chosen (for example, (").

   You must choose an enclosing character that is not present in the source data. Common enclosing characters include single quotation marks (') and double quotation marks (").
3. Save and close the CSV file.
4. In ODI Designer, display the Models view, and expand the 'Oracle BI Applications 7.9.5.2' folder.

   Identify the data stores that are associated with the modified CSV files. The CSV file that you modified might be associated with one or more data stores.
5. In ODI Designer, change the properties for each of these data stores to use the enclosing character, as follows:
   a. Double-click the data source, to display the DataStore: <*Name*> dialog.
   b. Display the Files tab.



c. Use the **Text Delimiter** field to specify the enclosing character that you used in step 2 to enclose the data.

d. Click OK to save the changes.

You can now load data from the modified CSV file.

11.4.3 Configuring Loads

This section explains how to customize the way that Oracle Business Intelligence Applications loads data into the Oracle Business Analytics Warehouse. For example, you might want to delete records from the Oracle Business Analytics Warehouse that have been deleted in the source system.

11.4.3.1 Filtering and Deleting Records

In a typical implementation, records that are deleted from your source system are not removed from the Oracle Business Analytics Warehouse. If you want to mark these records as deleted in the Oracle Business Analytics Warehouse, which were removed from the source system's database and archived in a separate database, you must enable the primary extract and delete mappings.

Primary extract mappings flag records that are deleted from the Oracle Business Analytics Warehouse. Delete mappings sets the DELETE_FLG column to 'Y' for these records in the warehouse tables. When enabled, primary extract and delete mappings by default look for any records removed from the source system's database. If these mappings find that the records no longer exist in that database, the mappings mark them as deleted in the data ware

11.4.3.2 About Primary Extract and Delete Mappings Process

Before you decide to enable primary extract and delete sessions, it is important to understand their function within the Oracle Business Analytics Warehouse. Primary extract and delete mappings allow your analytics system to determine which records are removed from the source system by comparing primary extract staging tables with the most current Oracle Business Analytics Warehouse table.

The primary extract mappings perform a full extract of the primary keys from the source system. Although many rows are generated from this extract, the data only extracts the Key ID and Source ID information from the source table. The primary extract mappings load these two columns into staging tables that are marked with a *_PE suffix.

The figure below provides an example of the beginning of the extract process. It shows the sequence of events over a two day period during which the information in the source table has changed. On day one, the data is extracted from a source table and loaded into the Oracle Business Analytics Warehouse table. On day two, Sales Order number three is deleted and a new sales order is received, creating a disparity between the Sales Order information in the two tables.

Figure 11–6 Extract and load mappings

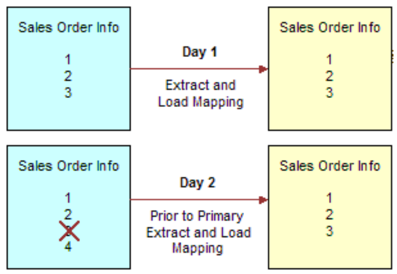
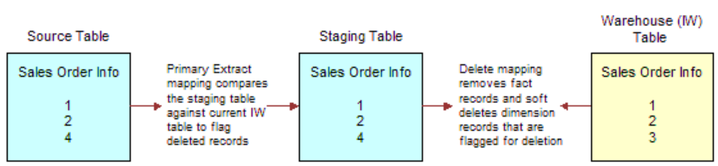


Figure 11–7 shows the primary extract and delete process that occurs when day two's information is extracted and loaded into the Oracle Business Analytics Warehouse from the source. The initial extract brings record four into the Oracle Business Analytics Warehouse. Then, using a primary extract mapping, the system extracts the Key IDs and the Source IDs from the source table and loads them into a primary extract staging table.

The extract mapping compares the keys in the primary extract staging table with the keys in the most current the Oracle Business Analytics Warehouse table. It looks for records that exist in the Oracle Business Analytics Warehouse but do not exist in the staging table (in the preceding example, record three), and sets the delete flag to Y in the Source Adapter mapplet, causing the corresponding record to be marked as deleted.

The extract mapping also looks for any new records that have been added to the source, and which do not already exist in the Oracle Business Analytics Warehouse; in this case, record four. Based on the information in the staging table, Sales Order number three is physically deleted from Oracle Business Analytics Warehouse, as shown in Figure 11–7. When the extract and load mappings run, the new sales order is added to the warehouse.

Figure 11–7 Primary Extract and Delete Mappings



11.4.3.3 About Working with Primary Extract and Delete Mappings

The primary extract (*_Primary) and delete mappings (*_IdentifyDelete and *_Softdelete) serve a critical role in identifying which records have been physically deleted from the source system. However, there are some instances when you can disable or remove the primary extract and delete mappings, such as when you want to retain records in the Oracle Business Analytics Warehouse that were removed from the source systems' database and archived in a separate database.

Because delete mappings use Source IDs and Key IDs to identify purged data, if you are using multiple source systems, you must modify the SQL Query statement to verify that the proper Source ID is used in the delete mapping. In addition to the primary extract and delete mappings, the configuration of the delete flag in the load mapping also determines how record deletion is handled.

You can manage the extraction and deletion of data in the following ways:

- Deleting the configuration for source-archived records
- Deleting records from a particular source
- Enabling delete and primary-extract sessions
- Configuring the Record Deletion flag
- Configuring the Record Reject flag

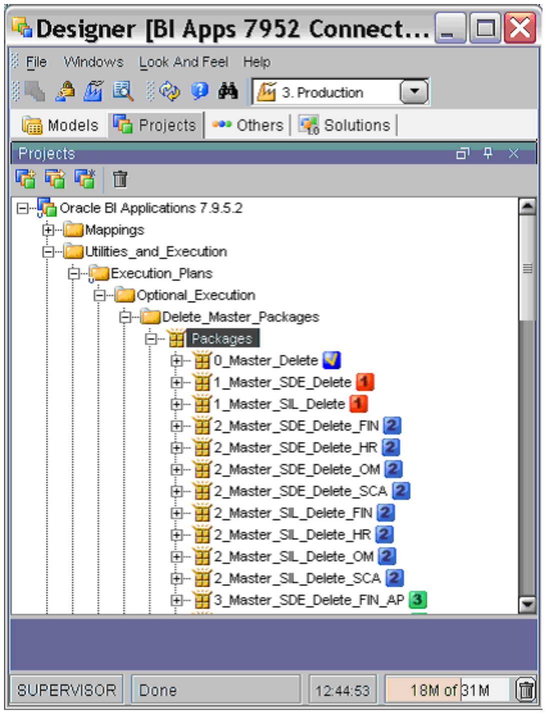
11.4.3.3.1 Deleting the Configuration for Source-Archived Records Some sources archive records in separate databases and retain only the current information in the main database. If you have enabled the delete mappings, you must reconfigure the delete mappings in the Oracle Business Analytics Warehouse to retain the archived data.

To retain source-archived records in the Oracle Business Analytics Warehouse, make sure the LAST_ARCHIVE_DATE parameter value is set properly to reflect your archive date. The delete mappings will not mark the archived records as 'deleted'. For more information about extract and delete mappings, see Section 11.4.3.3, "About Working with Primary Extract and Delete Mappings".

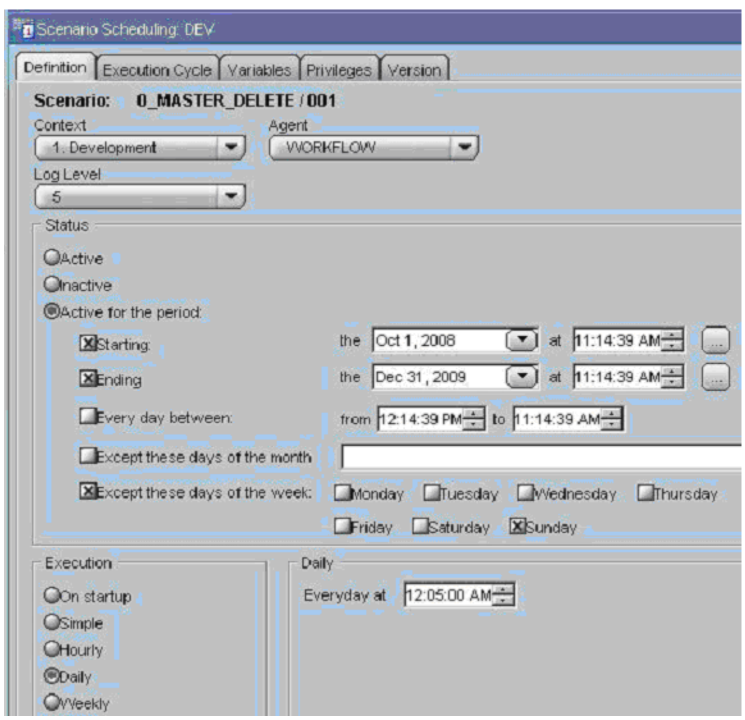
11.4.3.3.2 Enabling Delete and Primary Extract Sessions If you want to mark your source-deleted records as deleted in the Oracle Business Analytics Warehouse, you need to enable the delete and primary extract tasks for your application.

To enable delete and primary extract sessions:

1. In ODI Designer, display the Projects view, and expand the 'Oracle BI Applications 7.9.5.2' folder.
2. Expand the \Master_Packages\Tools\Delete_Master_Packages folder.



3. Double-click the package 0_Master_Delete to display the Scenario: <*Name*> dialog.



4. Use the scheduling settings to schedule the package.

   The jobs for marking the deleted records are already part of the master packages. These jobs will mark the deleted records identified in by the execution of the 0_Master_Delete package.

11.4.4 Configuring Slowly Changing Dimensions

The Oracle Business Analytics Warehouse provides Type II slowly changing dimension (SCD) functionality, which allows you to track the history of updates to dimension records. When a record in the Oracle Business Analytics Warehouse has an update, the updated information is posted into a new row and the old information is kept for historical reporting purposes.

The Oracle Business Analytics Warehouse identifies and applies the slowly changing dimension logic after data has been extracted and transformed to be source-independent. You can configure Oracle Business Intelligence Applications to support both Type I SCDs, in which data is overwritten with updates, and Type II SCDs, in which the original records are maintained while a new record stores the updated data. Whether you choose to apply Type I or Type II SCD functionality to a column depends whether the column contains historically significant attributes.

The table below shows the dimensions that are required using the SCD Type II update.

Table 11–2 Dimensions that are required using the SCD Type II update

Dimension	Adaptor
W_EMPLOYEE_D	Oracle E-Business
W_INVENTORY_PRODUCT_D	Oracle E-Business
W_POSITION_D	Oracle E-Business
W_PRODUCT_D	Oracle E-Business

By default, all dimensions are using Type I updates.

This behavior of TYPE I or TYPE II is managed by a SIL package level Parameter for each dimension called 'TYPE2_FLG'. By default, the value of the parameter is set to 'N' out-of-the-box (except for the above mentioned tables). To turn a dimension to Type II SCD, update enabled set the value of the parameter for the particular dimension - 'TYPE2_FLG' to 'Y'.

11.4.4.1 About Identifying Historically Significant Attributes

You might want to retain a history of all the updates to a particular dimension so that you can use them in reports. These dimensions are known as *historically significant* attributes.

For example, if a customer moves to a different region and you assign that customer a new regional salesperson and territory ID, you might want to keep records of that customer's account history with the original salesperson and territory ID. In this case, the salesperson and territory IDs are *historically significant* attributes.

In contrast, you might have a load that populates the telephone number field. If your business does not perform data analysis on phone number history, then this information might be considered a *historically insignificant* attribute.

Identifying attributes as significant or insignificant allows you to determine the type of SCD you require. However, before you can select the appropriate type of SCD, you must understand their differences.

11.4.4.1.1 About the Extract View The extract view of any given table in the Staging Area consists of four types of records:

- New records

- Changed records with data that is historically insignificant
- Changed records having historical significance
- Changed records whose changes have no significance of any kind and are ignored altogether

Of the four kinds of records, only the first three are of interest for the data mart. Of those three, brand new records and records whose changes are tracked as SCDs are both treated as new and become inserts into the Oracle Business Analytics Warehouse. Records with changes that are important but not historically tracked are overwritten in the Oracle Business Analytics Warehouse, based on the primary key.

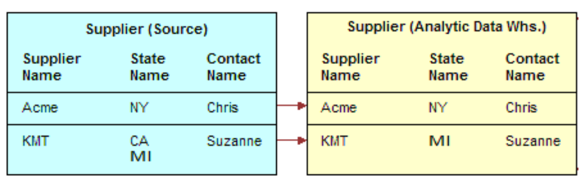
11.4.4.2 Type I and Type II Slowly Changing Dimensions

After you have correctly identified your significant and insignificant attributes, you can configure the Oracle Business Analytics Warehouse based on the type of slowly changing dimension (SCD) that best fits your needs—Type I or Type II.

11.4.4.2.1 Type I Slowly Changing Dimension A Type I SCD overwrites the column's value and is the default SCD for the Oracle Business Analytics Warehouse. Although a Type I does not maintain history, it is the simplest and fastest way to load dimension data. Type I is used when the old value of the changed dimension is not deemed important for tracking or is an historically insignificant attribute. For example, you might want to use Type I when changing incorrect values in a column.

In the figure below, the State Name column for the supplier KMT is changed in the source table Suppliers, because it was incorrectly entered as California. When the data is loaded into the Oracle Business Analytics Warehouse table, no historical data is retained and the value is overwritten. If you look up supplier values for California, records for KMT do not appear; they only appear for Michigan, as they have from the beginning.

Figure 11–8 An example Type 1 Slowly Changing Dimension

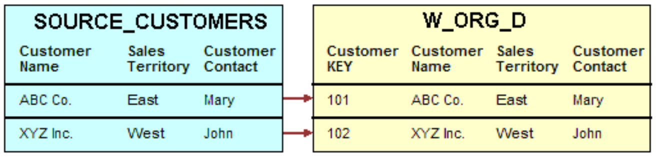


11.4.4.2.2 Type II Slowly Changing Dimension A Type II SCD creates another record and leaves the old record intact. Type II is the most common SCD because it allows you to track historically significant attributes. The old records point to all history prior to the latest change, and the new record maintains the most current information.

Slowly changing dimensions work in different parts of a star schema (the fact table and the dimension table). The figure below shows how an extract table (SOURCE_CUSTOMERS) becomes a Oracle Business Analytics Warehouse dimension table (`W_ORG_D`). Although there are other attributes that are tracked, such as Customer Contact, in this example there is only one *historically tracked attribute,* Sales Territory. This attribute is of historical importance because businesses frequently compare territory statistics to determine performance and compensation. Then, if a customer changes region, the sales activity is recorded with the region that earned it.

This example deals specifically with a single day's extract, which brings in a new record for each customer. The extracted data from `SOURCE_CUSTOMERS` is loaded into the target table W_ORG_D, and each record is assigned a unique primary key (ROW_WID).

Figure 11–9 An example Type 2 Slowly Changing Dimension

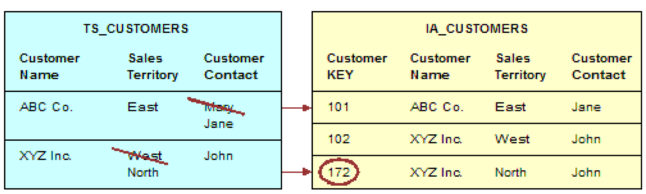


However, this data is not static; the next time a data extract shows a change for your customers in W_ORG_D, the records must change. This situation occurs when slowly changing dimensions are invoked. The figure below shows that records for the two customers, ABC Co., and XYZ inc. have changed when compared with the figure below. Notice that ABC's Customer Contact has changed from Mary to Jane, and XYZ's Sales Territory has changed from West to North.

As discussed earlier in this example, the Customer Contact column is historically insignificant; therefore a Type I SCD is applied and Mary is overwritten with Jane. Because the change in ABC's record was a Type I SCD, there was no reason to create a new customer record. In contrast, the change in XYZ's record shows a change of sales territory, an attribute that is historically significant. In this example, the Type II slowly changing dimension is required.

As shown in the figure below, instead of overwriting the Sales Territory column in the XYZ's record, a new record is added, assigning a new ROW_WID, 172, to XYZ in W_ORG_D. XYZ's original record, 102, remains and is linked to all the sales that occurred when XYZ was located in the West sales territory. However, new sales records coming in are now attributed to ROW_WID 172 in the North sales territory.

Figure 11–10 An example Type 2 Slowly Changing Dimension



11.4.4.2.3 Effective Dates Effective dates specify when a record was effective. For example, if you load a new customer's address on January 10, 2003 and that customer moves locations on January 20, 2003, the address is only effective between these dates. Effective Dates are handled in the following manner:

- If the source supplies both effective dates, these dates are used in the warehouse table.
- If the source does not supply both the effective to and effective from dates, then the Type II logic creates effective dates.
- If the source supplies one of the two effective dates, then you can set up the Oracle Business Analytics Warehouse to populate the missing effective dates using a wrapper mapping. This situation is discussed in this section. By default, these wrapper sessions are disabled and need to be enabled in order to be executed.

For example, in the W_ORG_D table previously discussed, XYZ moved to a new sales territory.

If your source system supplied historical data on the location changes, your table might contain a record for XYZ in the West sales territory with an effective from date of January 1, 2001 and an effective to date of January 1, 3714. If the next year your source indicates XYZ has moved to the North sales territory, then a second record is inserted with an effective from date of January 1, 2002, and an effective to date of January 1, 3714.

11.5 Customizing Stored Lookups, Domain Values, and Adding Indexes

This section contains miscellaneous information that applies to all three categories of customization in Oracle Business Intelligence Applications, and contains the following topics:

- Section 11.5.1, "About Stored Lookups"
- Section 11.5.2, "About Domain Values"
- Section 11.5.3, "How to add an index to an existing fact or dimension table"

11.5.1 About Stored Lookups

This section explains codes lookup and dimension keys.

11.5.1.1 Codes Lookup

Some source systems use intelligent codes that are intuitively descriptive, such as HD for hard disks, while other systems use non-intuitive codes (like numbers, or other vague descriptors), such as 16 for hard disks. While codes are an important tool with which to analyze information, the variety of codes and code descriptions used poses a problem when performing an analysis across source systems. The lack of uniformity in source system codes must be resolved to integrate data for the Oracle Business Analytics Warehouse.

The code lookup in the load mapping integrates both intelligent and non-intuitive codes by performing a separate extract for codes, and inserting the codes and their description into a codes table. The codes table provides the load mapping with a resource from which it can automatically perform a lookup for code descriptions.

The following components and process are used in a lookup:

11.5.1.1.1 W_CODE_D Table The load control table `W_CODE_D` consolidates all codes for future reference and assigns them a category and a single language for efficient lookup capability.

11.5.1.1.2 Codes Mappings The Oracle Business Analytics Warehouse uses mappings designed to extract codes from source systems and populate the `W_CODE_D` table in preparation for use by the load mapping.

To understand how codes mappings function, it is helpful to first understand the columns within `W_CODE_D`, (see table below).

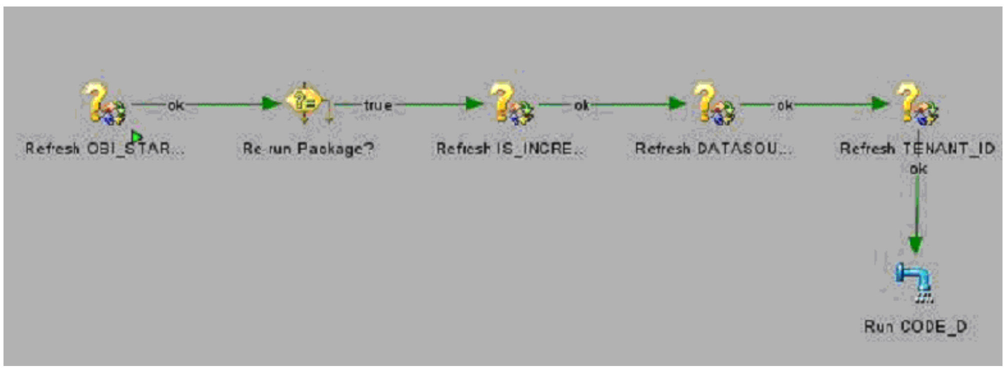
Table 11–3 Columns in Code Mapplet

Column	Description
DATASOURCE_NUM_ID	Unique identifier of the source system from which data was extracted
SOURCE_CODE1	The first code in the hierarchy of the various source system codes used to identify a particular code and description combinations

Table 11–3 (Cont.) Columns in Code Mapplet

Column	Description
SOURCE_CODE2	The second code in the hierarchy of the various source system codes used to identify a particular code and description combinations
SOURCE_CODE3	The third code in the hierarchy of the various source system codes used to identify a particular code and description combinations
SOURCE_DESC_1	Short description of the source system code
SOURCE_DESC_2	Long description for code

The naming convention for mappings designed for codes lookup is `SDE_[SOURCE]_CodeDimension_[CATEGORY]`. The figure below shows an example of a code package in ODI Designer.



11.5.1.2 About Resolving Dimension Keys

By default, dimension key resolution is performed by the Oracle Business Analytics Warehouse in the load mapping. The load interface uses prepackaged, reusable lookup transformations to provide pre-packaged dimension key resolution. This section describes how dimension keys are looked up and resolved.

There are two commonly used methods for resolving dimension keys. The first method, which is the primary method used, is to perform a lookup for the dimension key. The second method is to supply the dimension key directly into the fact load mapping.

11.5.1.2.1 Resolving the Dimension Key Using Lookup If the dimension key is not provided to the Load Interface through database joins, the load mapping performs the lookup in the dimension table. The load mapping does this using prepackaged Lookup Interfaces. To look up a dimension key, the Load Interface uses the INTEGRATION_ID, the DATASOURCE_NUM_ID, and the Lookup date, which are described in the table below.

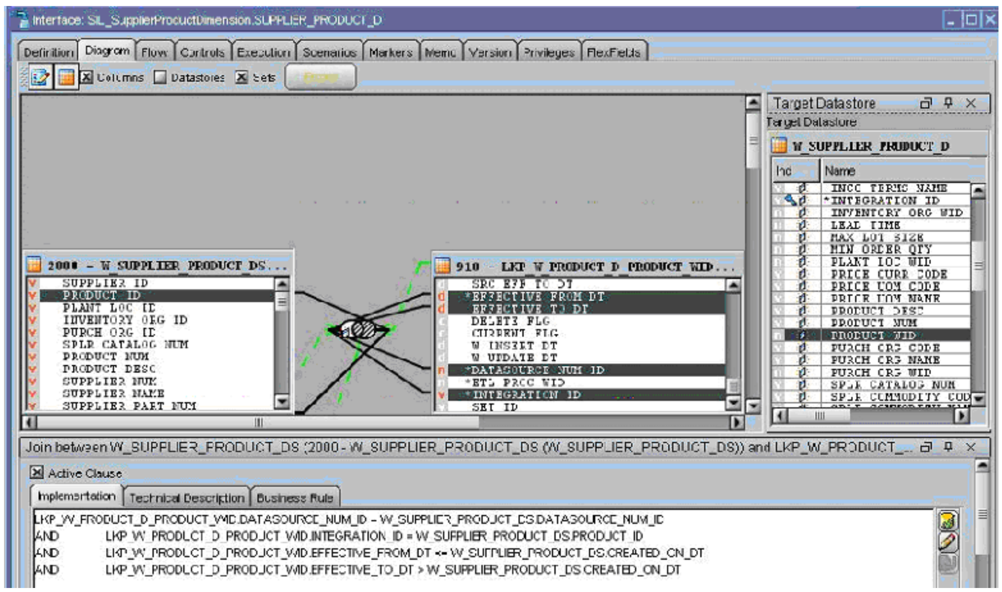
Table 11–4 Columns Used in the load mapping Dimension Key Lookup

Port	Description
INTEGRATION ID	Uniquely identifies the dimension entity within its source system. Formed from the transaction in the Source Adapter of the fact table.
DATASOURCE_NUM_ID	Unique identifier of the source system instance.
Lookup Date	The primary date of the transaction; for example, receipt date, sales date, and so on.

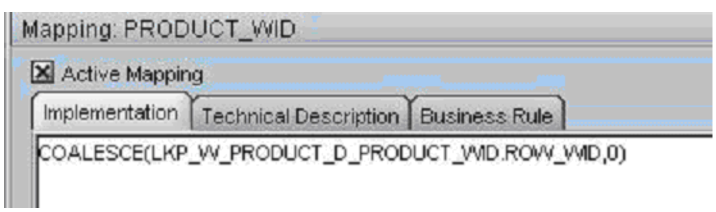
If Type II slowly changing dimensions are enabled, the load mapping uses the unique effective dates for each update of the dimension records. When a dimension key is looked up, it uses the fact's primary date to resolve the appropriate dimension key. The effective date range gives the effective period for the dimension record. The same entity can have multiple records in the dimension table with different effective periods due to Type II slowly changing dimensions. This effective date range is used to exactly identify a record in its dimension, representing the information in a historically accurate manner.

In the figure below, the Supplier Product Key Lookup Interface illustrates the four columns needed for the load interface lookup:

In the figure below, the INTEGRATION ID, DATASOURCE_NUM_ID, Lookup Date (EFFECTIVE_FROM_DT and EFFECTIVE_TO_DATE) are highlighted in the Supplier Product Key Lookup.



The transformation then outputs the Supplier Product Key (the dimension key) to the data warehouse table W_SUPPLIER_PRODUCT_D.



11.5.2 About Domain Values

The Oracle Business Analytics Warehouse foundation comprises a data model that accommodates data from disparate source systems. Data is sourced from operational systems and systematically molded into a source-independent format. After the data is made source independent, it can then be used to create key metrics for analytic reporting, so that metric calculations are not source dependent. This clear separation allows you to swap source systems or integrate additional source systems without having to reconfigure the metric calculations to accommodate each source system's requirements.

One method for transforming source data into a source-independent format is to convert the source-supplied values to domain values. Domain values are a set of distinct values used to calculate prepackaged metrics. These values are provided by the Oracle Business Analytics Warehouse to allow you to create metric calculations independent of source system values.

11.5.2.1 About the Domain Value Conversion Process

To best understand the domain value conversion process, consider an example of two source systems—Source System A and Source System B. Each source system stores two types of employee events—hire and rehire. Source system A uses H to denote a hire event and R to denote a rehire event, whereas source system B uses 1 to denote a hire event and 2 to denote a rehire event. When the Oracle Business Analytics Warehouse extracts data from both systems, it ports those source values through the extract package until the data reaches the W_EVENT_GRP_CODE column in the W_EVENT_TYPE_DS staging table. The load package then ports the extracted source values (H and R from source system A, and 1 and 2 from source system B) into the interface. Within the interface, source values are translated into domain values (HIR and REH) based on a set of rules that are particular to your business practices.

11.5.2.1.1 Preparing to Define the Rules You must define the rules so that the ODI interface knows how to map your specific source values to the given set of domain values. Before you set up the rules you must:

1. Analyze all of your source values and how they map to the prepackaged domain values. You might find that you need to create additional domain values for particular columns. The result of this preparation work is a list of each source value and how it is mapped to a domain value.
2. Implement this logic in the ODI Interface. To set up the logic, modify the Expression transformation in the ODI Interface for each affected column.

After the ODI Interface converts the source-specific values to domain values, the domain values are inserted into an Oracle Business Analytics Warehouse table. In this example, the HIR and REH values populate the W_EVENT_TYPES table (Interface SDE_ORA_EventTypeDimension_AdditionalEvents_FromFile.EVENT_TYPE_DS).

11.5.2.2 About the Importance of Domain Values

Values in the `W_EVENT_TYPES` table are used to create metrics in the front end. Some metrics are defined using domain values. For example, seven metrics use the HIR and REH event group code in their calculation. The following are the seven metrics, along with their descriptions and calculations:

11.5.2.2.1 Hire Count This metric counts all hires for a specified period. The calculation is:

```
SUM(CASE WHEN (CMMNEVTP.W_EVENT_GRP_CODE IN ('HIR','REH')) THEN EVNT.EVENT_CNT
ELSE 0 END)
```

11.5.2.2.2 Re-hires Ratio This metric determines the ratio of rehires to all employees hired during a specified period. The calculation is:

```
CASE WHEN SUM(CASE WHEN CMMNEVTP.W_EVENT_GRP_CODE IN ('REH','HIR') THEN
EVNT.EVENT_CNT ELSE 0 END) = 0 THEN 0 ELSE SUM(CASE WHEN CMMNEVTP.W_EVENT_GRP_CODE
IN ('REH') THEN EVNT.EVENT_CNT ELSE 0 END)/SUM(CASE WHEN CMMNEVTP.W_EVENT_GRP_CODE
IN ('REH','HIR') THEN EVNT.EVENT_CNT ELSE 0 END) END
```

11.5.2.2.3 New Hire Count This metric counts the head count hired for regular full-time positions. The calculation is:

```
SUM(CASE WHEN CMMNEMPT.FULL_TIME_FLAG = 'Y' AND CMMNEMPT.EMP_CAT_CODE = 'R' AND
(CMMNEVTP.W_EVENT_GRP_CODE = 'HIR' OR CMMNEVTP.W_EVENT_GRP_CODE = 'REH') AND
EVNT.EVENT_DK >= (CMMNDATE.DATE_KEY - 365) AND EVNT.EVENT_DK <= CMMNDATE.DATE_KEY
THEN EVNT.EVENT_CNT ELSE 0 END)
```

11.5.2.2.4 Newly Separated Veterans - New Hires This metric counts the regular full-time and part-time employees who belong to this category of veterans and were hired during the previous 12 months. The calculation is:

```
SUM(CASE WHEN CMMNEMPD.VETERAN_STAT_CODE = '4' AND CMMNEMPT.EMP_CAT_CODE = 'R' AND
(CMMNEVTP.W_EVENT_GRP_CODE = 'HIR' OR CMMNEVTP.W_EVENT_GRP_CODE = 'REH') AND
EVNT.EVENT_DK >= (CMMNDATE.DATE_KEY - 365) AND EVNT.EVENT_DK <= CMMNDATE.DATE_KEY
THEN EVNT.EVENT_CNT ELSE 0 END)
```

11.5.2.2.5 Other Protected Veterans - New Hires This metric counts regular full-time and part-time employees who belong to this category of veterans. The calculation is:

```
SUM(CASE WHEN CMMNEMPD.VETERAN_STAT_CODE = '3' AND CMMNEMPT.EMP_CAT_CODE = 'R' AND
(CMMNEVTP.W_EVENT_GRP_CODE = 'HIR' OR CMMNEVTP.W_EVENT_GRP_CODE = 'REH') AND
EVNT.EVENT_DK >= (CMMNDATE.DATE_KEY - 365) AND EVNT.EVENT_DK <= CMMNDATE.DATE_KEY
THEN EVNT.EVENT_CNT ELSE 0 END)
```

11.5.2.2.6 Special Disabled Veteran Head count - New Hires This metric counts regular full-time and part-time employees who belong to this category of veterans and were hired during the previous 12 months. The calculation is:

```
SUM(CASE WHEN CMMNEMPD.VETERAN_STAT_CODE = '1' AND CMMNEMPT.EMP_CAT_CODE = 'R' AND
(CMMNEVTP.W_EVENT_GRP_CODE = 'HIR' OR CMMNEVTP.W_EVENT_GRP_CODE = 'REH') AND
EVNT.EVENT_DK >= (CMMNDATE.DATE_KEY - 365) AND EVNT.EVENT_DK <= CMMNDATE.DATE_KEY
THEN EVNT.EVENT_CNT ELSE 0 END)
```

11.5.2.2.7 Vietnam Era Veteran Head count - New Hires This metric counts regular full-time and part-time employees who belong to this category of veterans and were hired during the previous 12 months. The calculation is:

```
SUM(CASE WHEN CMMNEMPD.VETERAN_STAT_CODE = '2' AND CMMNEMPT.EMP_CAT_CODE = 'R' AND
(CMMNEVTP.W_EVENT_GRP_CODE = 'HIR' OR CMMNEVTP.W_EVENT_GRP_CODE = 'REH') AND
EVNT.EVENT_DK >= (CMMNDATE.DATE_KEY - 365) AND EVNT.EVENT_DK <= CMMNDATE.DATE_KEY
THEN EVNT.EVENT_CNT ELSE 0 END)
```

11.5.2.3 About Extending the Domain Value Set

The Oracle Business Analytics Warehouse is also extensible in that you can create additional domain values for those columns that do not fit into the existing domain value definitions. However, before you modify the domain value set for a particular column, you first perform impact analysis on existing metrics. For example, the Oracle Business Analytics Warehouse prepackages the following two events:

- **New Hire.** This event occurs when a new person is hired.
- **New Position.** This event occurs when a position is created, but an existing employee might be hired internally.

If you have an event that represents both a New Hire and a New Position, you might have to create a third event that depicts both. If you create this new event type domain value, you need to include it in the applicable metric definitions so as to account for all hires and positions.

11.5.2.4 Configuring the Domain Value Set with CSV Worksheet Files

Domain values are a set of distinct values used to calculate prepackaged metrics. These values are provided by Oracle Business Analytics Warehouse to allow you to create metric calculations independent of source system values. Oracle Business Analytics Warehouse provides CSV worksheet files to map source system values to domain values.

You can add to these worksheet files if you need extra source system values and map them to domain values. You can also modify the worksheet files if you need to customize the domain values. You can use an existing domain value if you want to change the preconfigured metrics. Otherwise you can create a new domain value and create new metrics based on this domain value.

The source system values that are not mapped to a domain values in the CSV worksheet files have a question mark (?) as the domain value in the Oracle Business Analytics Warehouse. These values do not affect the domain values metrics.

If there are no worksheet files to map the source system values to the domain values, you need to modify the domain values using ODI (for more information, see Section 11.5.2.5, "Configuring the Domain Value Set Using ODI Designer").

To map source values to domain values using CSV worksheet files:

1. Identify the Oracle Business Analytics Warehouse table columns that use domain values.

   For a list of columns that use domain values, see the Oracle Business Analytics Warehouse Data Model Reference.

2. List all of your source values that qualify for conversion to one of the domain values.

3. Map each source value to a domain value.

   If any of your source system values do not map to a prepackaged domain value, and you can modify the list of domain values, then create a list of new domain values and map your orphaned source system values to your newly created domain values.

   You cannot modify all domain value sets. Also, you must check which metrics are affected by the modified domain value set. For more information, see the Oracle Business Analytics Warehouse Data Model Reference.

4. Open the CSV worksheet file in the $ODI_HOME\ oracledi\biapps_odi\odifiles\odidatafiles\lkpfiles folder (for example, C:\ODI\oracledi\biapps_odi\odifiles\odidatafiles\lkpfiles).

5. Edit the file to map your source values to the existing domain values.

   Alternately, if you want to add additional domain values, add them in this worksheet file.

6. Save and close the CSV file.

11.5.2.5 Configuring the Domain Value Set Using ODI Designer

If there are no worksheet files to map the source system values to the domain values, you need to modify the values using ODI Designer. For more information on configuring the domain value set with CSV worksheet files, see Section 11.5.2.4, "Configuring the Domain Value Set with CSV Worksheet Files".

Configuring the domain value set for a particular column, using ODI, entails one or both of the following activities:

- Mapping source-specific values to domain values
- Adding more domain values to the prepackaged set of values

Regardless of which activity you choose, the configuration occurs in the Expression transformation of the applicable Source Adapter mapplet. The following procedure shows how to configure the Expression transformation to change the domain values.

To map source values to domain values:

1. Identify all the Oracle Business Analytics Warehouse table columns that use domain values.

   For a list of columns that use domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

2. List all of your source values that qualify for conversion to one of the domain values.
3. Map each source value to a domain value.

   If any of your source system values do not map to a prepackaged domain value, and you might modify the list of domain values, then create a list of new domain values and map your orphaned source system values to your newly created domain values.

   You cannot modify all domain value sets. Also, you must check which metrics are affected by the modified domain value set. For more information, see the Oracle Business Analytics Warehouse Data Model Reference.

4. In ODI Designer, open the applicable extract inline view (inline views follows the SQ_* naming convention).
5. Open the Expression of the applicable column that uses domain values.

   Alternatively, if you want to add domain values, add them to this expression.

6. Save the changes.
7. Regenerate the scenario.

11.5.3 How to add an index to an existing fact or dimension table

Dimension and Fact Tables in the Oracle Business Analytics Warehouse use the following two types of index:

- E-LT Index

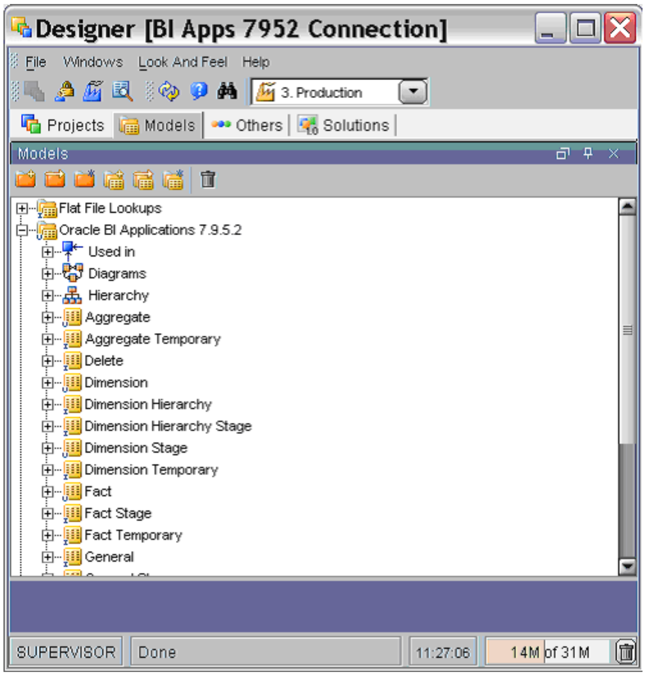
  E-LT Indexes are used for Unique/ Binary Tree index.

- Query Index

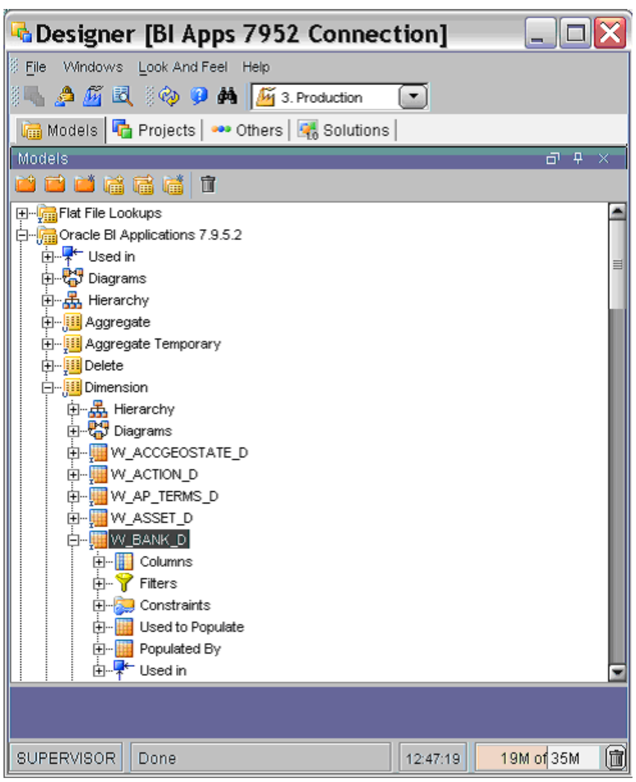
  Query Indexes are used for Non-Unique/ Bit Map Index.

To add an index to an existing fact or dimension table:

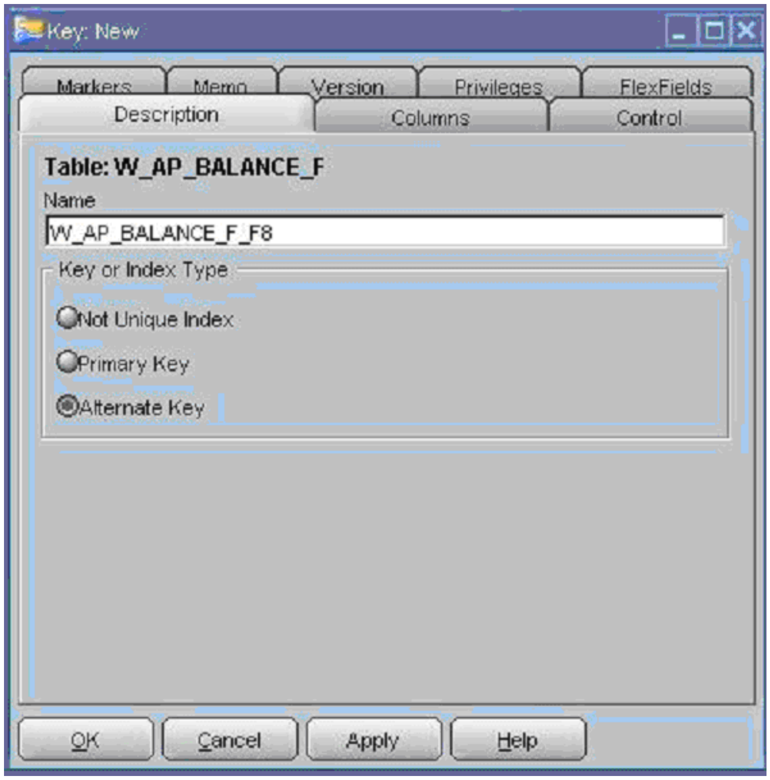
1. In ODI Designer, display the Models view, and expand the 'Oracle BI Applications 7.9.5.2' folder.



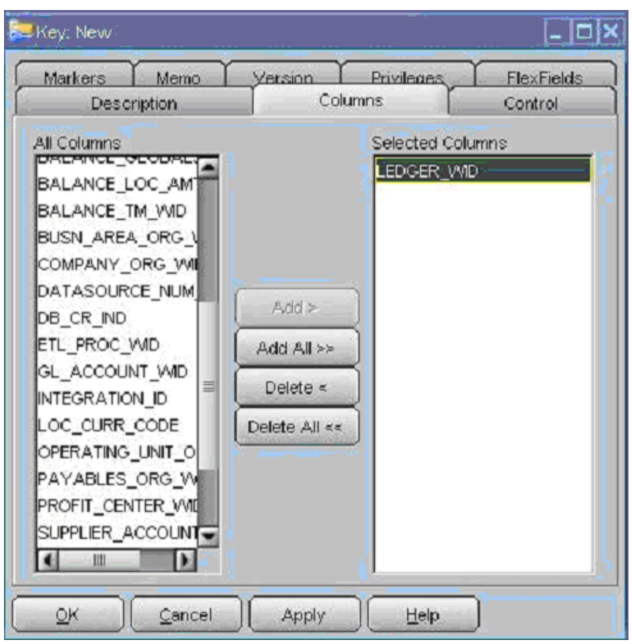
2. Expand the Fact or Dimension node as appropriate.
3. Expand the Table in which you want to create the index.



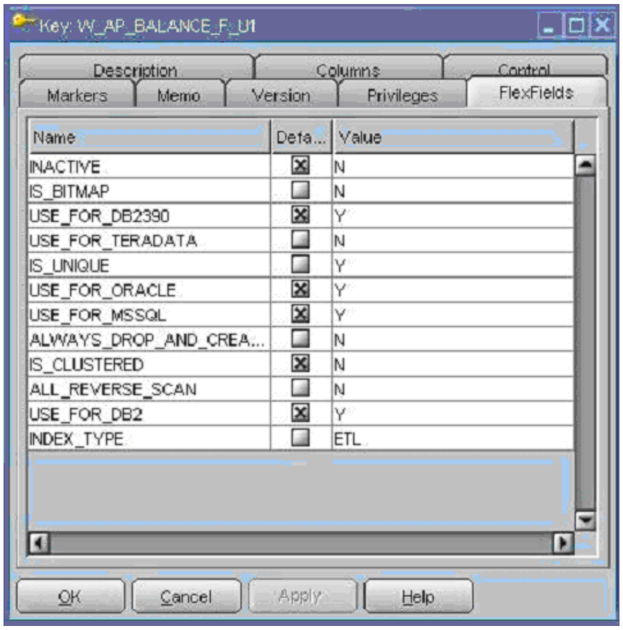
4. Right-click on the Constraints node, and select Insert Key to display the Key: New dialog.
5. Display the Description tab.



6. Select the **Alternate Key** radio button, and update the name of the Index in the **Name** field.
7. Display the Column tab.



8. Select the column on which you want to create the index.
9. Display the FlexFields tab.



10. Use the settings to specify the index type, as follows:

    - For 'Query' type indexes (the default), set the value of the IS_BITMAP parameter to 'Y' and the value of the IS_UNIQUE parameter to 'N'.
    - For 'E-LT' type indexes, clear the check box for the INDEX_TYPE parameter and set the value to 'ETL'. In addition, set the value of the IS_BITMAP parameter to 'N' and the value of the IS_UNIQUE parameter to 'Y'.

11. Save the changes.

Index

N

O

P