Oracle® Business Intelligence Applications

Configuration Guide for Informatica PowerCenter Users

Release 7.9.6.3

E19039-01

April 2011

Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users, Release 7.9.6.3

E19039-01

Contents

5 Configuring Oracle Financial Analytics

6 Configuring Oracle Supply Chain and Order Management Analytics

7 Configuring Oracle Human Resources Analytics

8 Configuring Oracle Sales Analytics

10 Configuring Oracle Price Analytics

11 Configuring Oracle Service Analytics

12 Configuring Oracle Marketing Analytics

13 Configuring Oracle Loyalty Analytics

14 Configuring Oracle Pharma Analytics

15 Configuring Oracle Project Analytics

16 Configuring the Oracle BI Repository

A Configuring Universal Adapter for CRM Analytics

Index

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, JD Edwards, and corporate data warehouses — into actionable insight that enables more effective actions, decisions, and processes.

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

Oracle BI Applications includes the following application families:

- Oracle Contact Center Telephony Analytics
- Oracle Financial Analytics
- Oracle Human Resources Analytics
- Oracle Loyalty Analytics
- Oracle Marketing Analytics
- Oracle Pharma Marketing Analytics
- Oracle Pharma Sales Analytics
- Oracle Price Analytics
- Oracle Procurement and Spend Analytics
- Oracle Project Analytics
- Oracle Sales Analytics
- Oracle Service Analytics
- Oracle Supply Chain and Order Management Analytics

Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users contains instructions for configuring and customizing Oracle BI Applications Release 7.9.6.3.

Oracle recommends reading the *Oracle Business Intelligence Applications Release Notes* before installing, using, or upgrading Oracle BI Applications. The *Oracle Business Intelligence Applications Release Notes* are available on the Oracle Business Intelligence Applications CD-ROM, or on the Oracle Technology Network at:

`http://www.oracle.com/technology/documentation/bi_apps.html`

Audience

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

> **Note:** Many configuration tasks described in this guide require you to manually enter values obtained from querying your source system (for example, Oracle E-Business Suite). These values are unique to your source system and implementation. To retrieve these values successfully, you need to have a good technical understanding of your source system. If you need assistance in obtaining values from your source system, you should consult with someone in your organization who possesses this knowledge, or consult the Oracle Support Services team for your source system. Make sure to check your configuration entries carefully to avoid data loss during the ETL process.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/us/corporate/accessibility/index.html`.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/support/contact.html` or visit `http://www.oracle.com/accessibility/support.html` if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle BI Applications Release 7.9.6.3 documentation set (available at `http://www.oracle.com/technology/documentation/bi_apps.html`):

- *Oracle Business Intelligence Applications Release Notes*
- *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*
- *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*
- *Oracle Business Intelligence Applications Upgrade Guide for Informatica PowerCenter Users*
- *Oracle Business Intelligence Applications Naming Conventions and Domain Values Guide*
- *Oracle Business Intelligence Applications Security Guide*
- *Oracle Business Analytics Warehouse Data Model Reference*

Also see the Oracle Business Intelligence Data Warehouse Administration Console documentation set (available at `http://www.oracle.com/technology/documentation/bi_dac.html`):

- *Oracle Business Intelligence Data Warehouse Administration Console Release Notes*

- *System Requirements and Supported Platforms for Oracle Business Intelligence Data Warehouse Administration Console*
- *Oracle Business Intelligence Data Warehouse Administration Console User's Guide*
- *Oracle Business Intelligence Data Warehouse Administration Console Installation, Configuration, and Upgrade 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 I helps you get started with configuring Oracle BI Applications. It contains the following sections:

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

1

What's New in This Release

This section lists changes described in this version of the documentation to support Release 7.9.6.3 of the software.

1.1 What's New in Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users

This guide includes the following changes:

- Sections in this guide that contain information about Informatica have been updated for Informatica Version 9.
- Information was included in this guide that was previously published in *Oracle Business Intelligence Applications New Features Guide*, Version 7.9.6.2.
- Tasks and procedures relating to the Oracle BI Administration Tool were updated to reflect changes to the underlying Oracle BI Enterprise Edition platform version.
- Additional information was provided about mapping GL accounts to group account numbers.
- Enhancements and corrections were included for CSV file configuration for several applications.
- Information relating to security has been updated. For example, references to user groups have been updated to application roles.

> **Note:** Many configuration tasks described in this guide require you to manually enter values obtained from querying your source system (for example, Oracle E-Business Suite). These values are unique to your source system and implementation. To retrieve these values successfully, you need to have a good technical understanding of your source system. If you need assistance in obtaining values from your source system, you should consult with someone in your organization who possesses this knowledge, or consult the Oracle Support Services team for your source system. Make sure to check your configuration entries carefully to avoid data loss during the ETL process.

2 Overview of Configuring Oracle Business Intelligence Applications

This section includes an overview of how to configure Oracle Business Intelligence Applications.

2.1 High-Level Overview of Configuring Oracle BI Applications

This section provides a high-level overview of the tasks you must complete to configure Oracle BI Applications.

Note: Before you perform the tasks described in this section, you must install and set up Oracle BI Applications according to the instructions in *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

High-Level Configuration Tasks

1. Perform the tasks in Section 3.1, "Source-Independent Configuration Steps."
2. Perform the tasks in one or more of the following sections, depending on the type of your source system or systems:
   - Section 3.2, "Oracle EBS-Specific Configuration Steps"
   - Section 3.3, "PeopleSoft-Specific Configuration Steps"
   - Section 3.4, "Oracle Siebel-Specific Configuration Steps"
   - Section 3.5, "Oracle's JD Edwards-Specific Configuration Steps"
3. Perform the tasks in one or more of the following sections depending on the applications you want to deploy:
   - Chapter 4, "Configuring Oracle Procurement and Spend Analytics"
   - Chapter 5, "Configuring Oracle Financial Analytics"
   - Chapter 6, "Configuring Oracle Supply Chain and Order Management Analytics"
   - Chapter 7, "Configuring Oracle Human Resources Analytics"
   - Chapter 8, "Configuring Oracle Sales Analytics"
   - Chapter 9, "Configuring Oracle Contact Center Telephony Analytics"
   - Chapter 10, "Configuring Oracle Price Analytics"
   - Chapter 11, "Configuring Oracle Service Analytics"
   - Chapter 12, "Configuring Oracle Marketing Analytics"

- Chapter 13, "Configuring Oracle Loyalty Analytics"
- Chapter 14, "Configuring Oracle Pharma Analytics"
- Chapter 15, "Configuring Oracle Project Analytics"

4. (Optional) Configure the Oracle BI Repository by following the tasks in Chapter 16, "Configuring the Oracle BI Repository."
5. (Optional) Customize the Oracle Business Analytics Warehouse ETL routines by following the tasks in Chapter 17, "Customizing the Oracle Business Analytics Warehouse."

Part II

Configuring Common Areas and Dimensions

Part II contains common configuration tasks that apply to Oracle BI Applications deployed with any source system, as well as mandatory and additional tasks that are specific to the various source systems.

Note: Before you configure Oracle BI Applications, you need to install and set up Oracle BI Applications by following the instructions in *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

Part II includes the following section:

- Chapter 3, "Configuring Common Areas and Dimensions"

3 Configuring Common Areas and Dimensions

This section contains mandatory and additional configuration tasks that apply to Oracle BI Applications deployed with any source system, as well as mandatory and additional tasks that are specific to the various source systems.

To configure Oracle BI Applications, you must first perform the steps in Section 3.1, "Source-Independent Configuration Steps."

> **Note:** Many configuration tasks described in this guide require you to manually enter values obtained from querying your source system (for example, Oracle E-Business Suite). These values are unique to your source system and implementation. To retrieve these values successfully, you need to have a good technical understanding of your source system. If you need assistance in obtaining values from your source system, you should consult with someone in your organization who possesses this knowledge, or consult the Oracle Support Services team for your source system. Make sure to check your configuration entries carefully to avoid data loss during the ETL process.

You must then perform the tasks in one of the following sections depending on the type of your source system:

- Section 3.2, "Oracle EBS-Specific Configuration Steps"
- Section 3.3, "PeopleSoft-Specific Configuration Steps"
- Section 3.4, "Oracle Siebel-Specific Configuration Steps"
- Section 3.5, "Oracle's JD Edwards-Specific Configuration Steps"

3.1 Source-Independent Configuration Steps

This section contains configuration steps that apply to Oracle BI Applications deployed with any source system. It contains the following topics:

- Section 3.1.1, "How to Configure Initial Extract Date"
- Section 3.1.2, "How to Configure Global Currencies"
- Section 3.1.3, "How to Configure Exchange Rate Types"
- Section 3.1.4, "About Configuring Calendars"

- Section 3.1.5, "About Mapping GL Accounts to Group Account Numbers for All Source Systems"
- Section 3.1.6, "Configuration Steps for Controlling Your Data Set for All Source Systems"
- Section 3.1.7, "About Support for Back-Dated Change Scenarios"
- Section 3.1.8, "Supported Back-Dated Change Scenarios That Affect Position Hierarchy"

3.1.1 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 the transactional data in the selected full extract mapping. The default date is January 01, 1970.

When you set the Initial Extract Date parameter, ensure 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 on June 5, set the date to June 5, 2005.

To configure the initial extract date:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab.
3. Edit the value of the $$INITIAL_EXTRACT_DATE parameter.
4. Save your changes.

3.1.2 How to Configure 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 document currency is 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 United Kingdom and filed an expense report for meal expenses in the UK, the document currency of the expense report will most likely be in GBP.
- Local currency. The local currency is the base currency of your ledger, or the currency in which your accounting entries are recorded.
- Global currencies. Oracle BI Applications provides three global currencies, which are the common currencies used by the Oracle Business Analytics Warehouse. 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 and local amounts into the target table. It also loads the exchange rates required to convert the document amount into each of the three global currencies. For fact tables, there are two amount columns covering the Local currency amount and the Document currency amount. In addition, there are three columns covering the Global currency (for example, global _amount1) and their corresponding exchange rate columns.

In most cases, the source system provides the document currency amount. This is the most common situation, and, thus, is the Oracle Business Analytics Warehouse's default for handling currency. 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 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 setup from DAC parameters and look up the corresponding exchange rates to each of the three global currencies.

To configure the global currencies you want to report:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab.
3. Locate the following parameters, and set the currency code values for them in the Value field:
   - $$GLOBAL1_CURR_CODE (for the first global currency).
   - $$GLOBAL2_CURR_CODE (for the second global currency).
   - $$GLOBAL3_CURR_CODE (for the third global currency).

   You can specify any currency, provided an exchange rate is specified for the conversion to the local currency. Ensure that you spell the currencies as they are spelled in your source OLTP system.
4. Save your changes.

3.1.3 How to Configure Exchange Rate Types

When Oracle BI 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 BI Applications also enables you to specify the exchange rate type to use to perform the conversion. Oracle BI Applications also provides three global exchange rate types for you to configure.

Oracle BI 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 Applications also enables you to configure the rate type that you want to use when converting the document currency to the local currency.

To configure exchange rate types:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab.
3. Locate the following DAC parameters and set the exchange rate type values for them in the Value field:

- $$GLOBAL1_RATE_TYPE
- $$GLOBAL2_RATE_TYPE
- $$GLOBAL3_RATE_TYPE
- $$DEFAULT_LOC_RATE_TYPE (the conversion rate type for document currency to local currency conversion).

Make sure you spell the exchange rate type values as they are spelled in your source OLTP system.

4. Save your changes

3.1.4 About Configuring Calendars

This section explains how to set up the different types of calendar that are supported by Oracle Business Intelligence Applications, and contains the following topics:

3.1.4.1 Overview of Calendars in Oracle BI Applications

Oracle Business Intelligence Applications Release 7.9.6.3 supports the following calendar formats:

- Enterprise (Global) - cross functional reporting calendar, which can be fiscal or gregorian.
- Fiscal - accounting or financial calendar.
- Gregorian - regular calendar that starts on January 1st and ends on December 31st.

- 13 Period - each year consists of 13 periods of 28 days.
- 4-4-5 - each year is composed of twelve periods of either four weeks of 28 days or five weeks of 35 days.

3.1.4.1.1 About Enabling Initialization Blocks Required for Calendars If you want to deploy multi-calendar (for example, with Oracle Financial Analytics), you must have the following initialization blocks enabled:

- EBS Single Sign-on Integration
- EBS Security Context
- Ledgers
- Operating Unit Organizations

These initialization blocks are not enabled by default. If you try to load data without enabling these initialization blocks, dashboards will not be populated with data, and the BI Server log might show initialization block errors (for example, nQSError:43059).

To enable initialization blocks for multi-calendar:

1. Using the Administration Tool, open OracleBIAnalyticsApps.rpd.

   The OracleBIAnalyticsApps.rpd file is located at:

   ```
   ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_
   obisn\repository
   ```

2. Choose **Manage**, then **Variables** to display the Variable Manager.
3. In the left-hand pane, select **Session**, then **Initialization Blocks**.
4. For each initialization block listed in this section, right-click the initialization block name in the right-hand pane, then select **Enable**.
5. Restart the Oracle BI Server.

3.1.4.1.2 About Calendar Tables This section describes the tables used for Time Dimension calendars (including Gregorian calendar, Fiscal calendar, and Enterprise Calendar).

Gregorian Calendar Tables

- W_WEEK_D
- W_MONTH_D
- W_QTR_D
- W_YEAR_D
- W_DAY_D

Fiscal Calendar Tables

- W_MCAL_DAY_D
- W_MCAL_WEEK_D
- W_MCAL_PERIOD_D
- W_MCAL_QTR_D
- W_MCAL_YEAR_D

Enterprise Calendar Tables

- W_ENT_WEEK_D
- W_ENT_PERIOD_D
- W_ENT_QTR_D
- W_ENT_YEAR_D

Table 3–1 shows Time Dimension Configuration and Context Tables.

Table 3–1 Time Dimension Configuration and Context Tables

Configuration Table	PeopleSoft Specific Configuration Table	Context Table
W_MCAL_CONFIG_G	W_MCAL_PSFT_SUMP_CONFIG_G	W_MCAL_CONTEXT_G

For more information about the configuration tables and context tables, see Section 3.1.4.2, "About Configuring Calendars."

Note: The following tables are obsolete in Oracle Business Intelligence Applications Version 7.9.6 and later:

- W_FSCL_WEEK_D
- W_FSCL_MONTH_D
- W_FSCL_QTR_D
- W_FSCL_YEAR_D

Note that the tables listed in the preceding list were populated through CSV files in prior releases. Refer to the upgrade documentation if you need to migrate data in these tables into the new fiscal calendar tables.

3.1.4.1.3 About Calendar Categories Calendars are categorized into two types:

- OLTP sourced (also known as Source Calendars)

  OLTP sourced calendars are calendars that are defined in ERP sources and brought into the warehouse through ETL maps.

- Warehouse generated (also known as Generated Calendars)

  Generated calendars are fiscal calendars generated in the warehouse based on configuration files.

Both source calendars and generated calendars are stored in the Multiple Fiscal Calendar (known as MCAL) tables. MCAL tables have the prefix W_MCAL.

3.1.4.2 About Configuring Calendars

This section explains how to configure the different types of supported calendar.

3.1.4.2.1 About Setting Up Gregorian Calendars Whatever calendar type you want to deploy, you must set up the start date and end date range for the Gregorian calendar. See Section 3.1.4.4, "How to Set Up the Gregorian Calendar Date Range" for more information.

3.1.4.2.2 Prerequisites to Populating the MCAL tables W_DAY_D is the base table that represents the time dimension in the Oracle Business Analytics Warehouse. This table must be populated as a prerequisite for the multiple fiscal calendar tables. If W_DAY_D is not populated, then the fiscal calendar tables will not be populated.

There are two parameters, $$START_DATE and $$END_DATE, for the task SIL_DayDimension that must be set up 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. After the records are created in W_DAY_D, the aggregate calendar tables are loaded by their respective SIL mapping. Then the fiscal calendar tables (known as MCAL tables) are populated.

Note: The parameters $$START_DATE and $$END_DATE must include all dates covered by any of the fiscal calendars brought into the warehouse. These parameters are the boundaries for the date dimension and related tables.

3.1.4.2.3 About Configuring Enterprise Calendars An Enterprise calendar (or reporting calendar) enables cross Subject Area analysis. Enterprise calendar tables have the W_ENT prefix.

Enterprise calendars can be set to one of the OLTP sourced fiscal calendars, or to one of the warehouse generated calendars. This can be done by setting the following source system parameters at the DAC container level:

- $$GBL_CALENDAR_ID
- $$GBL_DATSOURCE_NUM_ID

The following sections show how to set up the source system parameters for the Enterprise calendar in different scenarios, as follows:

- "Scenario 1: Using an Oracle EBS fiscal calendar as the Enterprise calendar"
- "Scenario 2: Using a PeopleSoft fiscal calendar as the Enterprise calendar"
- "Scenario 3: Using a warehouse generated calendar as the Enterprise calendar"
- "Scenario 4: Using a fiscal calendar loaded through the Universal Adapter as the Enterprise calendar"
- "Scenario 5: Using an Oracle JD Edwards EnterpriseOne fiscal calendar as the Enterprise calendar"
- "Scenario 6: Using an Oracle JD Edwards World fiscal calendar as the Enterprise calendar"

Scenario 1: Using an Oracle EBS fiscal calendar as the Enterprise calendar

Source System DAC Parameters for Oracle EBS Enterprise Calendars:

- GBL_CALENDAR_ID: This parameter is used to select the Enterprise Calendar. It should be the MCAL_CAL_NAME~MCAL_PERIOD_TYPE for Non-Generated Calendars. For example, GBL_CALENDAR_ID will be 'Accounting~41' if the Enterprise Calendar id='Accounting' and the calendar period_type='41.'
- GBL_DATASOURCE_NUM_ID: If Enterprise Calendar is not a Generated Calendar: It should be the DATASOURCE_NUM_ID of the source system from where the Calendar definition is taken. For example, if you have two data sources, PeopleSoft and Oracle, and the Global Calendar is from an Oracle data source, then this parameter value should specify an Oracle data source.

Scenario 2: Using a PeopleSoft fiscal calendar as the Enterprise calendar

Source System DAC Parameters for PeopleSoft Enterprise Calendars:

- GBL_CALENDAR_ID: This parameter is used to select the Enterprise Calendar. It should be the SETID~CALENDAR_ID for Non-Generated Calendars. For example, GBL_CALENDAR_ID will be 'SHARE~01' if the Enterprise Calendar id='01' and SETID='SHARE.'
- GBL_DATASOURCE_NUM_ID: If Global Calendar is not a Generated Calendar: It should be the DATASOURCE_NUM_ID of the source system from where the Calendar definition is taken. For example, if you have two data sources, PeopleSoft and Oracle, and the Global Calendar is from Oracle source, then this parameter value should specify an Oracle data source.

Scenario 3: Using a warehouse generated calendar as the Enterprise calendar

Source System DAC Parameters for Generated Enterprise Calendars:

- GBL_CALENDAR_ID: This parameter should be the CALENDAR_ID of the Generated Calendar (4-4-5 or 13 period type of Calendars). By default, the 4-4-5 calendar has a CALENDAR_ID of '10000,' and the 13-period calendar has a CALENDAR_ID of '10001.'
- GBL_DATASOURCE_NUM_ID: If Global Calendar is a Generated Calendar: It should be the DATASOURCE_NUM_ID value of the OLAP (Data warehouse).

Scenario 4: Using a fiscal calendar loaded through the Universal Adapter as the Enterprise calendar

Source System DAC Parameters for Universal Enterprise Calendars:

- GBL_CALENDAR_ID: This parameter should be the INTEGRATION_ID from the file_mcal_cal_d.csv file of the particular calendar which is defined as the Global Calendar.
- GBL_DATASOURCE_NUM_ID: If Global Calendar is not a Generated Calendar, then it should be the DATASOURCE_NUM_ID of the source system from where the Calendar definition is taken. If it is defined in the file_mcal_period_ds.csv file, then that value should be taken, or else as defined in DAC for the Universal adapter.

Scenario 5: Using an Oracle JD Edwards EnterpriseOne fiscal calendar as the Enterprise calendar

Source System DAC Parameters for Oracle JD Edwards EnterpriseOne to configure Enterprise Calendars:

- GBL_CALENDAR_ID: This parameter is used to select the Enterprise Calendar. It should be MCAL_CAL_ID. For example, GBL_CALENDAR_ID will be 'R' if the Enterprise Calendar id='R.'
- GBL_DATASOURCE_NUM_ID: If Enterprise Calendar is not a Generated Calendar, this value should be the DATASOURCE_NUM_ID of the source system from where the Calendar definition is taken. For example, if you have two data sources, Oracle and Oracle JD Edwards EnterpriseOne, and the Global Calendar is from an Oracle JD Edwards EnterpriseOne data source, then this parameter value should specify an Oracle JD Edwards EnterpriseOne data source.

Scenario 6: Using an Oracle JD Edwards World fiscal calendar as the Enterprise calendar

In Oracle’s JD Edwards World, the Enterprise Calendar is referred to as the World Calendar.

Source System DAC Parameters for Oracle JD Edwards World to configure World Calendars:

- GBL_CALENDAR_ID: This parameter is used to select the World Calendar. It should be MCAL_CAL_ID. For example, GBL_CALENDAR_ID will be 'R' if the World Calendar id='R.'
- GBL_DATASOURCE_NUM_ID: If World Calendar is not a Generated Calendar, this value should be the DATASOURCE_NUM_ID of the source system from where the Calendar definition is taken. For example, if you have two data sources, Oracle and Oracle JD Edwards World, and the Global Calendar is from an Oracle JD Edwards World data source, then this parameter value should specify an Oracle JD Edwards World data source.

3.1.4.2.4 About Configuring Warehouse Generated Fiscal Calendars Oracle Business Intelligence Applications Release 7.9.6.3 supports the following types of generated calendars:

- 13 period calendars
- 4-4-5 calendars (and variants)

3.1.4.2.5 About the Calendar Context table (W_MCAL_CONTEXT_G) This table is used by Oracle Financial Analytics and Oracle Project Analytics facts to look up the calendar ID for a given ledger or OU (Operating Unit). This must be populated for the fact tables to get loaded correctly (the default Execution Plan in DAC does this).

In Oracle EBS, the project calendar and Organization (OU) relationship is sourced from PA_IMPLEMENTATIONS_ALL into W_MCAL_CONTEXT_G. The project calendar comes from the column PA_IMPLEMTATIONS_ALL.PERIOD_SET_NAME. The GL calendar used in project analytics comes from the column PA_IMPLENTATIONS_ALL.SET_OF_BOOKS_ID.

3.1.4.3 Notes on Configuring Calendars

When you set up calendars, note the following:

- The W_MCAL_CONFIG_G table controls how generated calendars are created.
- If generating the 4-4-5 or 13 period calendars, W_MCAL_CONFIG_G must have at least one row for the 4-4-5 period or 13 period. There is no entry needed in this table for Oracle EBS or PeopleSoft source calendars.

  Note: Oracle's JD Edwards EnterpriseOne and World Adapters for Financial Analytics do not support 4-4-5 period calendars.
- W_MCAL_WEEK_D will be populated only for the generated calendars (for example, 13 period or 4-4-5 type of calendars), and hence the W_DAY_D week Enterprise columns will be null for non-generated calendars (known as OLTP sourced fiscal calendars). W_ENT_WEEK_D will not be populated if a non-generated calendar is chosen as the Enterprise Calendar.
- For 13 period calendars, there is no concept of Quarter, therefore all Quarter columns in W_MCAL_WEEK_D, W_MCAL_PERIOD_D, W_MCAL_YEAR_D will be null. W_ENT_QTR_D will not be populated if a 13 period calendar is chosen as the Enterprise Calendar.
- Table 3–2 outlines columns in the W_MCAL_CONFIG_G table, which is loaded from the file_mcal_config_g.csv.

Table 3–2 Columns in Configuration Table W_MCAL_CONFIG_G

Column Name	Column Description
CALENDAR_ID	The ID of the calendar that is being configured. This is the primary key for this table.
CALENDAR_NAME	The name of the calendar that is being configured.
CALENDAR_CLASS	Automatically generated.
PERIOD_TYPE	The type of the calendar period that is being configured, such as 4-4-5.
CAL_ST_DT	The date from which the calendar generation begins.
CAL_END_DT	The date at which the calendar generation ends.
CAL_OFFSET	The offset which identifies the start date of the calendar. Valid Start Day and Offset values are: ■ Monday 0 ■ Tuesday 1 ■ Wednesday 2 ■ Thursday 3 ■ Friday -3 ■ Saturday -2 ■ Sunday -1
WEEK_ALLOCATION_ RULE	This parameter determines how weeks are allocated in the calendar that is being configured. For example, 4-4-5, 5-4-4, 4-5-4, or 13 period.
REFERENCE_DATE	The date (in MMDD format) that the fiscal year starts. For example, if an organization's fiscal year is from October to September, then the reference_date value should be 0929. The previous fiscal year must end between three days before and three days after the date specified by REFERENCE_DATE (that is, within the range (REFERENCE_DATE - 3) to (REFERENCE_DATE + 3). In other words, if a REFERENCE_ DATE is 0131 for 31st January, the previous fiscal year cannot exceed 3rd February.
Other standard columns	W_INSERT_DT, W_UPDATE_DT, TENANT_ID, X_CUSTOM, and so on.

- The following table outlines Task Level DAC Parameters needed for Generated Calendars.

Table 3–3 Task Level DAC Parameters Needed for Generated Calendars

DAC Parameter Name	DAC Parameter Description
$$13P_CALENDAR_ID	Task: SIL_TimeDimension_MCalWeek13Period. Required if you want to populate the 13 period type of calendar in your Data Warehouse. The value should be the CALENDAR_ID as defined in the W_MCAL_CONFIG_G table for the 13 period type of calendar.
$$445P_CALENDAR_ID	Task: SIL_TimeDimension_MCalWeek445. Required if you want to populate the 445 period type of calendar in your Data Warehouse. The value should be the CALENDAR_ID as defined in the W_MCAL_CONFIG_G table for the 445 period type of 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 regardless of whether the month actually has 31 days. If the month has a fewer 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 Oracle BI Repository and will not affect the ETL or reporting.
- There are some attributes on the W_DAY_D table that are not mapped in the Physical layer of the Oracle BI Repository. Therefore, before creating any new attribute in the repository, 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.
- By default, Oracle BI Applications can generate up to 65536 rows. If you need more than 65536 rows, you can increase the capacity to 262144 rows (718 years) by doing the following:

  1. Duplicate 'SIL_DayDimension_GenerateRows7'.
  2. Rename it 'SIL_DayDimension_GenerateRows8'.
  3. Run this immediately after 'SIL_DayDimension_GenerateRows7'.

3.1.4.4 How to Set Up the Gregorian Calendar Date Range

This task is a prerequisite for all types of calendar. This task loads the standard Gregorian calendar into the day dimension table W_DAY_D.

To set up the gregorian calendar:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab.
3. Query for the task SIL_DayDimension.
4. Display the Parameters tab in the lower pane.
5. Use the $$START_DATE and $$END_DATE parameters to specify the date range of the calendar that you want to generate.

Note: The tasks that load the day dimension will run as part of the Execution Plan for your Subject Areas. There are no separate Subject Areas for common dimensions; they are included in the core Subject Areas.

3.1.4.5 How to Configure PeopleSoft Summary Calendars

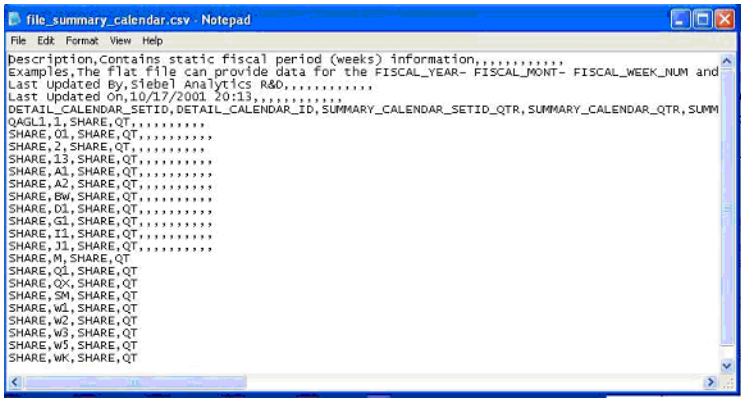
This configuration step must be performed only before loading PeopleSoft fiscal calendars. In PeopleSoft, a summary calendar is based on a detail calendar. Many summary calendars can be based off of single detail calendar. The file_summary_calendar.csv configuration file allows the association of a summary calendar to be used to determine fiscal quarters and year for PeopleSoft detail fiscal periods. For every detail calendar, input the summary CALENDAR_ID and SETID in the file file_summary_calendar.csv. The columns in this file are listed in Table 3–4.

Table 3–4 Fields in the file_summary_calendar.csv Configuration File

Column Name	Column Type	Column Description
DETAIL_CALENDAR_SETID	VARCHAR2(5)	SETID of the Detail Calendar being configured.
DETAIL_CALENDAR_ID	VARCHAR2(15)	CALENDAR_ID of the detail calendar being configured.
SUMMARY_CALENDAR_SETID_QTR	VARCHAR2(5)	SETID of the Quarterly Summary Calendar specified.
SUMMARY_CALENDAR_QTR	VARCHAR2(15)	CALENDAR_ID of the Quarterly Summary Calendar specified.
SUMMARY_CALENDAR_SETID_YEAR	VARCHAR2(5)	SETID of the Yearly Summary Calendar specified.
SUMMARY_CALENDAR_YEAR	VARCHAR2(15)	CALENDAR_ID of the Quarterly Calendar specified.
SUMMARY_CALENDAR_SETID_MONTH	VARCHAR2(5)	SETID of the Monthly Summary Calendar specified.
SUMMARY_CALENDAR_MONTH	VARCHAR2(15)	CALENDAR_ID of the Monthly Calendar specified.
SUMMARY_CALENDAR_SETID_HALF	VARCHAR2(5)	SETID of the Half Yearly Summary Calendar specified.
SUMMARY_CALENDAR_HALF	VARCHAR2(15)	CALENDAR_ID of the Half Yearly Calendar specified.
Other Standard columns	N/A	W_INSERT_DT, W_UPDATE_DT, TENANT_ID, X_CUSTOM, and so on.

Because a calendar is defined by its SETID and CALENDAR_ID in People Soft, for a particular Detail Calendar, the associated Summary calendar SETID and CALENDAR_ID are added. After updating this flat file file_summary_calendar.csv, run the ETL process. This flat file loads the table W_MCAL_PSFT_SUMP_CONFIG_G, which is used in the mapping that loads W_MCAL_PERIOD_D to resolve the Quarter Number for that Detail Period. You must make sure the summary calendar specified covers the date range included in the detail period. If this is not the case, then the MCAL_QTR column in W_MCAL_PERIOD_D will be null.

The following image shows example values in a file_summary_calendar.csv file.


file_summary_calendar.csv - Notepad
File Edit Format View Help

```
Description,Contains static fiscal period (weeks) information,,,,,,,,,,,,
Examples,The flat file can provide data for the FISCAL_YEAR- FISCAL_MONT- FISCAL_WEEK_NUM and
Last Updated By,Siebel Analytics R&D,,,,,,,,,,,,
Last Updated On,10/17/2001 20:13,,,,,,,,,,,,
DETAIL_CALENDAR_SETID,DETAIL_CALENDAR_ID,SUMMARY_CALENDAR_SETID_QTR,SUMMARY_CALENDAR_QTR,SUMM
QAGL1,1,SHARE,QT,,,,,,,,,,
SHARE,01,SHARE,QT,,,,,,,,,,
SHARE,2,SHARE,QT,,,,,,,,,,
SHARE,13,SHARE,QT,,,,,,,,,,
SHARE,A1,SHARE,QT,,,,,,,,,,
SHARE,A2,SHARE,QT,,,,,,,,,,
SHARE,BW,SHARE,QT,,,,,,,,,,
SHARE,D1,SHARE,QT,,,,,,,,,,
SHARE,G1,SHARE,QT,,,,,,,,,,
SHARE,I1,SHARE,QT,,,,,,,,,,
SHARE,J1,SHARE,QT,,,,,,,,,,
SHARE,M,SHARE,QT
SHARE,Q1,SHARE,QT
SHARE,QX,SHARE,QT
SHARE,SM,SHARE,QT
SHARE,W1,SHARE,QT
SHARE,W2,SHARE,QT
SHARE,W3,SHARE,QT
SHARE,W5,SHARE,QT
SHARE,WK,SHARE,QT
```

3.1.4.6 How to Set Up an Enterprise Calendar Using an Oracle EBS Source System

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab.
3. Set the value of $$GBL_CALENDAR_ID and $$GBL_DATSOURCE_NUM_ID as follows:

   - $$GBL_CALENDAR_ID: This parameter is used to select the Enterprise Calendar. Use the following EBS Source query to identify possible values for this parameter:

     ```
     SELECT period_set_name || '~'|| period_type FROM gl_periods;
     ```

     This query returns a list of choices for the $$GBL_CALENDAR_ID parameter. To select the Enterprise Calendar, use one of the returned values as the value for $$GBL_CALENDAR_ID. Note that the format of the parameter $$GBL_CALENDAR_ID is a concatenation of the PERIOD_SET_NAME and PERIOD_TYPE.

   - $$GBL_DATASOURCE_NUM_ID: This parameter is the data_source_num_id with which you have loaded the EBS calendars.

> **Note:** The tasks that load the Enterprise calendar will run as part of the Execution Plan for your Subject Area. There are no separate Subject Areas for common dimensions; they are included in the core Subject Areas.

3.1.4.7 How to Set Up an Enterprise Calendar Using a PeopleSoft Source System

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab.
3. Set the value of $$GBL_CALENDAR_ID and $$GBL_DATSOURCE_NUM_ID as follows:

   - $$GBL_CALENDAR_ID: This parameter is used to select the Enterprise Calendar. Use the following PeopleSoft Source query to identify possible values for this parameter:

     ```
     SELECT SETID|| '~'|| CALENDAR_ID FROM PS_CAL_DEFN_TBL;
     ```

     This query returns a list of choices for the $$GBL_CALENDAR_ID parameter. To select the Enterprise Calendar, use one of the returned values as the value for $$GBL_CALENDAR_ID. Note that the format of the parameter $$GBL_CALENDAR_ID is a concatenation of the SETID and CALENDAR_ID.

   - $$GBL_DATASOURCE_NUM_ID: This parameter is the data_source_num_id with which you have loaded the PeopleSoft calendars.

> **Note:** The tasks that load the Enterprise calendar will run as part of the Execution Plan for your Subject Area. There are no separate Subject Areas for common dimensions; they are included in the core Subject Areas.

3.1.4.8 How to Set Up an Enterprise Calendar Using an Oracle JD Edwards EnterpriseOne Source System

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab.
3. Set the value of $$GBL_CALENDAR_ID and $$GBL_DATSOURCE_NUM_ID as follows:
   - $$GBL_CALENDAR_ID: This parameter is used to select the Enterprise Calendar. Use the following Oracle JD Edwards EnterpriseOne Source query to identify possible values for this parameter:

     ```
     SELECT DISTINCT CDDTPN FROM F0008
     ```

     This query returns a list of choices for the $$GBL_CALENDAR_ID parameter. To select the Enterprise Calendar, use one of the returned values as the value for $$GBL_CALENDAR_ID. In the default configuration, the default calendar is "R."
   - $$GBL_DATASOURCE_NUM_ID: This parameter is the data_source_num_id with which you have loaded the Oracle JD Edwards calendars.

> **Note:** The tasks that load the Enterprise calendar will run as part of the Execution Plan for your Subject Area. There are no separate Subject Areas for common dimensions; they are included in the core Subject Areas.

3.1.4.9 How to Set Up an Enterprise Calendar Using an Oracle JD Edwards World Source System

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab.
3. Set the value of $$GBL_CALENDAR_ID and $$GBL_DATSOURCE_NUM_ID as follows:
   - $$GBL_CALENDAR_ID: This parameter is used to select the Enterprise Calendar. Use the following Oracle JD Edwards World Source query to identify possible values for this parameter:

     ```
     SELECT DISTINCT CDDTPN FROM F0008
     ```

     This query returns a list of choices for the $$GBL_CALENDAR_ID parameter. To select the Enterprise Calendar, use one of the returned values as the value for $$GBL_CALENDAR_ID. In the default configuration, the default calendar is "R."
   - $$GBL_DATASOURCE_NUM_ID: This parameter is the data_source_num_id with which you have loaded the Oracle JD Edwards calendars.

> **Note:** The tasks that load the Enterprise calendar will run as part of the Execution Plan for your Subject Area. There are no separate Subject Areas for common dimensions; they are included in the core Subject Areas.

3.1.4.10 How to Set Up a 13 Period Calendar

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab.
3. Set the value of $$GBL_CALENDAR_ID and $$GBL_DATSOURCE_NUM_ID as follows:
   - GBL_CALENDAR_ID: Should be the CALENDAR_ID of the Generated Calendar (4-4-5 or 13 period type of Calendars). By default the 4-4-5 calendar has a CALENDAR_ID of '10000' and the 13-period calendar has a CALENDAR_ID of '10001'.
   - GBL_DATASOURCE_NUM_ID: If Global Calendar is Generated Calendar: Should be the DATASOURCE_NUM_ID value of the OLAP (Data warehouse).
4. Using a text editor, edit the values in file_mcal_config_g.csv.
5. In DAC, set the value of 13P_CALENDAR_ID to 10001.

   Note: The task SIL_TimeDImension_McalWeek13Period will run as part of the Execution Plan for your Subject Area. There are no separate Subject Areas for common dimensions; they are included in the core Subject Areas.

3.1.4.11 How to Set Up a 4-4-5 Calendar

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab.
3. Set the value of $$GBL_CALENDAR_ID and $$GBL_DATSOURCE_NUM_ID as follows:
   - GBL_CALENDAR_ID: Should be the CALENDAR_ID of the Generated Calendar (4-4-5 or 13 period type of Calendars). By default the 4-4-5 calendar has a CALENDAR_ID of '10000' and the 13-period calendar has a CALENDAR_ID of '10001'.
   - GBL_DATASOURCE_NUM_ID: If Global Calendar is Generated Calendar: Should be the DATASOURCE_NUM_ID value of the OLAP (Data warehouse)
4. Using a text editor, edit the values in file_mcal_config_g.csv.
5. In DAC, set the value of 445P_CALENDAR_ID to 10000.

   Note: The task SIL_TimeDimension_McalWeek445 will run as part of the Execution Plan for your Subject Area. There are no separate Subject Areas for common dimensions; they are included in the core Subject Areas.

3.1.4.12 How to Use a Fiscal Calendar Loaded Through the Universal Adapter

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab.
3. Set the value of $$GBL_CALENDAR_ID and $$GBL_DATSOURCE_NUM_ID as follows:

- GBL_CALENDAR_ID: Should be the INTEGRATION_ID from the file_mcal_cal_d.csv file of the particular calendar which is defined as the Global Calendar.
- GBL_DATASOURCE_NUM_ID: If Global Calendar is not a Generated Calendar: Should be the DATASOURCE_NUM_ID of the source system from where the Calendar definition is taken, if its defined in the file_mcal_period_ds.csv file then that value should be taken or else as defined in DAC for Universal adapter.

4. Using a text editor, edit the values in the w_mcal_config_g.csv file.
5. Using a text editor, edit the values in the w_mcal_config_d.csv file.
6. If you are using Oracle Financial Analytics and Oracle Project Analytics to lookup the calendar ID for a given ledger OU, using a text editor, edit the values in the w_mcal_context_g.csv file.
7. Using a text editor, edit the values in the w_mcal_period_ds.csv file (if required).
8. In DAC, set the value of GBL_CALENDAR_ID and GBL_DATASOURCE_NUM_ID (as in scenario 4).

3.1.4.13 How to Use a PeopleSoft Summary Calendar

You must complete this task before running any Execution Plan that includes multiple calendar support and is sourcing fiscal calendars from a PeopleSoft source system.

To use a PeopleSoft Summary calendar:

1. Using a text editor, edit the values in summary_calendar.csv.
2. Run the Execution Plan for your Subject Area.

   The Execution Plan will automatically include the task that loads the PeopleSoft summary calendar configuration table.

3.1.4.14 Examples of Configuring the Universal Adapter

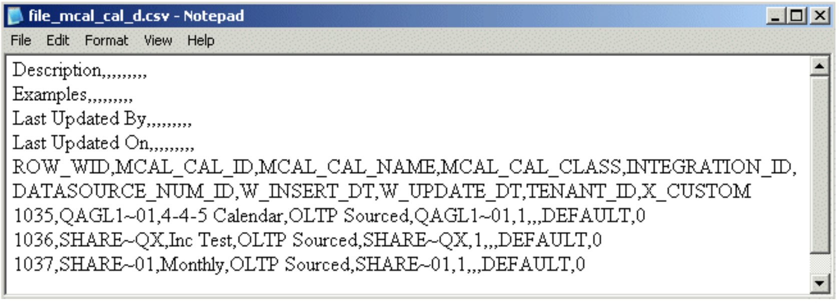
The Universal Adapter is provided to allow data from sources other than PeopleSoft, Oracle EBS, or Oracle JD Edwards, to be brought into the Multiple Calendar tables. The data from these sources must be brought in using the following CSV files:

- file_mcal_config_g.csv – loads W_MCAL_CONFIG_G.
- file_mcal_context_g.csv – loads W_MCAL_CONTEXT_G.
- file_mcal_cal_d.csv – loads W_MCAL_CAL_D.
- file_mcal_period_ds.csv – loads W_MCAL_PERIOD_DS (Staging Table).

3.1.4.14.1 Example CSV File Configurations This section includes example configuration settings for the Universal Adapter configuration files, and contains the following topics:

- "Example file_mcal_cal_d.csv Settings"
- "Example file_mcal_config_g.csv Settings"
- "Example file_mcal_context_g.csv Settings"
- "Example file_file_mcal_period_ds.csv Settings"

Example file_mcal_cal_d.csv Settings

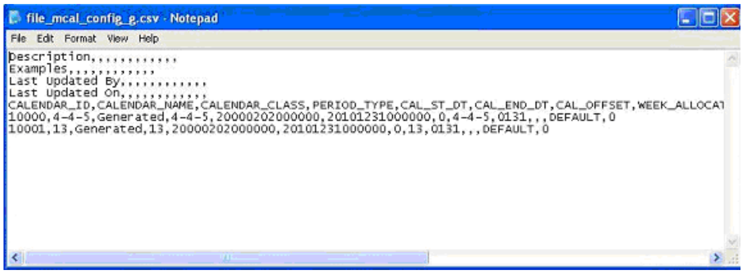


```
Description,,,,,,,,,
Examples,,,,,,,,,
Last Updated By,,,,,,,,,
Last Updated On,,,,,,,,,
ROW_WID,MCAL_CAL_ID,MCAL_CAL_NAME,MCAL_CAL_CLASS,INTEGRATION_ID,
DATASOURCE_NUM_ID,W_INSERT_DT,W_UPDATE_DT,TENANT_ID,X_CUSTOM
1035,QAGL1~01,4-4-5 Calendar,OLTP Sourced,QAGL1~01,1,,,DEFAULT,0
1036,SHARE~QX,Inc Test,OLTP Sourced,SHARE~QX,1,,,DEFAULT,0
1037,SHARE~01,Monthly,OLTP Sourced,SHARE~01,1,,,DEFAULT,0
```

Notes:

- Primary Key is the ROW_WID and must be unique.

Example file_mcal_config_g.csv Settings

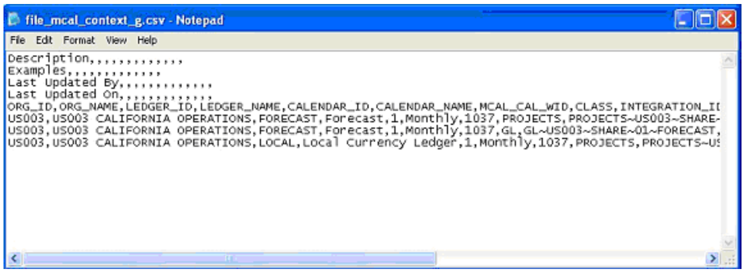


```
Description,,,,,,,,,,,,
Examples,,,,,,,,,,,,
Last Updated By,,,,,,,,,,,,
Last Updated On,,,,,,,,,,,,
CALENDAR_ID,CALENDAR_NAME,CALENDAR_CLASS,PERIOD_TYPE,CAL_ST_DT,CAL_END_DT,CAL_OFFSET,WEEK_ALLOCAT
10000,4-4-5,Generated,4-4-5,20000202000000,20101231000000,0,4-4-5,0131,,,DEFAULT,0
10001,13,Generated,13,20000202000000,20101231000000,0,13,0131,,,DEFAULT,0
```

Notes:

- To be used for Generated Calendars for all the adapters.
- CALENDAR_ID value for Generated Calendar is used in DAC task level parameter.
- DATE columns should be of the format YYYYMMDDHHMMSS (for example, 20000202000000 for 2nd February, 2000).
- CALENDAR_NAME for 13 Period type generated Calendar should be '13' or '13 Period'.
- REFERENCE_DATE should be of the format 'MMDD' (for example, 0131 for 31st January).

Example file_mcal_context_g.csv Settings

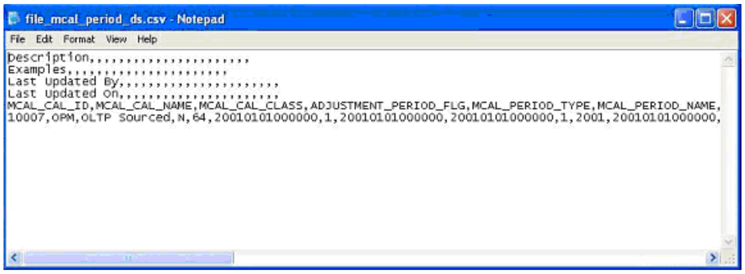


```
Description,,,,,,,,,,,,,
Examples,,,,,,,,,,,,,
Last Updated By,,,,,,,,,,,,,
Last Updated On,,,,,,,,,,,,,
ORG_ID,ORG_NAME,LEDGER_ID,LEDGER_NAME,CALENDAR_ID,CALENDAR_NAME,MCAL_CAL_WID,CLASS,INTEGRATION_ID
US003,US003 CALIFORNIA OPERATIONS,FORECAST,Forecast,1,Monthly,1037,PROJECTS,PROJECTS~US003~SHARE-
US003,US003 CALIFORNIA OPERATIONS,FORECAST,Forecast,1,Monthly,1037,GL,GL~US003~SHARE~01~FORECAST,
US003,US003 CALIFORNIA OPERATIONS,LOCAL,Local Currency Ledger,1,Monthly,1037,PROJECTS,PROJECTS~US
```

Notes:

- DATE columns should be of the format YYYYMMDDHHMMSS (for example, 20000202000000).

Example file_file_mcal_period_ds.csv Settings



```
Description,,,,,,,,,,,,,,,,,,,,,
Examples,,,,,,,,,,,,,,,,,,,,,
Last Updated By,,,,,,,,,,,,,,,,,,,,,
Last Updated On,,,,,,,,,,,,,,,,,,,,,
MCAL_CAL_ID,MCAL_CAL_NAME,MCAL_CAL_CLASS,ADJUSTMENT_PERIOD_FLG,MCAL_PERIOD_TYPE,MCAL_PERIOD_NAME,
10007,OPM,OLTP Sourced,N,64,20010101000000,1,20010101000000,20010101000000,1,2001,20010101000000,
```

Notes:

- DATE columns should be of the format YYYYMMDDHHMMSS (for example, 20000202000000).

3.1.4.15 How to Reload the Time Dimension Tables After the Data Warehouse Is Loaded

The data in time dimension tables is loaded once during the initial full load of the warehouse. Subsequently, the SIL_%_UpdateFlag mappings run during each incremental run 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. There are also other incremental tasks that run to populate the W_MCAL_% tables (Tables for Multiple Calendar Support). The range of dates for which data is loaded into the time dimension tables is determined by the parameters $$START_DATE and $$END_DATE.

Extending the data in the time dimension tables prior to the occurrence of $$END_DATE is recommended. If extension is not done prior to the occurrence of $$END_DATE, this will lead to failures, documented in technical note 1063484.1 on My Oracle Support.

In 7963, there is an Execution Plan provided for each container in DAC, to extend the Day Dimension when required. This execution plan includes all time dimension related tasks and initiates a full load ETL run of W_DAY_D and all the related time dimension tables, including the W_MCAL_% tables.

To extend the Day Dimension:

1. A subject area named 'Common dimension Extend Day Dimension' has been provided to facilitate the extension of the day dimension and related tables. By default, this subject area does not include the W_MCAL_% related tasks. If you have Multiple calendar support enabled in your Day Dimension, then include the configuration tag named 'Extend Day Dimension Multiple Calendar Support.' Then, assemble this Subject Area.

   Note: Only the subject area 'Common dimension Extend Day Dimension' should be included for extension. Do not include any other subject areas in this extension execution plan.

2. Build the corresponding execution plan (EP) for this subject area with same name.
3. Choose the Task SIL_DayDimension_XTND within the Execution Plan, and set the parameter values as follows:
   - $$START_DATE: Set this parameter to one day more than the old $$END_DATE value ($$END_DATE +1).

- $$END_DATE: Set this parameter to the new end date to which you want to extend.

4. Choose the SIL_TimeDimension_MCalPeriod_XTND within the Execution Plan and set the parameter values. Retain the original $$START_DATE, and choose a new $$END_DATE to which you want to extend. The $$END_DATE value should be same as the $$END_DATE value you chose for the task SIL_DayDimension_XTND in step 3. Note that this step is only needed if the configuration tag 'Extend Day Dimension Multiple Calendar Support' was included in step 1.

5. If necessary, change the following CSV files:

   - FILE_MCAL_CAL_D
   - FILE_MCAL_CONTEXT_G
   - FILE_MCAL_PERIOD_DS (if using the Universal Adapter)
   - FILE_MCAL_CONFIG_G

   If any of these files are used as sources, they must be extended for the new date ranges.

6. Run this Execution Plan and verify that the dates in W_D_DAY and related tables have been extended.

7. Rebuild all affected fact aggregates. For more information on which aggregates to rebuild, as well as detailed steps on how to rebuild them, refer to technical note 1063484.1 on My Oracle Support. Note that technical note 1063484.1 was written for pre-7.9.6.3 customers.

   Technical note 1063484 also provides steps for extending the time dimension and related tables. You can follow the steps in the technical note as an alternative to using the steps in this procedure.

3.1.4.16 ETL Process Flow Diagrams

The following diagrams show the ETL process flow and dependencies between the various Multiple Fiscal calendar tasks.

Figure 3–1 Process Flow Diagram for Time Dimension Load - Flows 1 & 2

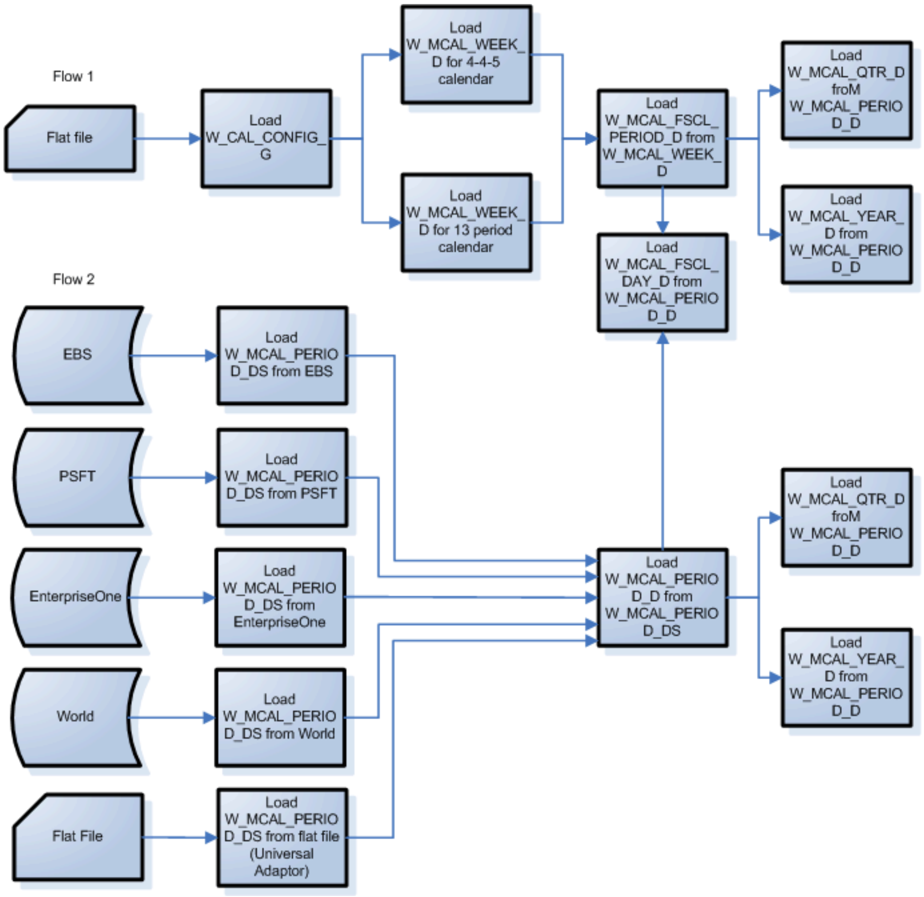
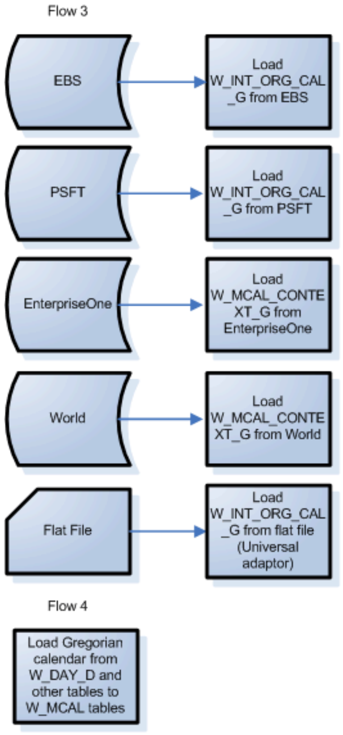

Flow 1
Flat file
Load W_CAL_CONFIG_G
Load W_MCAL_WEEK_D for 4-4-5 calendar
Load W_MCAL_WEEK_D for 13 period calendar
Load W_MCAL_FSCL_PERIOD_D from W_MCAL_WEEK_D
Load W_MCAL_FSCL_DAY_D from W_MCAL_PERIOD_D
Load W_MCAL_QTR_D froM W_MCAL_PERIOD_D
Load W_MCAL_YEAR_D from W_MCAL_PERIOD_D
Flow 2
EBS
PSFT
EnterpriseOne
World
Flat File
Load W_MCAL_PERIOD_DS from EBS
Load W_MCAL_PERIOD_DS from PSFT
Load W_MCAL_PERIOD_DS from EnterpriseOne
Load W_MCAL_PERIOD_DS from World
Load W_MCAL_PERIOD_DS from flat file (Universal Adaptor)
Load W_MCAL_PERIOD_D from W_MCAL_PERIOD_DS
Load W_MCAL_QTR_D froM W_MCAL_PERIOD_D
Load W_MCAL_YEAR_D from W_MCAL_PERIOD_D

Figure 3–2 Process Flow Diagram for Time Dimension Load - Flows 3 & 4


Flow 3
EBS
Load W_INT_ORG_CAL_G from EBS
PSFT
Load W_INT_ORG_CAL_G from PSFT
EnterpriseOne
Load W_MCAL_CONTEXT_G from EnterpriseOne
World
Load W_MCAL_CONTEXT_G from World
Flat File
Load W_INT_ORG_CAL_G from flat file (Universal adaptor)
Flow 4
Load Gregorian calendar from W_DAY_D and other tables to W_MCAL tables

3.1.5 About Mapping GL Accounts to Group Account Numbers for All Source Systems

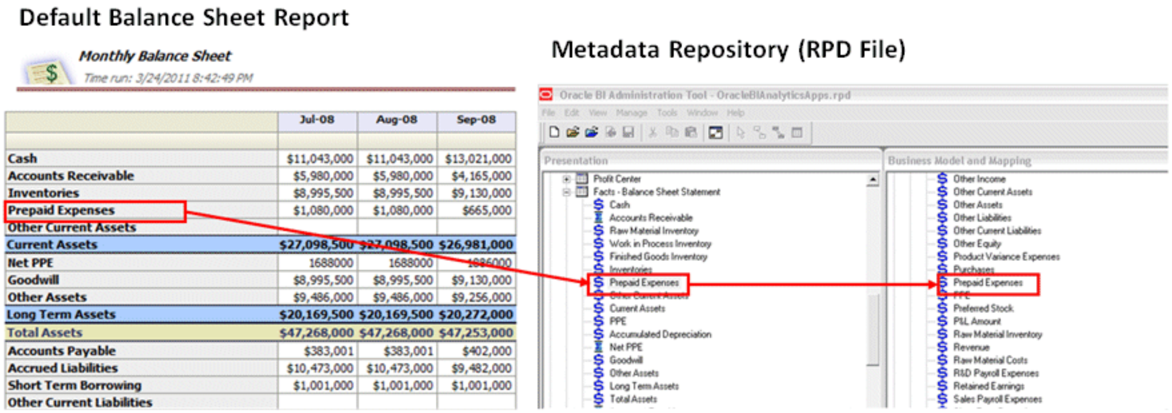
This section provides information about mapping General Ledger Accounts to Group Account Numbers.

To gain a better understanding of the purpose of Group Accounts, you can look at one of the pre-built reports in the Financial Analytics General Ledger Dashboard, the Monthly Balance Sheet report. The following example focuses on the Prepaid Expenses metric and explains how it is computed.

The Prepaid Expenses metric in the report comes directly from the "Facts - Balance Sheet Statement" table within the "Financials - GL Balance Sheet" subject area in the Presentation layer of the RPD file. The Presentation layer metric, in turn, comes from a corresponding item in the Business Model and Mapping layer.

Figure 3–3 shows the Monthly Balance Sheet Report and the corresponding metrics in the RPD file.

Figure 3–3 Monthly Balance Sheet Report and RPD Metrics



	Jul-08	Aug-08	Sep-08
Cash	$11,043,000	$11,043,000	$13,021,000
Accounts Receivable	$5,980,000	$5,980,000	$4,165,000
Inventories	$8,995,500	$8,995,500	$9,130,000
Prepaid Expenses	$1,080,000	$1,080,000	$665,000
Other Current Assets			
Current Assets	$27,098,500	$27,098,500	$26,981,000
Net PPE	1688000	1688000	1686000
Goodwill	$8,995,500	$8,995,500	$9,130,000
Other Assets	$9,486,000	$9,486,000	$9,256,000
Long Term Assets	$20,169,500	$20,169,500	$20,272,000
Total Assets	$47,268,000	$47,268,000	$47,253,000
Accounts Payable	$383,001	$383,001	$402,000
Accrued Liabilities	$10,473,000	$10,473,000	$9,482,000
Short Term Borrowing	$1,001,000	$1,001,000	$1,001,000
Other Current Liabilities			

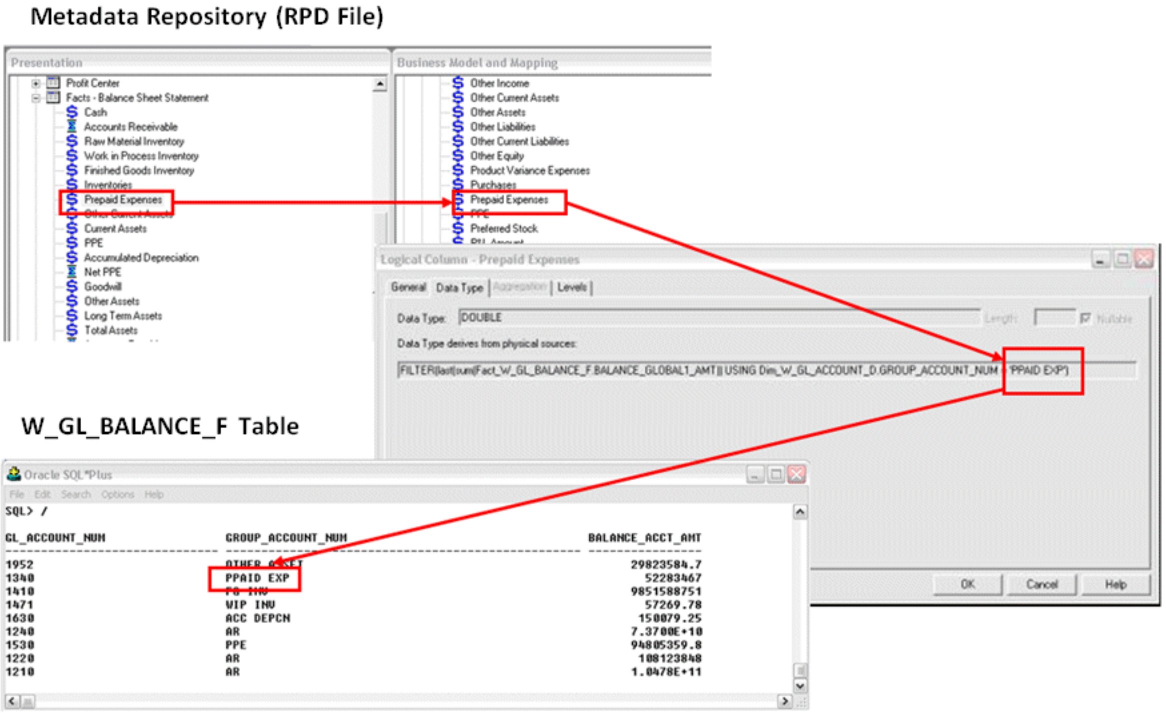
The fact table (the GL Balance fact) in the Business Model and Mapping layer joins with the GL Account Dimension and filters for the GL Accounts that belong to the Group Account Number 'PPAID EXP.'

When you look at the physical table W_GL_BALANCE_F, you can see that the metric is calculated in this case from one GL Account (1340), which is associated to the group account number 'PPAID EXP.' (Note that this example does not include other dimensionality, such as ledger, for the sake of simplicity).

To validate that account 1340 belongs to the Group Account Number PPAID EXP, you can view the configuration for the Group Account Number mapping in the file_group_acct_codes_*source*.csv file (for example, file_group_acct_codes_ora.csv for Oracle EBS).

Figure 3–4 shows the W_GL_BALANCE_F table and the corresponding RPD metrics.

Figure 3–4 W_GL_BALANCE_F Table and RPD Metrics



Data Type: DOUBLE

Data Type derives from physical sources:

FILTER(last(sum(Fact_W_GL_BALANCE_F.BALANCE_GLOBAL1_AMT)) USING Dim_W_GL_ACCOUNT_D.GROUP_ACCOUNT_NUM = 'PPAID EXP')

```
SQL> /

GL_ACCOUNT_NUM                 GROUP_ACCOUNT_NUM                                         BALANCE_ACCT_AMT
------------------------------ --------------------------------------------------------- ----------------
1952                           OTHER ASSET                                                     29823584.7
1340                           PPAID EXP                                                         52283467
1410                           FG INV                                                          9851588751
1471                           WIP INV                                                           57269.78
1630                           ACC DEPCN                                                        150079.25
1240                           AR                                                               7.3700E+10
1530                           PPE                                                             94805359.8
1220                           AR                                                               108123848
1210                           AR                                                               1.0478E+11
```

See also the following source-specific sections on this topic:

- Section 3.2.1.4, "About Mapping Oracle GL Natural Accounts to Group Account Numbers"
- Section 3.3.1.2, "About Mapping PeopleSoft GL Accounts to Group Account Numbers"
- Section 3.5.1.4, "About Mapping JD Edwards EnterpriseOne or JD Edwards World GL Accounts to Group Account Numbers"

3.1.6 Configuration Steps for Controlling Your Data Set for All Source Systems

This section contains additional configuration steps that apply to Oracle BI Applications deployed with any source system. It contains the following topics:

- Section 3.1.6.1, "How to Configure Data Source Numbers"

3.1.6.1 How to Configure Data Source Numbers

Data Source Numbers are identification numbers that are assigned to data sources so that the data can be identified in the data warehouse. For example, if you have a single Oracle EBS R12 data source, the value '9' is used by default. If you have two Oracle EBS R12 data sources that you want to differentiate in the data warehouse, you must assign a different data source number (for example, 30) to the second data source.

Data Source Numbers are stored in the DATASOURCE_NUM_ID column in the data warehouse.

Oracle BI Applications is installed with a number of pre-defined data source templates that you can edit to specify OLTP and OLAP data sources. The data source templates are pre-seeded with the recommended data source numbers, as specified in Table 3–5. If you specify a data source using one of the pre-defined templates, Oracle recommends that you use the data source number defined out-of-the-box. If you create a new data source without using one of the pre-defined templates, Oracle recommends that you specify the DATASOURCE_NUM_ID for that data source category as listed in Table 3–5. For example, if you specify an Oracle R12 EBS data source, Oracle recommends that you specify the DATASOURCE_NUM_ID value '9'. For instructions on how to specify data sources, see the section 'Setting Physical Data Sources' in *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

Table 3–5 shows the data sources that can be used with Oracle BI Applications and their default DATASOURCE_NUM_ID values.

Note: Not all of the data source types listed in Table 3-5 are supported by the current version of Oracle Business Intelligence Applications. For a list of supported source systems, refer to *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*.

Table 3–5 Data Sources and Associated DATASOURCE_NUM_ID Values

Data Source Name	Data Source Number
JDE_8_11_SP1_CTRL JDE_8_11_SP1_DATA JDE_8_11_SP1_DD JDE_8_11_SP1_FlatFile JDE_8_11_SP1_SYS	17

Table 3–5 (Cont.) Data Sources and Associated DATASOURCE_NUM_ID Values

Data Source Name	Data Source Number
JDE_8.12_CTRL JDE_8.12_DATA JDE_8.12_DD JDE_8_12_FlatFile JDE_8.12_SYS	15
JDE_9.0_CTRL JDE_9.0_DATA JDE_9.0_DD JDE_9_0_FlatFile JDE_9.0_SYS	25
JDEW_9_2_CTRL JDEW_9_2_DATA JDEW_9_2_DD JDEW_9_2_FlatFile JDEW_9_2_SYS	24
ORA_11_5_8 ORA_11_5_8_Flatfile	2
ORA_11_5_9 ORA_11_5_9_Flatfile	5
ORA_11_5_10 ORA_11_5_10_Flatfile	4
ORA_R12 ORA_R12_Flatfile	9
ORA_R1211 ORA_R1211_Flatfile	26
ORA_R1212 ORA_R1212_Flatfile	27
ORA_R1213 ORA_R1213_Flatfile	1000
PSFT_8_4_FINSCM PSFT_8_4_FlatFile	7
PSFT_8_8_FINSCM PSFT_8_8_FlatFile	8
PSFT_8_8_HCM	6
PSFT_8_9_FINSCM PSFT_8_9_FlatFile	11
PSFT_8_9_HCM PSFT_8_9_HCM_FlatFile	10

Table 3–5 (Cont.) Data Sources and Associated DATASOURCE_NUM_ID Values

Data Source Name	Data Source Number
PSFT_9_0_ELM PSFT_9_0_HCM PSFT_9_0_HCM_FlatFile	12
PSFT_9_0_FINSCM PSFT_9_0_FlatFile	13
PSFT_9_1_ELM PSFT_9_1_HCM PSFT_9_1_HCM_FlatFile	28
PSFT_9_1_FINSCM PSFT_9_1_FlatFile	29
SEBL_63 SEBL_6_3_Flatfile	1
SEBL_753 SEBL_7_5_3_Flatfile	1
SEBL_771 SEBL_7_7_1_Flatfile	1
SEBL_78 SEBL_7_8_Flatfile	1
SEBL_80 SEBL_8_0_Flatfile	1
SEBL_811 SEBL_8_1_1_Flatfile	1
SEBL_VERT_753 SEBL_VERT_7_5_3_Flatfile	1
SEBL_VERT_771 SEBL_VERT_7_7_1_Flatfile	1
SEBL_VERT_78 SEBL_VERT_7_8_Flatfile	1
SEBL_VERT_80 SEBL_VERT_80_Flatfile	1
SEBL_VERT_811 SEBL_VERT_8_1_1_Flatfile	1
UNIV	3

To configure Data Source Numbers:

1. In DAC, go to the Setup view, and display the Physical Data Sources tab.
2. Select the data source that you want to configure.
3. On the Edit subtab, use the **Data Source Number** field to set the value.
4. Click Save.

If you do choose to modify the data source number and if you implement Procurement and Spend Analytics Family of Products, it is mandatory that you perform the steps in section Section 4.2.2.1, "How to Configure the DAC Parameter for Purchase Cycle Line."

3.1.7 About Support for Back-Dated Change Scenarios

"Back-dated changes" refer to changes occurring in OLTP records after those records have already been processed and loaded into the warehouse. Position hierarchy represents the employee and supervisor reporting relationship that is maintained in the Oracle BI Applications data warehouse. In HR, it is the Supervisor Hierarchy; and in CRM, it is the Position Hierarchy. Position Hierarchy supports historical versions of the hierarchies with the current stamp on the current hierarchy rows. Position hierarchy is delivered with history tracking enabled, and the attributes that are tracked for Type 2 slowly changing are an employee's organization and supervisor. In addition, for CRM position hierarchy, a change in an employee's position is also tracked for Type 2 slowly changing.

Starting with release 7.9.6.2, back-dated changes and deletes in the OLTP that affect the supervisor and position hierarchy are processed accordingly by the ETL to reflect the retroactive changes in the warehouse.

3.1.8 Supported Back-Dated Change Scenarios That Affect Position Hierarchy

The Position Hierarchy is a column-flattened structure that is loaded from the Position dimension, which maintains the parent-child relationships. The scenarios described in this section illustrate the effects of how back-dated changes affect the position dimension which in turn will affect the Position Hierarchy.

In an OLTP source system such as Oracle EBS, corrections can be made to historical rows or the current rows. Corrections can be made to OLTP tables that provide the primary data source for the data warehouse or to tables that serve as auxiliary lookups in the ETL process. Corrections can affect the following attributes:

- Attributes that do not capture history, which are treated as a type 1 slowly changing dimension (SCD) in the data warehouse.
- Attributes that capture history, which are treated as a type 2 SCD in the data warehouse.

Position hierarchy is affected by back-dated changes in different tables that directly or indirectly influence the ETL process. For example, Division Name can be changed retroactively without corresponding employee assignment changes in the source OLTP system. This back-dated Division Name change will trigger historical updates to the Position Hierarchy.

There are three main types of OLTP tables that affect Supervisor or Position Hierarchy. The following examples represent the tables in a source OLTP system:

- Type A: The primary driving table that contains all historical assignment data, for example PER_ALL_ASSIGNMENTS_F. This table is the primary driver for the employee assignment history in the data warehouse.
- Type B: Auxiliary source table which keeps no history (Type 1 SCD) in OLTP but the warehouse tracks attribute changes as Type 2 SCD, for example HR_ORGANIZATION_UNITS. This table is the source for attributes that are history tracked in the warehouse.
- Type C: Auxiliary source table which keeps no history in OLTP and the warehouse also keeps no historical changes for the attribute (Type 1 SCD), such as

PER_JOBS. This table is the source for attributes that are not history tracked in the warehouse.

The functionality in Oracle BI Applications Release 7.9.6 assumes that:

- Position hierarchy tracks history as a type 2 SCD in the data warehouse
- Division Name tracks history as a type 2 slowly changing attribute in the data warehouse
- Person Name (a type 1 slowly changing attribute) does not track history in the data warehouse
- Job (a type 1 slowly changing attribute) does not track history in the warehouse

The following table contains an example of employee P1's assignment history in the warehouse before the back-dated changes:

Table 3–6 Example of Employee P1's Assignment History Before the Back-Dated Changes

Row	Person	Division	Manager	Start Date	End Date
Row 1	P1	CRM	Mgr1	2006	2008
Row 2	P1	My Division	Mgr2	2008	2009
Row 3	P1	My Division	Mgr3	2009	4712

The following sections describe the types of OLTP back-dated changes that affect the Position Hierarchy and the solutions provided in the warehouse to address the back-dated change scenarios.

3.1.8.1 Scenario 1

A division name changed as a correction to the current record, for example Division Name 'My Division' was renamed to 'My New Division' in 2010. The employee did not change divisions as a result of a transfer. This is a Type B change as the source table does not track Division Name history, which is a Type 2 slowly changing attribute in the warehouse.

Option 1 This option is treated as a Type 2 (SCD) change and will introduce a new Type 2 (SCD) row in the warehouse. This option is enabled if DAC parameter UPDATE_CORRECTIONS_FLG is set to 'N'.

Person	Division	Manager	Start Date	End Date
P1	CRM	Mgr1	2006	2008
P1	My Division	Mgr2	2008	2009
P1	My Division	Mgr3	2009	2010
P1	My New Division	Mgr3	2010	4712

Option 2 This option is treated as a correction and will change historical data only without a new Type 2 row in the warehouse. This option is enabled if DAC parameter UPDATE_CORRECTIONS_FLG is set to 'Y'.

Person	Division	Manager	Start Date	End Date
P1	CRM	Mgr1	2006	2008
P1	My New Division	Mgr2	2008	2009

Person	Division	Manager	Start Date	End Date
P1	My New Division	Mgr3	2009	4712

3.1.8.2 Scenario 1a

A division name changed as a correction to a historical record, for example Division Name 'My Division' was retroactively renamed to 'MD' 2008. The employee did not change divisions as a result of a transfer. This is also a Type B change, as Division Name history is not tracked in the source OLTP system but is tracked as a Type 2 SCD in the data warehouse.

Option Update the name change in the relevant history records.

Person	Division	Manager	Start Date	End Date
P1	CRM	Mgr1	2006	2008
P1	MD	Mgr2	2008	2009
P1	My New Division	Mgr3	2009	4712

3.1.8.3 Scenario 2

Auxiliary table change: Job information is changed in the source OLTP system that is referred to in the historical data in the warehouse, for example Job name is changed to lower case. This is a Type C change where neither the OLTP nor the data warehouse track historical changes.

Option New Job is propagated to all historical rows.

Person	Division	Manager	Start Date	End Date
P1	CRM	Mgr1, job2	2006	2008
P1	My Division	Mgr2, job2	2008	2009
P1	My Division	Mgr3, job2	2009	4712

3.1.8.4 Scenario 3

An employee changed assignment divisions as a result of a transfer, for example Employee P1 transferred to division 'GRC' reporting to Mgr4 in 2010. This is a Type A change, as the change occurs on a main OLTP driving table with history tracking.

Option The data warehouse will insert a new row for tracking the employee as having a new manager. This is the standard case.

Person	Division	Manager	Start Date	End Date
P1	CRM	Mgr1	2006	2008
P1	My Division	Mgr2	2008	2009
P1	My Division	Mgr3	2009	2010
P1	GRC	Mgr4	2010	4712

3.1.8.5 Scenario 3a

This is a variation of Scenario 3, for example Employee's transfer from 'CRM' to 'My Division' actually occurred in 2007, not in 2008. This is a correction to the historical assignment record. The back-dated change is to the effective_from and effective_to date of the driving OLTP history table.

Option 1 Update the historical data in the warehouse. This does not require fact table updates. This option is enabled if DAC parameter UPDATE_CORRECTIONS_FLG is set to 'Y'.

Row	Person	Division	Manager	Start Date	End Date
1	P1	CRM	Mgr1	2006	2007
2	P1	My Division	Mgr2	2007	2009
3	P1	My Division	Mgr3	2009	4712

Option 2 This option introduces a new warehouse row to track changes. This option is enabled if DAC parameter UPDATE_CORRECTIONS_FLG is set to 'N'.

Row	Person	Division	Manager	Start Date	End Date
1	P1	CRM	Mgr1	2006	2007
4	P1	My Division	Mgr2	2007	2008 (NEW)
2	P1	My Division	Mgr2	2008	2009
3	P1	My Division	Mgr3	2009	4712

Prior to the back-dated change, the fact table had some transactions pointing to the row 1 of the hierarchy table and some with foreign keys to the row 2. Fact rows with foreign key to row 1 will continue to have the same foreign key or the foreign key will be updated to match either row 2 or row 4 depending on the transaction date.

3.1.8.6 Scenario 3b

A back-dated change occurs in the source OLTP system that results in the splitting of the record. For example, the employee changed managers from Mgr1 to Mgr5 in 2007. In the source OLTP system, the original assignment record with Mgr1 has a new end date of 2007 and a new record was added to assign the employee to the new manager Mgr5 in 2007.

Option The warehouse will respond to the OLTP source change as described in the following table.

Row	Person	Division	Manager	Start Date	End Date
1	P1	CRM	Mgr1	2006	2007
4	P1	CRM	Mgr5	2007	2008 (NEW)
2	P1	My Division	Mgr2	2008	2009
3	P1	My Division	Mgr3	2009	4712

Fact rows with foreign keys to the row 1 will continue to have the same foreign key or the foreign key will be updated to row 4 depending on the transaction date.

3.1.8.7 Scenario 3c

A back-dated change gets cascaded to all the records since a particular date, for example Mgr2 in row 3 should be Mgr4 and also change managers for all future rows since 2008. The employee's current assignment records are as follows in the data warehouse:

Row	Person	Division	Manager	Start Date	End Date
1	P1	CRM	Mgr1	2006	2007
2	P1	CRM	Mgr5	2007	2008 (NEW)
3	P1	My Division	Mgr2	2008	2009
4	P1	My Division	Mgr3	2009	4712

Option Update the historical and current records in the warehouse:

Row	Person	Division	Manager	Start Date	End Date
1	P1	CRM	Mgr1	2006	2007
2	P1	CRM	Mgr5	2007	2008 (NEW)
3	P1	My Division	Mgr4	2008	2009
4	P1	My Division	Mgr4	2009	4712

3.1.8.8 Scenario 3d

A back-dated change is a row delete in OLTP. Row 2 is deleted in OLTP and row 1 is updated to the end date of 2009.

Option Transactions will not be deleted in the warehouse, but rows will be updated to reflect the changed attributes in the time range, including changes initiated by auxiliary tables. This will void fact foreign key updates.

Row	Person	Division	Manager	Start Date	End Date
1	P1	CRM	Mgr1	2006	2008
2	P1	CRM	Mgr1	2008	2009 (NEW)
3	P1	My Division	Mgr3	2009	4712

3.2 Oracle EBS-Specific Configuration Steps

This section contains configuration steps that apply to Oracle BI Applications deployed with Oracle EBS source systems.

This section contains the following topics:

- Section 3.2.1, "Configuration Required Before a Full Load for Oracle EBS"
- Section 3.2.2, "Configuration Steps for Controlling Your Data Set for Oracle EBS"

3.2.1 Configuration Required Before a Full Load for Oracle EBS

This section contains configuration steps required before a full data load that apply to Oracle BI Applications deployed with Oracle EBS source systems. It contains the following topics:

- Section 3.2.1.1, "Configuration of Product Hierarchy (Except for GL, HR Modules)"
- Section 3.2.1.2, "How to Assign UNSPSC Codes to Products"
- Section 3.2.1.3, "How to Configure the Master Inventory Organization in Product Dimension Extract for Oracle Adapters (Except for GL & HR Modules)"
- Section 3.2.1.4, "About Mapping Oracle GL Natural Accounts to Group Account Numbers"
- Section 3.2.1.5, "How to Make Corrections to the Group Account Number Configuration"
- Section 3.2.1.6, "About Configuring GL Account Hierarchies"

3.2.1.1 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.

Hierarchy is defined by using segment values in the flexfield structure of Oracle EBS. Although Oracle EBS allows up to 20 segments, Oracle Business Intelligence Applications supports only 10 level of hierarchy (segment). These hierarchies are obtained from the Oracle EBS Category Set and Category tables. The default product hierarchies are configured out-of-the-box as follows:

- Purchasing Category Set is assigned a value of 2, using the DAC source system parameter PROD_CAT_SET_ID1.
- Inventory Category Set is assigned a value of 27, using the DAC source system parameter INV_PROD_CAT_SET_ID1.

You configure these parameters and create new PROD_CAT_SET*n* and INVPROD_CAT_SET*n* parameters in DAC, based on the Oracle EBS Category Set IDs that you want to load into the Oracle Business Analytics Warehouse.

To configure your product category sets, do the following:

- Identify the category sets you want to report (for more information, see Section 3.2.1.1.1, "How to Identify Category Sets from Oracle EBS").
- Configure the DAC parameters for product hierarchy (for more information, see Section 3.2.1.1.2, "How to Configure DAC Source System Parameters for Product Hierarchy").

3.2.1.1.1 How to Identify Category Sets from Oracle EBS

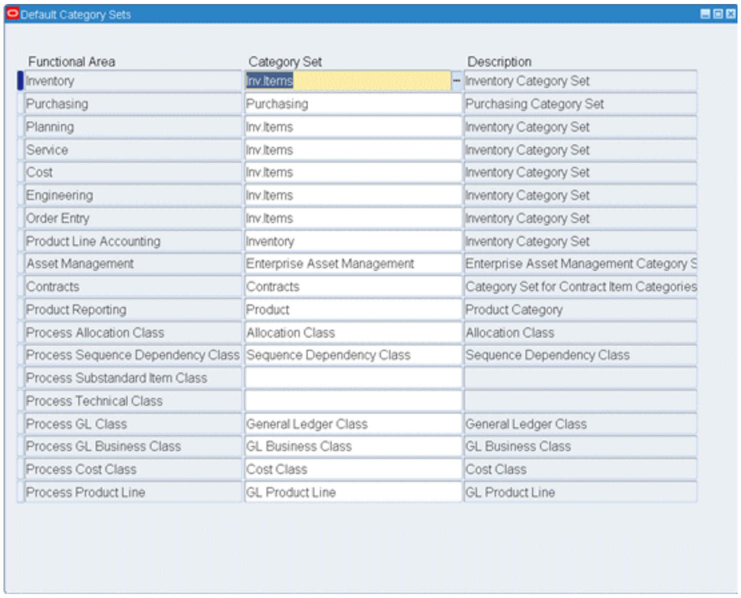
These steps are part of the task in Section 3.2.1.1, "Configuration of Product Hierarchy (Except for GL, HR Modules)."

To find out the category sets that your organization is using:

1. Log in to the Oracle EBS instance.
2. Click Setup, then Items, then Categories, and then Default Category Sets.
3. Look for the functional area Inventory and place the cursor in the Category Set field.

   Figure 3–5 shows an example of the Default Category Sets screen.

Figure 3–5 Default Category Sets Screen in Oracle EBS



Functional Area	Category Set	Description
Inventory	Inv.Items	Inventory Category Set
Purchasing	Purchasing	Purchasing Category Set
Planning	Inv.Items	Inventory Category Set
Service	Inv.Items	Inventory Category Set
Cost	Inv.Items	Inventory Category Set
Engineering	Inv.Items	Inventory Category Set
Order Entry	Inv.Items	Inventory Category Set
Product Line Accounting	Inventory	Inventory Category Set
Asset Management	Enterprise Asset Management	Enterprise Asset Management Category S
Contracts	Contracts	Category Set for Contract Item Categories
Product Reporting	Product	Product Category
Process Allocation Class	Allocation Class	Allocation Class
Process Sequence Dependency Class	Sequence Dependency Class	Sequence Dependency Class
Process Substandard Item Class		
Process Technical Class		
Process GL Class	General Ledger Class	General Ledger Class
Process GL Business Class	GL Business Class	GL Business Class
Process Cost Class	Cost Class	Cost Class
Process Product Line	GL Product Line	GL Product Line

4. Choose Help, then Diagnostics, then Examine, and specify the user password for the application.

5. Click the Field LOV button and select CATEGORY_SET_ID, and record the value.

6. Repeat steps 3-5 for the Purchasing functional area.

3.2.1.1.2 How to Configure DAC Source System Parameters for Product Hierarchy These steps are part of the task in Section 3.2.1.1, "Configuration of Product Hierarchy (Except for GL, HR Modules)".

To configure the DAC source system parameters for product hierarchy:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.

2. Display the Source System Parameters tab.

3. Locate the INV_PROD_CAT_SET_ID1 and PROD_CAT_SET_ID1 parameters, and for each parameter do the following:

   a. Use the Value field to specify a category set value (that is, replace the default Category Set ID 2 or 27 with a new value).

      Note: The value for INV_PROD_CAT_SET_ID parameter should be set to the appropriate Inventory Category Set value. The value for the PROD_CAT_SET_ID parameter should be set to the appropriate Purchase Category Set value.

      You can configure up to ten Product Category Hierarchies and ten Inventory Category Hierarchies.

4. If you want to set up multiple product hierarchies, specify values for the additional INV_PROD_CAT_SET_ID*n* and PROD_CAT_SET_ID*n* parameters that you want to deploy.

5. Click Save.

Notes:

- 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.
- For information about adding or customizing Product Category Hierarchies, see Section 16.4, "How to Set Up Product Category Hierarchies."

3.2.1.2 How to Assign UNSPSC Codes to Products

This section explains how to assign United Nations Standard Products and Services Code (UNSPSC) codes to products and commodities. The United Nations Standard Products and Services Code® (UNSPSC®) provides an open, global multi-sector standard for efficient, accurate classification of products and services.

You can assign UNSPSC codes to your Products by adding the UNSPSC codes to the file_unspsc.csv file. The codes are then loaded into the W_PROD_CAT_DH table by the SDE_UNSPSC process. You must then perform manual classification by mapping the codes in FILE_ITEM_TO_UNSPSC.csv to provide the link between the product (D) and product category (DH) tables, using the load process PLP_ItemToUNSPSC_Classification.

Note the following about UNSPSC codes:

- You must buy your own UNSPSC data directly from UNSPSC.org, and then map the data in file_unspsc.csv for loading into Oracle BI Applications.
- The file_unspsc.csv file is used by the SDE_UNSPSC process and contains sample data. Replace the sample data with your own UNSPSC data. Make sure to perform this task before performing the manual classification steps described later in this section.
- The UNSPSC_CODE column must be unique for each record in the file. This column is specific to Oracle BI Applications and does not come from the purchased UNSPSC data. You can map this column to the COMMODITY_CODE column.
- The FUNCTION_CODE and FUNCTION_NAME columns are additional columns that might be included in the purchased UNSPSC data. If you do not see these columns in the purchased UNSPSC data, you can map these columns to the COMMODITY_CODE and COMMODITY_NAME columns, respectively. These columns are used to copy the levels in the hierarchy down to the base level.
- Ensure that the DATASOURCE_NUM_ID in the file is the same as that defined for the Oracle EBS Adapter. The default values are '9' for Oracle EBS R12 and '4' for Oracle EBS R11510.

To assign UNSPSC codes to products:

1. To identify the Products used in your deployment, run a select statement on the W_PRODUCT_D table.

   For example, you might use Oracle SQLDeveloper to run the following SQL command:

   ```
   SELECT INTEGRATION_ID, PRODUCT_NAME, PART_NUMBER FROM W_PRODUCT_D;
   ```

   Note: In this example SQL statement, the INTEGRATION_ID is the product that needs classification. The PRODUCT_NAME and PART_NUM are additional attributes to assist in classifying the UNSPSC Codes.

2. Add the product IDs and UNSPSC codes to the FILE_ITEM_TO_UNSPSC.csv file.

The FILE_ITEM_TO_UNSPSC.csv file is located in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).

Populate the PRODUCT_ID field with the INTEGRATION_ID values you retrieved in Step 1. Populate the UNSPSC_CODE field with the UNSPSC Code values you purchased from UNSPSC.org. Also, ensure that the DATASOURCE_NUM_ID is the same as your Oracle EBS Adapter.

3. In Informatica PowerCenter Designer, run the \PLP\PLP_ItemToUNSPSC_Classification workflow to update the rows in the W_PRODUCT_D table.

3.2.1.3 How to Configure the Master Inventory Organization in Product Dimension Extract for Oracle Adapters (Except for GL & HR Modules)

In Oracle 11i and R12.x 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' has been enabled for configuration of this Master Organization based on the configuration in the OLTP. By default, the organization ID (that is set by the $$MASTER_ORG parameter in DAC) is set to 204. This organization ID 204 must be changed based on the individual implementation of OLTP in your deployment.

> **Note:** This ETL implementation supports the best practice prescribed by Oracle for the creation of Single Master Organization for defining the Product master. You can assign Multiple MASTER Organizations also under the same parameter by providing a comma-separated string of Organization codes (for example, '204','458').

To set the Master Inventory Organization in the Product dimension extract:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab, and query for the SDE_ORA_ProductDimension task.
3. In the lower pane, display the Parameters tab, and locate the $$MASTER_ORG parameter.
4. Use the **Value** field to specify an appropriate value.

   For example, 204. To define multiple master organizations, specify a comma-separated list of organization codes in single quotation marks (for example, '204', '301').
5. Save your changes.

3.2.1.4 About Mapping Oracle GL Natural Accounts to Group Account Numbers

This section explains how to map Oracle General Ledger Accounts to Group Account Numbers, and includes the following topics:

- An overview of setting up Oracle General Ledger Accounts - see Section 3.2.1.4.1, "Overview of Mapping Oracle GL Accounts to Group Account Numbers."
- A description of how to map Oracle General Ledger Accounts to Group Account Numbers - see Section 3.2.1.4.2, "How to Map Oracle GL Account Numbers to Group Account Numbers."

- An example that explains how to add Group Account Number metrics to the Oracle BI Repository - see Section 3.2.1.4.3, "Example of Adding Group Account Number Metrics to the Oracle BI Repository."

> **Note:** It is critical that the GL account numbers are mapped to the group account numbers (or domain values) because the metrics in the GL reporting layer use these values. For a list of domain values for GL account numbers, see *Oracle Business Analytics Warehouse Data Model Reference*.

See also Section 3.1.5, "About Mapping GL Accounts to Group Account Numbers for All Source Systems" for additional information.

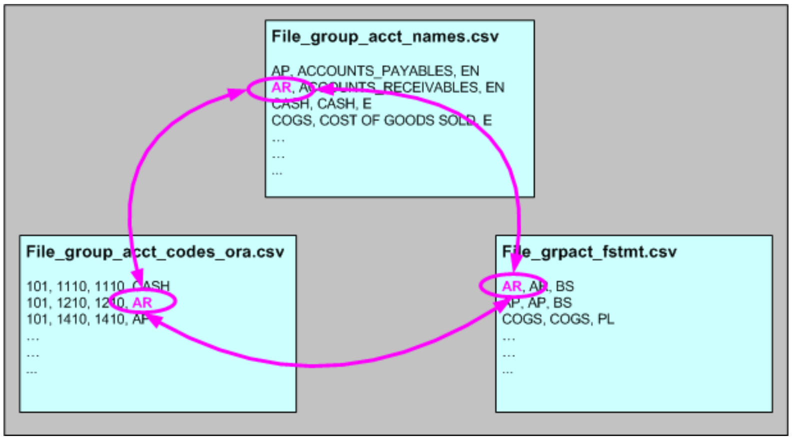
3.2.1.4.1 Overview of Mapping Oracle GL Accounts to Group Account Numbers Group Account Number Configuration is an important step in the configuration of Financial Analytics, because it determines the accuracy of the majority of metrics in the General Ledger and Profitability module. Group Accounts in combination with Financial Statement Codes are also leveraged in the GL reconciliation process, to ensure that subledger data reconciles with GL journal entries. This topic is discussed in more detail later in this section.

You set up General Ledger accounts using the following configuration files that are located in the $PMServer\SrcFiles directory:

- file_group_acct_names.csv - this file specifies the group account names and their corresponding group account codes.
- file_group_acct_codes_ora.csv - this file maps General Ledger accounts to group account codes.
- file_grpact_fstmt.csv - this file maps Financial Statement Item Codes to group account codes.

Before you load your data, you must ensure that the account values are mapped consistently across these three configuration files. Figure 3–6 shows how the GROUP_ACCOUNT_NUM field in file_group_acct_names.csv must map to the GROUP_ACCT_NUM field in file_group_acct_codes_ora.csv and the GROUP_ACCT_NUM field in file_grpact_fstmt.csv.

Figure 3–6 Configuration Files for Configuring Group Accounts for Oracle EBS



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 of the General Ledger accounts (for example, cash account, payroll account). Refer to the GROUP_ACCOUNT_NUM column in the file_group_acct_names.csv file for group account codes that you can use (for example, AR, CASH, COGS). 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, Profit and Loss, Cash Flow statements).

The logic for assigning the group accounts is located in the file_group_acct_codes_ora.csv file. Table 3–7 shows an example configuration of the file_group_acct_codes_ora.csv file.

Table 3–7 Example Configuration of file_group_acct_codes_ora.csv File

CHART OF ACCOUNTS 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 Table 3–7, 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 (that is, CA). 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.csv file. You can then assign GL accounts to the new group of account numbers in the file_group_acct_codes_ora.csv file.

You must also 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. Table 3–8 shows the Financial Statement Item codes to which Group Account Numbers must map, and their associated base fact tables.

Table 3–8 Financial Statement Item Codes and Associated Base Fact Tables

Financial Statement Item Codes	Base Fact Tables
AP	AP base fact (W_AP_XACT_F)
AR	AR base fact (W_AR_XACT_F)
COGS	Cost of Goods Sold base fact (W_GL_COGS_F)
REVENUE	Revenue base fact (W_GL_REVN_F)
TAX	Tax base fact (W_TAX_XACT_F)[1]

Table 3–8 (Cont.) Financial Statement Item Codes and Associated Base Fact Tables

Financial Statement Item Codes	Base Fact Tables
OTHERS	GL Journal base fact (W_GL_OTHER_F)

[1] Oracle EBS adapters for Financial Analytics do not support the Tax base fact (W_TAX_XACT_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. This association is important to perform GL reconciliation and to ensure the subledger data reconciles with GL journal entries. It is possible that after an invoice has been transferred to GL, a GL user might adjust that invoice in GL. In this scenario, it is important to ensure that the adjustment amount is reflected in the subledger base fact as well as balance tables. To determine such subledger transactions in GL, the reconciliation process uses Financial Statement Item codes.

Financial Statement Item codes are internal codes used by the ETL process to process the GL journal records during the GL reconciliation process against the subledgers. When the ETL 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 ETL 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 the ETL program 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 GROUP_ACCOUNT_NUM column of the file_group_acct_names.csv file.

3.2.1.4.2 How to Map Oracle GL Account Numbers to Group Account Numbers

This section explains how to map Oracle General Ledger Account Numbers to Group Account Numbers.

> **Note:** If you add new Group Account Numbers to the file_group_acct_codes_ora.csv file, you must also add metrics to the Oracle BI repository. See Section 3.2.1.4.3, "Example of Adding Group Account Number Metrics to the Oracle BI Repository" for more information.

To map Oracle GL account numbers to group account numbers:

1. Open the file_group_acct_codes_ora.csv file with a text editor in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).
2. For each Oracle GL account number that you want to map, create a new row in the file containing the following fields:

Field Name	Description
CHART OF ACCOUNTS ID	The ID of the GL chart of account.

Field Name	Description
FROM ACCT	The lower limit of the natural account range. This is based on the natural account segment of your GL accounts.
TO ACCT	The higher limit of the natural account range. This is based on the natural account segment of your GL accounts.
GROUP_ACCT_NUM	This field denotes the group account number of the Oracle General Ledger account, as specified in the file_group_acct_names.csv file. For example, 'AP' for Accounts Payables, 'CASH' for cash account, 'GEN PAYROLL' for payroll account, and so on.

For example:

```
101, 1110, 1110, CASH
101, 1210, 1210, AR
101, 1220, 1220, AR
```

> **Note:** You can optionally remove the unused rows in the CSV file.

3. Ensure that the values that you specify in the file_group_acct_codes_ora.csv file are consistent with the values that are specified in the file_group_acct_names.csv file and the file_grpact_fstmt.csv file.

   See Figure 3–6, "Configuration Files for Configuring Group Accounts for Oracle EBS" for an example.

4. Save and close the CSV file.

3.2.1.4.3 Example of Adding Group Account Number Metrics to the Oracle BI Repository If you add new Group Account Numbers to the file_group_acct_codes_ora.csv file (as described in Section 3.2.1.4.2, "How to Map Oracle GL Account Numbers to Group Account Numbers"), you must also add metrics to the Oracle BI repository to expose the new Group Account Numbers. This example illustrates how you can add Group Account Number metrics using the Oracle BI Administration Tool.

This example assumes that you have a new Group Account Number called 'Payroll', and you want to add a new metric to the Presentation layer called 'Payroll Expense'.

To add a new metric in the logical table Fact – Fins – GL Other Posted Transaction:

1. Using the Administration Tool, open OracleBIAnalyticsApps.rpd.

   The OracleBIAnalyticsApps.rpd file is located at:

   ```
   ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_
   obisn\repository
   ```

2. In the Business Model and Mapping layer:

   a. Create a logical column named 'Payroll Expense' in the logical table 'Fact – Fins – GL Other Posted Transaction'.

      For example, right-click the Core\Fact - Fins - GL Other Posted Transaction\ object and choose **New Object**, then **Logical Column**, to display the Logical Column dialog. Specify Payroll Expense in the **Name** field.

   b. Display the Aggregation tab, and then choose 'Sum' in the **Default aggregation rule** drop-down list.

c. Click **OK** to save the details and close the dialog.

d. Expand the Core\Fact - Fins - GL Other Posted Transaction\Sources\ folder and double click the Fact_W_GL_OTHER_GRPACCT_FSCLYR_A source to display the Logical Table Source dialog.

e. Display the Column Mapping tab.

f. Select **Show unmapped columns**.

g. Locate the Payroll Expense expression, and click the Expression Builder button to open Expression Builder.

h. Use Expression Builder to specify the following SQL statement:

```
CASE WHEN "Oracle Data Warehouse"."Catalog"."dbo".
"Dim_W_GL_GROUP_ACCOUNT_D"."GROUP_ACCOUNT_NUM" = 'PAYROLL'
THEN "Oracle Data Warehouse"."Catalog"."dbo".
"Fact_Agg_W_GL_OTHER_GRPACCT_FSCLYR_A"."OTHER_GLOBAL1_AMT" ELSE NULL END
```

The case condition refers to the new Group Account Number 'Payroll' and uses it as a filter.

i. Repeat steps d to h for each Logical Table Source. Modify the expression in step h appropriately for each LTS by using the appropriate fact table that corresponds to the Logical Table Source.

Steps d to h must be repeated for each Logical Table Source because in this example, there are multiple Logical Table Sources for fact table and aggregation tables in this logical table. Modify the expression in step h appropriately for each Logical Table Source by using the appropriate fact table to which it corresponds.

3. Save the details.

4. To expose the new repository objects in end users' dashboards and reports, drag the new objects from the Business Model and Mapping layer to an appropriate folder in the Presentation layer.

To add a new metric in the logical table Fact – Fins – GL Balance:

1. Using the Administration Tool, open OracleBIAnalyticsApps.rpd.

   The OracleBIAnalyticsApps.rpd file is located at:

```
ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_
obisn\repository
```

2. In the Business Model and Mapping layer:

a. Create a logical column named 'Payroll Expense' in logical table 'Fact – Fins – GL Balance'.

For example, right-click the Core\Fact – Fins – GL Balance object and choose **New Object**, then **Logical Column**, to display the Logical Column dialog. Specify Payroll Expense in the **Name** field.

b. In the Column Source tab, select **Derived from existing columns using an expression**.

c. Click the Expression Builder button to display Expression Builder.

d. Use Express Builder to specify the following SQL statement:

```
FILTER("Core"."Fact - Fins - GL Balance"."Activity Amount" USING
"Core"."Dim - GL Account"."Group Account Number" = 'Payroll')
```

The case condition refers to the new Group Account Number 'Payroll'.

3. Save the details.
4. To expose the new repository objects in end users' dashboards and reports, drag the new objects from the Business Model and Mapping layer to an appropriate folder in the Presentation layer.

3.2.1.5 How to Make Corrections to the Group Account Number Configuration

Note: Refer to Section 3.2.1.4, "About Mapping Oracle GL Natural Accounts to Group Account Numbers" for general concepts about group account number and Financial Statement Item code.

When a user maps a GL natural account to an incorrect group account number, incorrect accounting entries might be inserted into the fact table. For example, the natural account 1210 is mistakenly classified under 'AR' group account number when it should be classified under 'AP' group account number. When this happens, the ETL program will charge all the GL journal lines to account 1210 and try to reconcile these GL journal lines against subledger accounting records in the AR fact table (W_AR_XACT_F). Since these GL journal lines did not come from AR, the ETL program will not be able to find the corresponding subledger accounting records for these GL journal lines. In this case, the ETL program will insert 'Manual' records into the AR fact table because it thinks that these GL journal lines are 'Manual' journal entries created directly in the GL system charging to the AR accounts. This entire process is called the GL reconciliation process.

Too revert these 'Manual' entries in the AR fact table, use the 'Group Account Number Clean Up' program provided in Oracle BI 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, the ETL program will try to look for the corresponding subledger accounting records in the AP fact (W_AP_XACT_F), provided that you have reassigned the natural account 1210 to the 'AP' group account number in the file_group_acct_codes_ora.csv file.

> **Note:** The Financials – Group Account Number Clean Up subject area should be executed only when you need to correct group account number assignments. This subject area should not be included in the standard Financials execution plan. You must run it separately.

To do group account correction:

1. Correct the mapping of GL natural account to the group account in the input CSV file called file_group_acct_codes_ora.csv.

   For example, before correction, a CSV file has the following values:

   CHART OF ACCOUNTS ID = 101

   FROM ACCT = 1110

   TO ACCT = 1110

   GROUP_ACCT_NUM = CASH

   After correction, if the account '1210' originally belonged to the 'AP' group account number 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_NUM = AR

2. In DAC, do the following:

   a. Go to the Design view, and select the appropriate custom container from the drop-down list.

   b. Display the Subject Areas tab.

   c. Query for the 'Financials – General Ledger' Subject Area.

   d. Display the Configuration Tags subtab, and verify and which of the following Configuration Tags is marked as 'Inactive':

      - Financials – Calculate GL Balance

      - Oracle – Extract GL Balance

      By default, 'Financials – Calculate GL Balance' should be marked as inactive.

   e. Query for the 'Financials – Group Account Number Clean Up' Subject Area and do the following:

      - If in the preceding step, the configuration tag 'Financials – Calculate GL Balance' is marked as 'Inactive', then 'Financials – Calculate GL Balance Reverse' should be marked as 'Inactive' as well.

      - If in the preceding step, 'Oracle – Extract GL Balance' is marked as 'Inactive', then 'Financials – Calculate GL Balance Reverse' should be marked as active (that is, the **Inactive** check box should be cleared).

3. If you need to make any change in the preceding step, then you must display the Subject Areas tab and reassemble the Subject Area named 'Financials – Group Account Number Clean Up'.

4. Rebuild and execute the appropriate Execution Plan that you are using to load the Subject Area named 'Financials – Group Account Number Clean Up'.

   For example, depending on which version of Oracle EBS you have, you might be using one of the following out-of-the-box Execution Plans such as 'Financials – Group Account Number Clean Up R12' or 'Financials – Group Account Number Clean Up R1211'.

3.2.1.6 About Configuring GL Account Hierarchies

You must configure GL Account Hierarchies if you are deploying one of the following applications:

- Oracle Financial Analytics
- Oracle Procurement and Spend Analytics
- Oracle Supply Chain and Order Management Analytics

The are two ways to configure general ledger account hierarchies:

- Using General Ledger Accounting Flexfield Value Sets Definitions

  For instructions, see Section 3.2.1.6.1, "How to Configure GL Segments and Segment Hierarchies Using Flexfield Value Set Definitions."

- Using the Financial Statement Generator (FSG) Report Definition

This option is available only if your source system is Oracle EBS. For instructions, see Section 3.2.1.6.2, "How to Configure GL Account Hierarchies Using Financial Statement Generator (FSG) Report Definition (for Oracle EBS)."

Whichever method you choose to set up GL account hierarchies, you store the hierarchy information in the W_HIERARCHY_D table.

As an example, the hierarchy for a GL account called 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.

Figure 3–7 shows an example of this hierarchy for US Acct.

Figure 3–7 Example of Hierarchy for US Acct

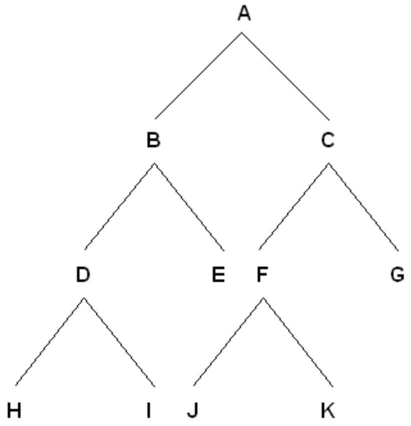


Table 3–9 shows how the hierarchy for US Acct would be stored in the W_HIERARCHY_D table.

Table 3–9 Example of Hierarchy for 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

3.2.1.6.1 How to Configure GL Segments and Segment Hierarchies Using Flexfield Value Set Definitions You must configure GL account hierarchies if you are deploying Oracle Financial Analytics, Oracle Procurement and Spend Analytics, and Oracle Supply Chain and Order Management Analytics.

Thirty segments are supported in which you can store accounting flexfields. Flexfields are flexible enough to support complex data configurations. For example:

- You can store data in any segment.
- You can use more or fewer segments per chart of accounts, as required.
- You can specify multiple segments for the same chart of accounts.

Example of Data Configuration for a Chart of Accounts

A single company might have a US chart of accounts and an APAC chart of accounts, with the following data configuration:

Table 3–10 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 5
Line of Business (LOB)	Stores in segment 1	Stores in segment 4

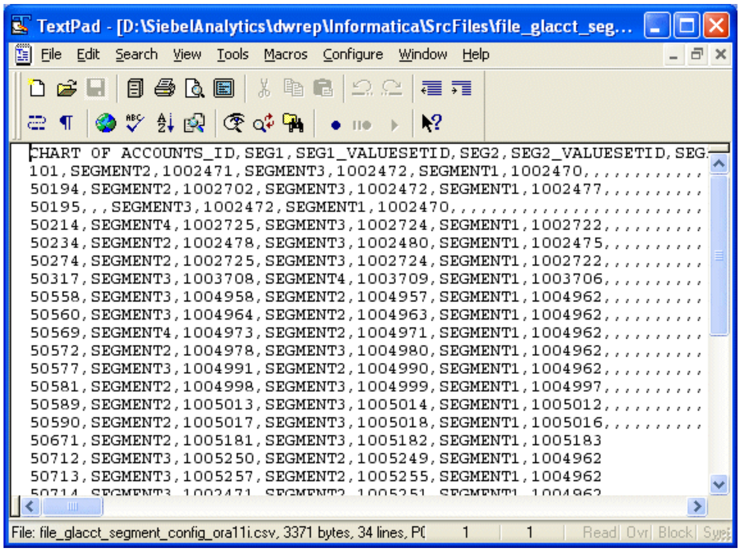
This example shows that in Chart of Account 4256, 'Company' is stored in the segment 3 column in the Oracle EBS table GL_CODE_COMBINATIONS_ALL. In Chart of Account COA4257, 'Company' is stored in the segment 1 column in GL_CODE_COMBINATIONS_ALL table. The objective of this configuration file is to ensure that when segment information is extracted into the data 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 the segment 1 column in W_GL_ACCOUNT_D; and Cost Center segments from COA 4256 and 4257 in the segment 2 column in W_GL_ACCOUNT_D, and so forth.

How to Set Up the GL Segment Configuration File

Before you run the ETL process for GL accounts, you must specify the segments that you want to analyze using the ETL configuration file named file_glacct_segment_config_*source_system*.csv, located in $PMServer\SrcFiles (for example, *INFA_HOME*\server\infa_shared\SrcFiles).

Figure 3–8 file_glacct_segment_config_ora.csv File Opened in a Text Editor



TextPad - [D:\SiebelAnalytics\dwrep\Informatica\SrcFiles\file_glacct_seg...
File Edit Search View Tools Macros Configure Window Help

```
CHART OF ACCOUNTS_ID,SEG1,SEG1_VALUESETID,SEG2,SEG2_VALUESETID,SEG
101,SEGMENT2,1002471,SEGMENT3,1002472,SEGMENT1,1002470,,,,,,,,,,,,
50194,SEGMENT2,1002702,SEGMENT3,1002472,SEGMENT1,1002477,,,,,,,,,,
50195,,,SEGMENT3,1002472,SEGMENT1,1002470,,,,,,,,,,,,,,,,,,,,,,,,
50214,SEGMENT4,1002725,SEGMENT3,1002724,SEGMENT1,1002722,,,,,,,,,,
50234,SEGMENT2,1002478,SEGMENT3,1002480,SEGMENT1,1002475,,,,,,,,,,
50274,SEGMENT2,1002725,SEGMENT3,1002724,SEGMENT1,1002722,,,,,,,,,,
50317,SEGMENT3,1003708,SEGMENT4,1003709,SEGMENT1,1003706,,,,,,,,,,
50558,SEGMENT3,1004958,SEGMENT2,1004957,SEGMENT1,1004962,,,,,,,,,,
50560,SEGMENT3,1004964,SEGMENT2,1004963,SEGMENT1,1004962,,,,,,,,,,
50569,SEGMENT4,1004973,SEGMENT2,1004971,SEGMENT1,1004962,,,,,,,,,,
50572,SEGMENT2,1004978,SEGMENT3,1004980,SEGMENT1,1004962,,,,,,,,,,
50577,SEGMENT3,1004991,SEGMENT2,1004990,SEGMENT1,1004962,,,,,,,,,,
50581,SEGMENT2,1004998,SEGMENT3,1004999,SEGMENT1,1004997,,,,,,,,,,
50589,SEGMENT2,1005013,SEGMENT3,1005014,SEGMENT1,1005012,,,,,,,,,,
50590,SEGMENT2,1005017,SEGMENT3,1005018,SEGMENT1,1005016,,,,,,,,,,
50671,SEGMENT2,1005181,SEGMENT3,1005182,SEGMENT1,1005183
50712,SEGMENT3,1005250,SEGMENT2,1005249,SEGMENT1,1004962
50713,SEGMENT3,1005257,SEGMENT2,1005255,SEGMENT1,1004962
50714,SEGMENT3,1002471,SEGMENT2,1005251,SEGMENT1,1004962
```

File: file_glacct_segment_config_ora11i.csv, 3371 bytes, 34 lines, PC
1
1
Read Ovr Block Sync

In file_glacct_segment_config_*source_system*.csv, you must specify the segments of the same type in the same column. For example, you might store all Cost Center segments from all charts of accounts in one column, and all Company segments from all charts 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 Company segments from all COAs in ACCOUNT_SEG1_CODE column in W_GL_ACCOUNT_D.

- 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.

- Store GL account balances at Company and Cost Center level instead of at GL Code Combination level, in W_GL_BALANCE_A (where you store GL account balances at aggregated level).

  Note: The ETL logic for populating W_GL_BALANCE_A depends on the last row in the file where CHART OF ACCOUNTS_ID = AGGREGATION. Even if you are not using GL Account Hierarchies using ValueSet definitions, you still need to have this line in the CSV file. The out-of-the-box setting for this value is Y for SEG1 through SEG6 (that is, "AGGREGATION,Y, ,Y, ,Y, ,Y, ,Y, ,Y").

- Set up the GL Segment configuration file for U.S. Federal Financial Analytics.

  For U.S. Federal Financials Analytics, the first two segments are reserved for Fund and Program segments respectively. So, to use one or both of these, configure file_glacct_segment_config_*source_system*.csv in this particular order:

  1. Put your Fund segment column name in the 'SEGMENT1' column in the CSV file.

2. Put your Program segment column name in the 'SEGMENT2' column in the CSV file.

If you do not have any one of these reserved segments in your source system, leave that particular segment empty in the CSV file. To configure any other segments that you may have other than Fund and Program, configure these segments starting from SEGMENT3.

Steps to Configure GL Segments and Hierarchies Using Value Set Definitions

1. Configure file_glacct_segment_config_*source_system*.csv, as follows:

   a. Navigate to $PMServer\SrcFiles (for example, *INFA_HOME*\server\infa_shared\SrcFiles).

   b. Open file_glacct_segment_config_*source_system*.csv in a text editor.

   c. Follow the steps in "How to Set Up the GL Segment Configuration File" to configure the file.

2. Perform the following steps in DAC if you are extracting the Value Set Hierarchies for your GL Segments. If you are planning to use only the GL Segments without the value set hierarchies, skip this step and proceed to step 3.

   a. Navigate to the Subject Areas tab, and query for 'Financials – General Ledger'.

   b. In the Configuration Tags subtab, do the following:

   c. Query for the tag 'Oracle – Extract Value Set Hierarchies', and make sure the **Inactive** check box is not selected, and query for the tag 'Oracle – Extract FSG Hierarchies', and make sure the **Inactive** check box is selected.

   d. Click Assemble to reassemble the Subject Area.

   e. Navigate to the Execution Plans tab in the Execute view, and rebuild all Execution Plans that contain the 'Financials – General Ledger' Subject Area.

      For instructions on building Execution Plans, see *Oracle Business Intelligence Data Warehouse Administration Console Guide*.

   f. Run the Execution Plan for General Ledger Accounts.

3. Make the following changes in the RPD metadata using the Oracle BI Administration Tool. The metadata contains multiple logical tables that represent each GL Segment, such as Dim – GL Segment1, Dim – GL Segment2, and so on. Because all these logical tables are mapped to the same physical table, W_GL_SEGMENT_D, a filter should be specified in the logical table source of these logical tables in order to restrain the output of the logical table to represent only that particular segment. You must set the filter on the physical column SEGMENT_LOV_ID to the ValueSet IDs that are applicable for that particular segment. The list of the ValueSet IDs would be the same as the ValueSet IDs you configured in the CSV file mentioned above.

   To specify a filter in the Business Model and Mapping layer of the Oracle BI Repository, do the following:

   a. Using the Administration Tool, open OracleBIAnalyticsApps.rpd.

      The OracleBIAnalyticsApps.rpd file is located at:

      ```
      ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_
      obisn\repository
      ```

   b. Expand each logical table, for example, Dim - GL Segment1, and open the logical table source under it.

c. Display the Content tab.

d. In the 'Use this WHERE clause…' box, apply a filter on the corresponding physical table alias of W_GL_SEGMENT_D. For example:

```
"Oracle Data Warehouse"."Catalog"."dbo"."Dim_W_GL_SEGMENT_D_
Segment1"."SEGMENT_LOV_ID" IN (comma seperated valuesetids)
```

Enter all ValueSet IDs, separated by commas, that correspond to this segment.

4. Oracle Financial Analytics supports up to 30 segments in the GL Account dimension, and by default delivers 10 GL Segment dimensions in the RPD. If you need more than 10 GL Segments, perform the following steps to add new segments:

a. In the Physical layer:

Create a new physical alias of W_GL_SEGMENT_D as "Dim_W_GL_SEGMENT_D_SegmentXX". To do this, right-click the physical table W_GL_SEGMENT_D and select **New Object** and then **Alias**. Name the new alias Dim_W_GL_SEGMENT_D_SegmentXX. Similarly, create a new alias of W_HIERARCHY_D as "Dim_W_HIERARCHY_D_SegmentXX".

In the Physical Diagram, create a Physical Foreign Key between Dim_W_HIERARCHY_D_SegmentXX and Dim_W_GL_SEGMENT_D_SegmentXX similar to the one between Dim_W_HIERARCHY_D_Segment1 and Dim_W_GL_SEGMENT_D_Segment1. The direction of the foreign key should be from W_HIERACHY_D to W_GL_SEGMENT_D; for example, on a '0/1':N cardinality join, W_HIERARCHY_D will be on the '0/1' side and W_GL_SEGMENT_D will be on the 'N' side. See *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* for more information about how to create physical foreign key joins.

Similarly, create a physical foreign key join between Dim_W_GL_SEGMENT_D_SegmentXX and Dim_W_GL_ACCOUNT_D, with W_GL_SEGMENT_D on the '1' side and W_GL_ACCOUNT_D on the 'N' side.

Save your changes.

b. In the Business Model and Mapping layer:

Create a new logical table "Dim - GL SegmentXX" similar to "Dim – GL Segment1". This logical table should have a logical table source that is mapped to the physical tables created above (for example, it will have both Dim_W_GL_SEGMENT_D_SegmentXX and Dim_W_HIERARCHY_D_SegmentXX). This logical table should also have all attributes similar to "Dim – GL Segment1" properly mapped to the respective physical tables, Dim_W_GL_SEGMENT_D_SegmentXX and Dim_W_HIERARCHY_D_SegmentXX.

In the Business Model Diagram, create a logical join from "Dim – GL SegmentXX" to all the relevant logical fact tables similar to "Dim – GL Segment1", with the GL Segment Dimension Logical table on the '0/1' side and the logical fact table on the 'N' side. To see all the relevant logical fact tables, first include Dim – GL Segment1 on the Business Model Diagram, and then right-click that table and select **Add Direct Joins**.

Add the content filter in the logical table source of "Dim – GL SegmentXX" as described in the previous step.

Create a dimension by right-clicking "Dim – GL SegmentXX", and select **Create Dimension**. Rename this to "GL SegmentXX". Make sure the drill-down structure is similar to "GL Segment1". If you are not sure how to do this, follow these steps: By default, the dimension will have two levels: the Grand Total Level and the Detail Level. Rename these levels to "All" and "Detail" respectively. Right-click the "All" level and select "New Object" and then "Child Level". Name this level as Level1. Similarly, create a level under Level1 and name it as Level2. Repeat this process until you have Level19 under Level18. Now drag the "Detail" level under "Level19" so that "Detail" is the last level of the hierarchy. Now, from the new logical table Dim - GL SegmentXX, drag Level1 Code and Level1 Name attributes to the "Level1" level of the hierarchy. Now go the Properties of the Level and from the Keys tab, create two new keys, one for Level1 Code and one for Level1 Name. When you create keys, make sure the 'Use for Drilldown' option is OFF for Level1 Code and ON for Level1 Name. Also, make sure the Primary Key drop-down is set to Level1 Code. Now, proceed similarly to all 19 levels by dragging the appropriate two attributes to the appropriate levels and create the keys similar to above. For the "Detail" level, drag the Level20 Code and Level20 Name attributes into it and create the keys similar to above.

Open the Logical Table Source of the new logical table created, Dim - GL SegmentXX. Set the Aggregation Content in the Content tab by setting the Logical Level to "Detail" level of the "GL SegmentXX" dimension/hierarchy created above.

Similarly, set the aggregation content to all relevant fact logical table sources. Open all Logical Table Sources of all the logical fact tables that are relevant to the new logical table one at a time. Go to the Content tab. If you see that it is set to the "Detail" level of other GL Segment Dimensions such as GL Segment1, GL Segment2, and so on, then set it to the "Detail" level of the "GL Segment XX" dimension/hierarchy created above. If not, skip that logical table source and go to the next one.

c. Drag your new "Dim - GL Segment XX" dimensions into the appropriate subject areas in the Presentation layer. Typically, you can expose these GL Segment dimensions in all subject areas where the GL Account dimension is exposed. You can also find all appropriate subject areas by right-clicking Dim – GL Segment1 and select Query Related Objects, then selecting Presentation, and then selecting Subject Area.

d. Save your changes and check global consistency.

5. Each GL Segment denotes a certain meaningful ValueSet(s) in your OLTP. To clearly identify each segment in the report, you can rename the presentation table "GL SegmentX", logical dimension "GL SegmentX", and logical table "Dim - GL SegmentX" according to its own meaning.

   For example, if you populate Product segment into Segment1, you can rename logical table "Dim - GL Segment1" as "Dim – GL Segment Product" or any other appropriate name and then rename the tables in the Presentation layer accordingly.

3.2.1.6.2 How to Configure GL Account Hierarchies Using Financial Statement Generator (FSG) Report Definition (for Oracle EBS) You must configure GL account hierarchies if you are deploying Oracle Financial Analytics, Oracle Procurement and Spend Analytics, and Oracle Supply Chain and Order Management Analytics. For information about the two ways you can configure GL account hierarchies, see Section 3.2.1.6, "About Configuring GL Account Hierarchies."

If you need to define GL account hierarchies based on multiple segments within a chart of accounts, you can use the Oracle FSG report definition in Oracle EBS to define them.

You should first use the Oracle FSG form to define a row set or a column set, then Oracle BI Applications will extract the row set or column set definition and convert them into hierarchies.

Oracle FSG hierarchies are extracted from following Oracle 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.

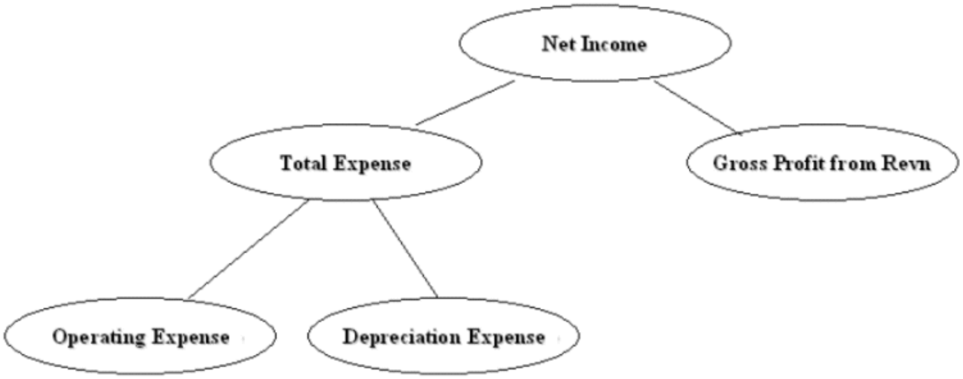
- RG_REPORT_CALCULATIONS

  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 following diagram 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 following diagram shows how an income state is derived from a hierarchy:



This hierarchy would be converted into a flattened hierarchy and stored in W_ HIERARCHY_D in the following format:

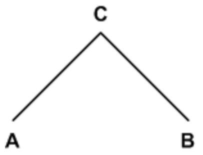
Table 3–11 Example of Flattened Hierarchy Stored in W_HIERARCHY_D

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 through the GL 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.

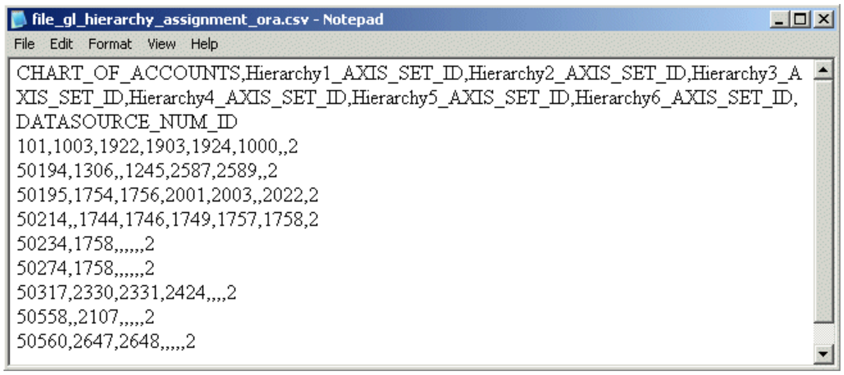
Note: Mathematical operators, such as '+', '-', '*', '/' (addition, subtraction, multiplication, division, and so on) are not extracted from the FSG definitions. For example, both A + B = C and A - B = C would give the same hierarchy, with a node C having two child nodes A and B, as shown in the following diagram:



About the ETL Process for Oracle FSG Report

Before you run the ETL process for GL accounts, you must specify the hierarchies that you want to reference. To specify the FSG hierarchies that you want to reference, use the file file_gl_hierarchy_assignment_ora.csv, which is located in $PMServer\SrcFiles (for example, *INFA_HOME*\server\infa_shared\SrcFiles).

Figure 3–9 file_gl_hierarchy_assignment_ora.csv File Opened in a Text Editor


file_gl_hierarchy_assignment_ora.csv - Notepad
File Edit Format View Help

```
CHART_OF_ACCOUNTS,Hierarchy1_AXIS_SET_ID,Hierarchy2_AXIS_SET_ID,Hierarchy3_A
XIS_SET_ID,Hierarchy4_AXIS_SET_ID,Hierarchy5_AXIS_SET_ID,Hierarchy6_AXIS_SET_ID,
DATASOURCE_NUM_ID
101,1003,1922,1903,1924,1000,,2
50194,1306,,1245,2587,2589,,2
50195,1754,1756,2001,2003,,2022,2
50214,,1744,1746,1749,1757,1758,2
50234,1758,,,,,2
50274,1758,,,,,2
50317,2330,2331,2424,,,,2
50558,,2107,,,,,2
50560,2647,2648,,,,,2
```

In this file, for each chart of accounts, you can specify six FSG hierarchies, using axis_set_id, which is a column from the 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 must use the DATASOURCE_NUM_ID value to distinguish between them.

For example, suppose you have an income statement FSG report and a balance sheet FSG report and you want to input both of their hierarchy structures into the data warehouse. Oracle BI Applications assumes that both reports are derived from the same set of GL accounts with CHART_OF_ACCOUNTS=101. The axis_set_id of the income statement is 1001, and for the 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, assume you want to associate their HIER1 column (in GL_ACCOUNT_D) with the income statement hierarchy structure and HIER3 column with balance sheet hierarchy structure.

In this case, you would add one row into file_gl_hierarchy_assignment_ora.csv with fields set as follows:

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 the income statement hierarchy row ID in W_HIERARCHY_D, and HIER3 column will be the foreign key to the balance sheet hierarchy row ID in W_HIERARCHY_D.

Note: Axis_set_id must be specified in file_gl_hierarchy_assignment_ora.csv for Financial Analytics to load the hierarchies.

To set up hierarchies with FSG Report Definition:

1. Configure the file_gl_hierarchy_assignment_ora.csv file to specify the hierarchies you want to reference for each CHART_OF_ACCOUNTS.
   a. Navigate to $PMServer\SrcFiles (for example, *INFA_HOME*\server\infa_shared\SrcFiles).
   b. Open file_gl_hierarchy_assignment_ora.csv in a text editor.
   c. Specify the segments you want to analyze.
   d. Save and close the file.
2. In DAC, do the following:
   a. Go to the Design view, and select the appropriate custom container from the drop-down list.
   b. Click the Subject Areas tab, and query for all Financials subject areas.

c. For each of these subject areas, in the Configuration Tags subtab, check to see whether the subject area has the following two configuration tags:

  * Oracle - Extract FSG Hierarchies

  * Oracle - Extract Value Set Hierarchies

  For subject areas with these two tags, do the following:

  * Activate the 'Oracle – Extract FSG Hierarchies' tag by deselecting the **Inactive** check box.

  * Deactivate the 'Oracle – Extract Value Set Hierarchies' tag by selecting the **Inactive** check box.

  * Click Assemble to reassemble the subject area.

d. Navigate to the Execution Plans tab in the Execute view, and rebuild all execution plans where any of the subject areas assembled in the previous step are used.

  For instructions on building Execution Plans, see *Oracle Business Intelligence Data Warehouse Administration Console Guide*.

e. Run the Execution Plan for General Ledger Accounts.

3. Using the Oracle BI Administration Tool, in the Physical layer of the Oracle BI Repository, 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.

4. Using the Oracle BI Administration Tool, in the Physical layer of the Oracle BI Repository, 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

5. Using the Oracle BI Administration Tool, in the Business Model and Mapping layer of the Oracle BI Repository, 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 the 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, refer to the preinstalled example logical table Dim - FSG Hierarchy 1 in the Oracle Business Analytics Warehouse

6. Using the Oracle BI Administration Tool, in the Business Model and Mapping layer of the Oracle BI Repository, create a new dimension based on the logical table that you created in the previous step.

   Refer to 'FSG Hierarchy 1' as an example.

7. In the Business Model and Mapping layer, look for all the logical fact tables that have logical joins to the logical hierarchy table Dim - FSG Hierarchy1. You must 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 tables, 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 logical 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.

8. Using the Oracle BI Administration Tool, in the Presentation layer of the Oracle BI Repository, drag the new hierarchies into the Presentation folder.

   Note: You can rename the hierarchies in the Presentation layer if required.

3.2.2 Configuration Steps for Controlling Your Data Set for Oracle EBS

This section contains additional configuration steps that apply to Oracle BI Applications deployed with Oracle EBS source systems. It contains the following topic:

- Section 3.2.2.1, "How to Configure the Make-Buy Indicator"

3.2.2.1 How to Configure the Make-Buy Indicator

The Make-Buy indicator specifies whether a product material is made in-house or bought from an outside vendor. Out-of-the-box, the make-buy indicator is set to 1 for M (that is, **M**ade in-house) or 2 for B (that is, **B**ought from an outside vendor). If your organization uses these indicator codes, you do not need to change your out-of-the-box ETL mappings.

If your organization uses different indicator codes, you must follow the steps provided in this section to change the indicator mapping in the mapplet 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 Informatica PowerCenter Designer, open the appropriate Oracle EBS adapter (for example, SDE_ORA*Version*_Adaptor), and expand the Mapplets sub-folder.

2. Using the Mapplet Designer, edit the mapplet mplt_SA_ORA_ProductDimension.

3. Double-click the Expression transformation EXP_PRODUCTS to open the Edit Transformations dialog, and display the Ports tab.

4. Use the Expression Editor to edit the expression for the EXT_MAKE_BUY_IND port.

   For example, if you want your indicator code to be 0 for Make and 1 for Buy, change the Formula to:

```
DECODE (INP_PLANNING_MAKE_BUY_CODE,0,'M',1,'B',NULL)
```

5. Validate the mapplet, and save your changes to the repository.

3.3 PeopleSoft-Specific Configuration Steps

This section contains configuration steps that apply to Oracle BI Applications deployed with PeopleSoft source systems.

This section contains the following topics:

- Section 3.3.1, "Configuration Required Before a Full Load for PeopleSoft"
- Section 3.3.2, "Configuration Steps for Controlling Your Data Set for PeopleSoft"

3.3.1 Configuration Required Before a Full Load for PeopleSoft

This section contains configuration steps required before a full data load that apply to Oracle BI Applications deployed with PeopleSoft source systems. It contains the following topics:

- Section 3.3.1.1, "About the GL Account Dimension and Chartfields for PeopleSoft"
- Section 3.3.1.2, "About Mapping PeopleSoft GL Accounts to Group Account Numbers"
- Section 3.3.1.3, "How to Make Corrections to the Group Account Number Configuration"
- Section 3.3.1.4, "About Configuring GL Account Hierarchies"

3.3.1.1 About the GL Account Dimension and Chartfields for PeopleSoft

The GL Account dimension in the Oracle Business Analytics Warehouse is at a granularity of a combination of chartfields. PeopleSoft Financials provides several chartfields for GL accounts, such as account, alternate account, operating unit, department, and so on. The ETL program extracts all possible combinations of these chartfields that you have used and stores each of these chartfields individually in the GL Account dimension. It extracts the combinations of chartfields used from the following PeopleSoft account entry tables:

- PS_VCHR_ACCTG_LINES (Accounts Payable)
- PS_ITEM_DST (Accounts Receivable)
- PS_BI_ACCT_ENTRY (Billings)
- PS_CM_ACCTG_LINE (Costing)
- PS_JRNL_LN (General Ledger)

The GL Account dimension (W_GL_ACCOUNT_D) in the Oracle Business Analytics Warehouse provides a flexible and generic data model to accommodate up to 30 chartfields. These are stored in the generic columns named ACCOUNT_SEG1_CODE, ACCOUNT_SEG2_CODE and so on up to ACCOUNT_SEG30_CODE, henceforth referred to as segments. These columns store the actual chartfield value that is used in your PeopleSoft application.

Mapping PeopleSoft Chartfields

A CSV file has been provided to map the PeopleSoft chartfields to the generic segments. Use this file to specify which PeopleSoft application chartfield should be populated in which segment. The file is called file_glacct_segment_config_psft.csv and is located in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).

The first row in the file is a header row; do not modify this line. The second row in the file is where you specify how to do the mapping. The value for the column ROW_ID is hard coded to '1'; there is no need to change this.

Note that the file contains 30 columns – SEG1, SEG2, up to SEG30. You will have to specify which chartfield needs to be populated in each of these columns by specifying one of the supported values for the chartfields. The following list shows the chartfields currently supported for the PeopleSoft application.

Note: Values are case sensitive. You must specify the values exactly as shown in the following list.

- Account
- Alternate Account
- Operating Unit
- Fund Code
- Department
- Program Code
- Class Field
- Budget Reference
- Product
- Project
- Affiliate
- Fund Affiliate
- Operating Unit Affiliate
- ChartField 1
- ChartField 2
- ChartField 3
- Statistics Code
- PC Business Unit
- Activity ID
- Analysis Type
- Resource Type
- Resource Category
- Resource Sub Category

Note: You only need to include the chartfields in the CSV file that you want to map.

Configuring Projects-Specific Chartfields To use the six chartfields related to Projects (PC Business Unit, Activity ID, Analysis Type, Resource Type, Resource Category, and Resource Sub Category), you must set the DAC parameter $$INCLUDE_PROJ_CHARTFIELDS to 'Y.' You only need to perform this procedure if you want to leverage the six Projects chartfields in your analyses.

To include Projects-specific chartfields in DAC:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab.
3. Set the parameter $$INCLUDE_PROJ_CHARTFIELDS to 'Y'.
4. Run a full ETL.

If you do not use these chartfields, then you can set the parameter $$INCLUDE_PROJ_CHARTFIELDS to a default value 'N' in DAC.

> **Note:** At any point, if you change this parameter value, then a full load ETL must be run.

3.3.1.2 About Mapping PeopleSoft GL Accounts to Group Account Numbers

This section explains how to map General Ledger Accounts to Group Account Numbers, and includes the following topics:

- An overview of setting up General Ledger Accounts - see Section 3.3.1.2.1, "Overview of Mapping GL Accounts to Group Account Numbers."
- A description of how to map General Ledger Accounts to Group Account Numbers - see Section 3.3.1.2.2, "How to Map GL Account Numbers to Group Account Numbers."
- An example that explains how to add Group Account Number metrics to the Oracle BI Repository - see Section 3.3.1.2.3, "Example of Adding Group Account Number Metrics to the Oracle BI Repository."

> **Note:** It is critical that the GL account numbers are mapped to the group account numbers (or domain values) because the metrics in the GL reporting layer use these values. For a list of domain values for GL account numbers, see *Oracle Business Analytics Warehouse Data Model Reference*.

See also Section 3.1.5, "About Mapping GL Accounts to Group Account Numbers for All Source Systems" for additional information.

3.3.1.2.1 Overview of Mapping GL Accounts to Group Account Numbers Group Account Number Configuration is an important step in the configuration of Financial Analytics, as it determines the accuracy of the majority of metrics in the General Ledger and Profitability module. Group Accounts in combination with Financial Statement Codes are also leveraged in the GL reconciliation process, to ensure that subledger data reconciles with GL journal entries. This topic is discussed in more detail later in this section.

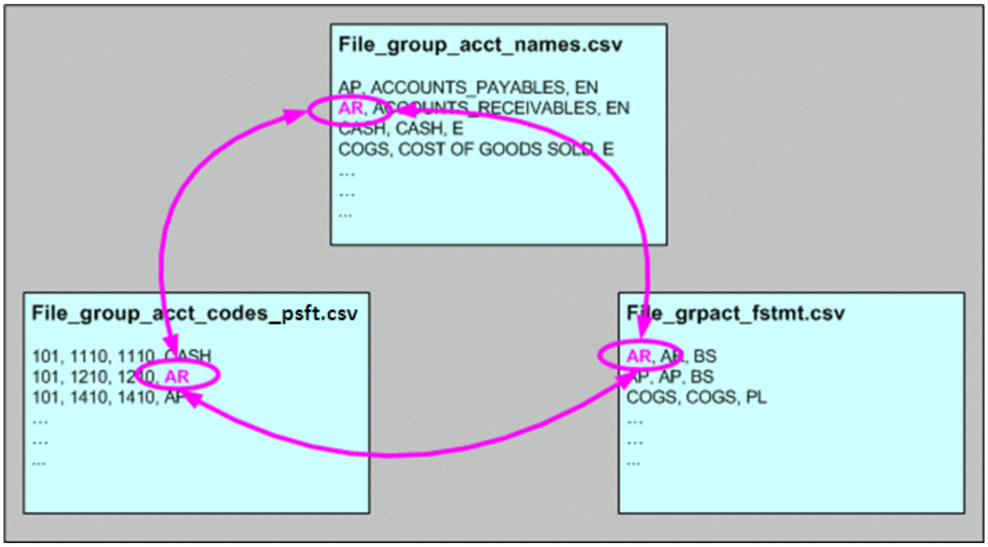
You can categorize your PeopleSoft General Ledger accounts into specific group account numbers. The GROUP_ACCT_NUM field denotes the nature of the General Ledger accounts.

You set up General Ledger accounts using the following configuration files that are located in the $PMServer\SrcFiles directory:

- file_group_acct_names.csv - this file specifies the group account names and their corresponding group account codes.
- file_group_acct_codes_psft.csv - this file maps General Ledger accounts to group account codes.
- file_grpact_fstmt.csv - this file maps Financial Statement Item Codes to group account codes.

Before you load your data, you must ensure that the account values are mapped consistently across these three configuration files. Figure 3–10 shows how the GROUP_ACCOUNT_NUM field in file_group_acct_names.csv must map to the GROUP_ACCT_NUM field in file_group_acct_codes_psft.csv and the GROUP_ACCT_NUM field in file_grpact_fstmt.csv.

Figure 3–10 Configuration Files for Configuring Group Accounts for PeopleSoft



Examples include Cash account, Payroll account, and so on. For a list of the Group Account Number domain values, see *Oracle Business Analytics Warehouse Data Model Reference*. The group account number configuration is used during data extraction as well as front-end reporting. For example, the group account number configuration is used heavily in both Profitability Analysis (Income Statement) and General Ledger analysis. The logic for assigning the accounts is located in the file_group_acct_codes_psft.csv file. This file is located in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).

Table 3–12 Layout of file_group_acct_codes_psft.csv File

BUSINESS_UNIT	FROM_ACCT	TO_ACCT	GROUP_ACCT_NUM
AUS01	101010	101099	AP
AUS01	131010	131939	AR
AUS01	152121	152401	COGS

Table 3–12 (Cont.) Layout of file_group_acct_codes_psft.csv File

BUSINESS_UNIT	FROM_ACCT	TO_ACCT	GROUP_ACCT_NUM
AUS01	171101	173001	OTHER
AUS01	240100	240120	REVENUE
AUS01	251100	251120	TAX[1]

[1] Oracle's PeopleSoft adapters for Financial Analytics do not support the Tax base fact (W_TAX_XACT_F).

In Table 3–12, in the first row, all accounts within the account number range from 101010 to 101099 containing a Business Unit equal to AUS01 are assigned to AP. Each row maps all accounts within the specified account number range and with the given Business Unit. If you need to assign a new group of account numbers, you can then assign GL accounts to the new group of account numbers in the file_group_acct_codes_psft.csv file.

The file_grpact_fstmt.csv file specifies the relationship between a group account number and a Financial Statement Item code. Table 3–13 shows the Financial Statement Item codes to which Group Account Numbers must map, and their associated base fact tables.

Table 3–13 Financial Statement Item Codes and Associated Base Fact Tables

Financial Statement Item Codes	Base Fact Tables
AP	AP base fact (W_AP_XACT_F)
AR	AR base fact (W_AR_XACT_F)
COGS	Cost of Goods Sold base fact (W_GL_COGS_F)
REVENUE	Revenue base fact (W_GL_REVN_F)
TAX	Tax base fact (W_TAX_XACT_F)[1]
OTHERS	GL Journal base fact (W_GL_OTHER_F)

[1] Oracle's PeopleSoft adapters for Financial Analytics do not support the Tax base fact (W_TAX_XACT_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. This association is important to perform GL reconciliation and ensure the subledger data reconciles with GL journal entries. It is possible that after an invoice has been transferred to GL, a GL user might adjust that invoice in GL. In this scenario, it is important to ensure that the adjustment amount is reflected in the subledger base fact as well as balance tables. To determine such subledger transactions in GL, the reconciliation process uses Financial Statement Item codes.

Financial Statement Item codes are internal codes used by the ETL process to process the GL journal records during the GL reconciliation process against the subledgers. When the ETL 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 ETL 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 the

ETL program will try to go against the Revenue base fact table (W_GL_REVN_F), and try to locate the corresponding matching Revenue accounting entry.

3.3.1.2.2 How to Map GL Account Numbers to Group Account Numbers This section explains how to map General Ledger Account Numbers to Group Account Numbers.

> **Note:** If you add new Group Account Numbers to the file_group_acct_codes_psft.csv file, you must also add metrics to the Oracle BI repository. See Section 3.3.1.2.3, "Example of Adding Group Account Number Metrics to the Oracle BI Repository" for more information.

To map PeopleSoft GL account numbers to group account numbers:

1. Open the file_group_acct_codes_psft.csv file with a text editor in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).
2. For each GL account number that you want to map, create a new row in the file containing the following fields:

Field Name	Description
BUSINESS_UNIT	The ID of the BUSINESS UNIT.
FROM ACCT	The lower limit of the natural account range. This is based on the natural account segment of your GL accounts.
TO ACCT	The higher limit of the natural account range. This is based on the natural account segment of your GL accounts.
GROUP_ACCT_NUM	This field denotes the group account number of the General Ledger account, as specified in the file_group_acct_names.csv file. For example, 'AP' for Accounts Payables, 'CASH' for cash account, 'GEN PAYROLL' for payroll account, and so on.

For example:

```
AUS01, 1110, 1110, CASH
AUS01, 1210, 1210, AR
AUS01, 1220, 1220, AR
```

> **Note:** You can optionally remove the unused rows in the CSV file.

3. Ensure that the values that you specify in the file_group_acct_codes_psft.csv file are consistent with the values that are specified in the file_group_acct_names.csv file and the file_grpact_fstmt.csv file.

   See Figure 3–10, "Configuration Files for Configuring Group Accounts for PeopleSoft" for an example.

4. Save and close the CSV file.

3.3.1.2.3 Example of Adding Group Account Number Metrics to the Oracle BI Repository If you add new Group Account Numbers to the file_group_acct_codes_psft.csv file (as described in Section 3.3.1.2.2, "How to Map GL Account Numbers to Group Account

Numbers"), you must also add metrics to the Oracle BI repository to expose the new Group Account Numbers. This example illustrates how you can add Group Account Number metrics using the Oracle BI Administration Tool.

This example assumes that you have a new Group Account Number called 'Payroll,' and you want to add a new metric to the Presentation layer called 'Payroll Expense.'

To add a new metric in the logical table Fact - Fins - GL Other Posted Transaction:

1. Using the Administration Tool, open OracleBIAnalyticsApps.rpd.

   The OracleBIAnalyticsApps.rpd file is located at:

   ```
   ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_
   obisn\repository
   ```

2. In the Business Model and Mapping layer:

   a. Create a logical column named 'Payroll Expense' in the logical table 'Fact - Fins - GL Other Posted Transaction.'

      For example, right-click the Core\Fact - Fins - GL Other Posted Transaction object and choose **New Object**, then **Logical Column**, to display the Logical Column dialog. Specify Payroll Expense in the **Name** field.

   b. Display the Aggregation tab, and then choose 'Sum' in the **Default aggregation rule** drop-down list.

   c. Click **OK** to save the details and close the dialog.

   d. Expand the Core\Fact - Fins - GL Other Posted Transaction\Sources folder and double click the Fact_W_GL_OTHER_GRPACCT_FSCLYR_A source to display the Logical Table Source dialog.

   e. Display the Column Mapping tab.

   f. Select **Show unmapped columns**.

   g. Locate the Payroll Expense expression, and click the Expression Builder button to open Expression Builder.

   h. Use Expression Builder to specify the following SQL statement:

      ```
      CASE WHEN "Oracle Data Warehouse"."Catalog"."dbo".
      "Dim_W_GL_GROUP_ACCOUNT_D"."GROUP_ACCOUNT_NUM" = 'PAYROLL'
      THEN "Oracle Data Warehouse"."Catalog"."dbo".
      "Fact_Agg_W_GL_OTHER_GRPACCT_FSCLYR_A"."OTHER_GLOBAL1_AMT" ELSE NULL END
      ```

      The case condition refers to the new Group Account Number 'Payroll' and uses it as a filter.

   i. Repeat steps d to h for each Logical Table Source. Modify the expression in step h appropriately for each LTS by using the appropriate fact table that corresponds to the Logical Table Source.

      Steps d to h must be repeated for each Logical Table Source because in this example, there are multiple Logical Table Sources for fact table and aggregation tables in this logical table. Modify the expression in step h appropriately for each Logical Table Source by using the appropriate fact table to which it corresponds.

3. Save the details.

4. To expose the new repository objects in end users' dashboards and reports, drag the new objects from the Business Model and Mapping layer to an appropriate folder in the Presentation layer.

To add a new metric in the logical table Fact - Fins - GL Balance:

1. Using the Administration Tool, open OracleBIAnalyticsApps.rpd.

   The OracleBIAnalyticsApps.rpd file is located at:

   ```
   ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_
   obisn\repository
   ```

2. In the Business Model and Mapping layer:

   a. Create a logical column named 'Payroll Expense' in logical table 'Fact - Fins - GL Balance.'

      For example, right-click the Core\Fact - Fins - GL Balance\ object and choose **New Object**, then **Logical Column**, to display the Logical Column dialog. Specify Payroll Expense in the **Name** field.

   b. In the Column Source tab, select **Derived from existing columns using an expression**.

   c. Click the Expression Builder button to display Expression Builder.

   d. Use Express Builder to specify the following SQL statement:

      ```
      FILTER("Core"."Fact - Fins - GL Balance"."Activity Amount"
      USING "Core"."Dim - GL Account"."Group Account Number" = 'Payroll')
      ```

      The case condition refers to the new Group Account Number 'Payroll.'

3. Save the details.

4. To expose the new repository objects in end users' dashboards and reports, drag the new objects from the Business Model and Mapping layer to an appropriate folder in the Presentation layer.

3.3.1.3 How to Make Corrections to the Group Account Number Configuration

Note: Refer to Section 3.3.1.2, "About Mapping PeopleSoft GL Accounts to Group Account Numbers" for general concepts about group account number and Financial Statement Item code.

When a user maps a GL natural account to an incorrect group account number, incorrect accounting entries might be inserted into the fact table. For example, the natural account 1210 is mistakenly classified under 'AR' group account number when it should be classified under 'AP' group account number. When this happens, the ETL program will charge all the GL journal lines to account 1210 and try to reconcile these GL journal lines against subledger accounting records in the AR fact table (W_AR_XACT_F). Since these GL journal lines did not come from AR, the ETL program will not be able to find the corresponding subledger accounting records for these GL journal lines. In this case, the ETL program will insert 'Manual' records into the AR fact table because it thinks that these GL journal lines are 'Manual' journal entries created directly in the GL system charging to the AR accounts. This entire process is called the GL reconciliation process.

To revert these 'Manual' entries in the AR fact table, use the 'Group Account Number Clean Up' program provided in Oracle BI 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, the ETL program will try to look for the corresponding subledger accounting records in the AP fact (W_AP_XACT_F), provided that you have reassigned the natural account 1210 to the 'AP' group account number in the file_group_acct_codes_psft.csv file.

> **Note:** The Financials – Group Account Number Clean Up subject area should be executed only when you need to correct group account number assignments. This subject area should not be included in the standard Financials execution plan. You must run it separately.

To do group account correction:

1. Correct the mapping of GL natural account to the group account in the input CSV file called file_group_acct_codes_psft.csv.

   For example, before correction, a CSV file has the following values:

   BUSINESS_UNIT = AUS01

   FROM ACCT = 1110

   TO ACCT = 1110

   GROUP_ACCT_NUM = CASH

   After correction, if the account '1210' originally belonged to the 'AP' group account number and after correcting the GL natural account to the group account, the CSV file would have the following values:

   BUSINESS_UNIT = AUS01

   FROM ACCT = 1210

   TO ACCT = 1210

   GROUP_ACCT_NUM = AR

2. In DAC, do the following:

   a. Go to the Design view, and select the appropriate custom container from the drop-down list.

   b. Display the Subject Areas tab.

   c. Query for the 'Financials – General Ledger' Subject Area.

   d. Display the Configuration Tags subtab, and verify and which of the following Configuration Tags is marked as 'Inactive':

      - Financials – Calculate GL Balance

      - Oracle – Extract GL Balance

      By default, 'Financials – Calculate GL Balance' should be marked as inactive.

   e. Query for the 'Financials – Group Account Number Clean Up' Subject Area and do the following:

      - If in the preceding step, the configuration tag 'Financials – Calculate GL Balance' is marked as 'Inactive', then 'Financials – Calculate GL Balance Reverse' should be marked as 'Inactive' as well.

      - If in the preceding step, 'Oracle – Extract GL Balance' is marked as 'Inactive', then 'Financials – Calculate GL Balance Reverse' should be marked as active (that is, the **Inactive** check box should be cleared).

3. If you need to make any change in the preceding step, then you must display the Subject Areas tab and reassemble the Subject Area named 'Financials – Group Account Number Clean Up'.
4. Rebuild and execute the appropriate Execution Plan that you are using to load the Subject Area named 'Financials – Group Account Number Clean Up'.

3.3.1.4 About Configuring GL Account Hierarchies

Oracle Business Intelligence Applications supports hierarchy for all these segments. If you have created a 'tree' in PeopleSoft for any of these chartfields, you can extract these trees into Oracle Business Analytics Warehouse's hierarchy to analyze your facts at any levels of the hierarchy. See also Section 3.3.2.2, "About Configuring Department Trees" for information about extracting trees, Section 16.4, "How to Set Up Product Category Hierarchies" for information about including the trees as hierarchies in the RPD file.

GL Balance Aggregate The Oracle Business Analytics Warehouse data model has a fact table (W_GL_BALANCE_F) that stores GL Balance for all your GL accounts. To compress this fact table for better performance, it also provides an aggregate table (W_GL_BALANCE_A) built on top of this fact table that stores GL Balances for up to six segments of your choice. You can configure the number of segments you want in the aggregate table and which segments you want. This configuration is done at the third line of the file_glacct_segment_config_psft.csv file. Specify a value of 'Y' under the segment column that you want to be included in the aggregate table.

Note: You can have up to six 'Y's in the file. You do not have to use all six. For example, if you want only three segments in your aggregate, you only need to specify three 'Y's.

Example CSV File Configuration As an example for the configuration of the file_glacct_segment_config_psft.csv file, consider the following scenario:

Your system uses four chartfields: Account, Alt Account, Operating Unit, and Dept. Assume that you are interested in analyzing your data by only three of the four chartfields: Account, Operating Unit, Department, and that you typically view your GL Balances at the combination of Account and Department. Only occasionally do you want to view your GL balances as the combination of all three chartfields. In this scenario, the CSV file would look similar to the values shown in Table 3–14.

Table 3–14 Example CSV Chartfield Mapping Values

ROW_ID	SEG1	SEG2	SEG3
1	Account	Operating Unit	Department
AGGREGATION	Y		Y

With this configuration, W_GL_ACCOUNT_D would store the 'Account' chartfield value in SEGMENT1 column, 'Operating Unit' chartfield value in SEGMENT2 column, and so forth. The GL Balance aggregate table, W_GL_BALANCE_A would store GL Balances for each unique combination of 'Account' chartfield and 'Department' chartfield.

3.3.2 Configuration Steps for Controlling Your Data Set for PeopleSoft

This section contains additional configuration steps that apply to PeopleSoft. It contains the following topics:

- Section 3.3.2.1, "About Employee Dimension Extraction"
- Section 3.3.2.2, "About Configuring Department Trees"
- Section 3.3.2.3, "How to Configure Department Trees and Business Unit Trees"
- Section 3.3.2.4, "How to Configure Country Codes for PSFT"

3.3.2.1 About Employee Dimension Extraction

If you only manage your resources in HRMS or create only non-employees in FSCM and have HRMS integration for employees, you will need to run the integration broker to synchronize HRMS tables with FSCM tables.

For more information about implementing PeopleSoft Resource Management for employee and non-employee source data, see *PeopleSoft Resource Management 9.0 PeopleBook*.

3.3.2.2 About Configuring Department Trees

Oracle Business Intelligence Applications supports PeopleSoft department-based organization hierarchy. ETL mappings extract and flatten PeopleSoft department trees into a flattened organization hierarchy. ETL parameters also enable you to flatten department trees by SetID and tree name.

Supported Tree Structures

Oracle HR Analytics supports winter tree and summer tree structure types. Winter trees have nodes but do not have detail values. Summer trees have both nodes and detail values. Nodes are grouped into levels, and Oracle HR Analytics supports only strictly enforced tree levels in which all nodes on the same level represent the same type of entity. For detailed information about tree structures, see the PeopleSoft documentation.

How Oracle HR Analytics Handles Department Trees

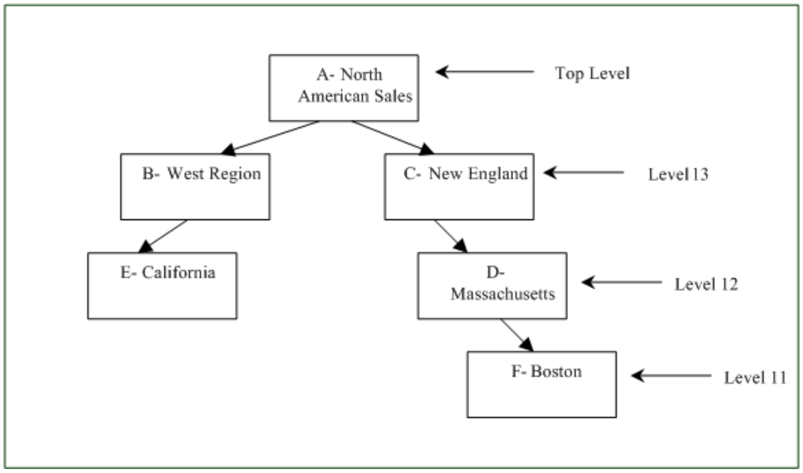
PeopleSoft departments and the associated department tress are supported in Oracle HR Analytics as the organization dimension (W_INT_ORG_D) and the flattened organizational hierarchy structure (W_INT_ORG_DH).

Oracle HR Analytics flattens trees up to 15 levels deep, level 0 to 14, with level 0 as the bottom node. During the tree flattening ETL process, every tree node is inserted into W_INT_ORG_DH along with the path from the top-most node on the tree. If a node is less than 15 levels deep, the node value will be repeated in all levels below the node level.

Example of How a Department Tree Is Populated

The following diagram and tables give an example of how the department tree is populated into W_INT_ORG_D and W_INT_ORG_DH. This example uses the tree name "NA Sales" and the setID "Share."

Figure 3–11 How the Department Tree Populates Data Warehouse Tables



The department table (PS_DEPT_TBL) populates the Internal Organization dimension table (W_INT_ORG_D) as follows:

Table 3–15 How PS_DEPT_TBL Populates W_INT_ORG_D

ROW_ID	ORG_NUM	ORG_NAME	HR_ORG_FLAG
1	A	American Sales	Y
2	B	West Region	Y
3	C	New England	Y
4	D	Massachusetts	Y
5	E	California	Y
6	F	Boston	Y

The department tree populates the Internal Organization hierarchy table W_INT_ORG_DH as follows:

Table 3–16 How PS_DEPT_TBL Populates W_INT_ORG_DH

ORG_WID	ORG_HIER(1-11)_NUM	ORG_HIER(1-11)_NAME	ORG_HIER12_NUM	ORG_HIER12_NAME	ORG_HIER13_NUM	ORG_HIER13_NAME	ORG_TOP_NUM	ORG_TOP_NAME	HIERARCHY_NAME	W_HIERARCHY_CLASS	FIXED_HIER_LEVEL	HR_ORG_FLG
1	A	North American Sales	A	North American Sales	A	North American Sales	A	North American Sales	Share~NA Sales	HR Org	14	Y
2	B	West Region	B	West Region	B	West Region	A	North American Sales	Share~NA Sales	HR Org	13	Y
3	C	New England	C	New England	C	New England	A	North American Sales	Share~NA Sales	HR Org	13	Y

Table 3–16 (Cont.) How PS_DEPT_TBL Populates W_INT_ORG_DH

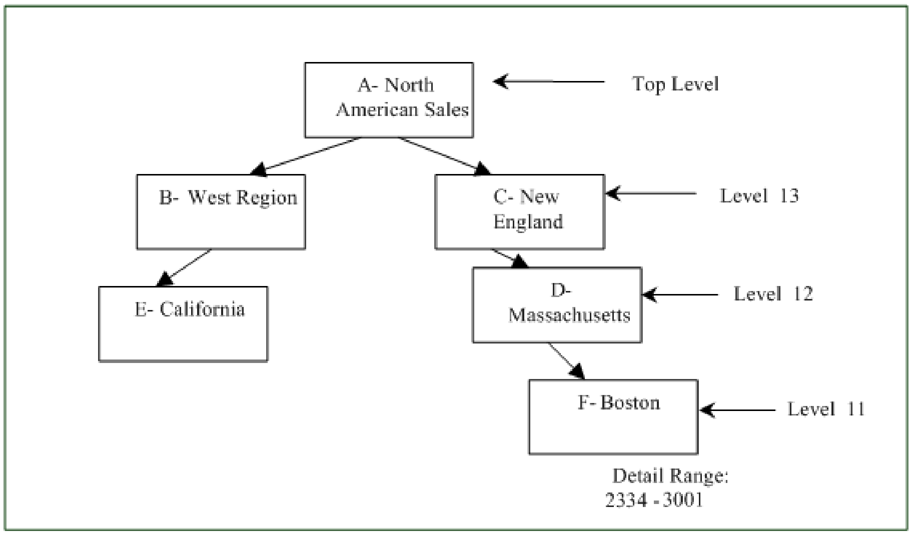
ORG_WID	ORG_HIER(1-11)_NUM	ORG_HIER(1-11)_NAME	ORG_HIER12_NUM	ORG_HIER12_NAME	ORG_HIER13_NUM	ORG_HIER13_NAME	ORG_TOP_NUM	ORG_TOP_NAME	HIERARCHY_NAME	W_HIERARCHY_CLASS	FIXED_HIER_LEVEL	HR_ORG_FLG
4	D	Massachusetts	D	Massachusetts	C	New England	A	North American Sales	Share~NA Sales	HR Org	12	Y
5	E	California	E	California	B	West Region	A	North American Sales	Share~NA Sales	HR Org	11	Y
6	F	Boston	D	Boston	C	New England	A	North American Sales	Share~NA Sales	HR Org	12	Y

How a Summer Tree Is Flattened

The tree flattening process also supports summer trees. A summer tree is a tree with detail ranges. If a tree has detail ranges specified for the bottom nodes, the extraction process creates a number of nodes in W_INT_ORG_HIER corresponding to departments in the specified node range.

If a tree is a summer tree, the granularity of the data returned from the ETL source Qualifier is one row per specified range. The lowest parent nodes of the tree can be repeated multiple times because multiple ranges can be created. The following diagram shows how a summer tree is flattened.

Figure 3–12 How a Summer Tree Is Flattened



The detail ranges populate the Internal Organization dimension table W_INT_ORG_D as follows:

Table 3–17 How Detail Ranges Populate W_INT_ORG_D

ROW_WID	ORG_NUM	ORG_NAME	HR_ORG_FLG
7	2334	Appliances	Y
8	2340	Home Theater	Y
9	3001	MP3 Players	Y

The summer tree detail range is populated in W_INT_ORG_DH as follows:

Table 3–18 How Detail Ranges Populate W_INT_ORG_DH

ORG_WID	ORG_HIER (1-10)	ORG_HIER11	ORG_HIER12	ORG_HIER13	ORG_TOP	FIXED_HIER_LVL	HR_ORG_FLG
7	Boston	Boston	Massachusetts	New England	North American Sales	11	Y
8	Boston	Boston	Massachusetts	New England	North American Sales	10	Y
9	Boston	Boston	Massachusetts	New England	North American Sales	10	Y

How the Flattened Internal Organization Hierarchy Is Presented in Oracle BI Enterprise Edition

The Oracle HR Analytics Presentation Catalog delivers a 15-level employee organization. Employee Organization hierarchy levels are mapped to Internal Organization dimension and hierarchy tables as follows:

RPD Presentation Layer	Physical Table Mapping
Employee Organization Number	W_INT_ORG_D.ORG_NUM
Employee Organization Name	W_INT_ORG_D.ORG_NAME
Employee Organization Hierarchy Name	W_INT_ORG_DH.HIERARCHY_NAME
Hierarchy Version	W_INT_ORG_DH.HIERARCHY_VERSION
Employee Organization Hierarchy 1	W_INT_ORG_DH.HIER1_NUM
Employee Organization Hierarchy 2	W_INT_ORG_DH.HIER2_NUM
Employee Organization Hierarchy 3	W_INT_ORG_DH.HIER3_NUM
Employee Organization Hierarchy 4	W_INT_ORG_DH.HIER4_NUM
Employee Organization Hierarchy 5	W_INT_ORG_DH.HIER5_NUM
Employee Organization Hierarchy 6	W_INT_ORG_DH.HIER6_NUM
Employee Organization Hierarchy 7	W_INT_ORG_DH.HIER7_NUM
Employee Organization Hierarchy 8	W_INT_ORG_DH.HIER8_NUM
Employee Organization Hierarchy 9	W_INT_ORG_DH.HIER9_NUM
Employee Organization Hierarchy 10	W_INT_ORG_DH.HIER10_NUM
Employee Organization Hierarchy 11	W_INT_ORG_DH.HIER11_NUM
Employee Organization Hierarchy 12	W_INT_ORG_DH.HIER12_NUM

RPD Presentation Layer	Physical Table Mapping
Employee Organization Hierarchy 13	W_INT_ORG_DH.HIER13_NUM
Employee Organization Hierarchy 14	W_INT_ORG_DH.HIER14_NUM

The following table describes HR Organization dimension and dimension hierarchy tables:

Table Name	Description	Source Tables
W_INT_ORG_DS	HR Organization dimension staging table	PS_DEPT_TBL
W_INT_ORG_D	HR Organization dimension table	W_INT_ORG_DS
W_INT_ORG_DHS	HR Organization dimension hierarchy staging table	PSTREESTRCT PSTREENODE PSTREELEVEL
W_INT_ORG_DH	HR Organization dimension hierarchy table	W_INT_ORG_DHS

Table 3–19 shows the temporary tables and their corresponding sessions that are used to handle the extraction and load of the trees.

Table 3–19 Temporary Tables and Corresponding Sessions for Extracting and Loading Trees

Sequence	Temporary Table Name	Session
Sequence 1	W_PSFT_INT_ORG_DEPT_DH_TMP	SDE_PSFT_Stage_InternalOrganizationDimension_DepartmentHierarchy_GetDept
Sequence 2	W_PSFT_INT_ORG_TREE_TMP	1. SDE_PSFT_Stage_InternalOrganizationDimension_Hierarchy_Extract 2. SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_Extract
Sequence 3	W_PSFT_INT_ORG_VERT_DH_TMP	SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_GetHierarchyLevel
Sequence 4	W_PSFT_INT_ORG_DTLRGE_DH_TMP	SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_DeriveRange
Sequence 5	W_PSFT_INT_ORG_FLAT_DH_TMP	SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_Flatten

3.3.2.3 How to Configure Department Trees and Business Unit Trees

Oracle Business Intelligence Applications supports both PeopleSoft Department organization hierarchies, and Business Unit-based organization hierarchies. ETL mappings extract and flatten PeopleSoft Department or Business Unit trees into a flattened organization hierarchy. ETL parameters also enable you to flatten Department or Business Unit trees by SetID and tree name.

DAC provides the parameter $$TREE_SETID_NAME_LIST to configure the tree flattening ETL process. $$TREE_SETID_NAME_LIST supports two PeopleSoft tree parameters, SETID and TREE_NAME.

No DAC parameter is provided for the PeopleSoft tree parameter EFFDT. The tree extract mapping has built-in logic to extract the current effectively-dated tree (excluding future date) for the specified tree name. There is no increment extraction for PeopleSoft trees, and a full extraction is always performed during each ETL process.

To configure Department and Business Unit Trees:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab.
3. Select the task SDE_PSFT_Stage_InternalOrganizationDimension_Hierarchy_Extract, and display the Parameters subtab.
4. Use the $$TREE_SETID_NAME_LIST parameter to specify the tree set ID and tree name in the format of '*setid~tree_name*'.

   Wrap the *setid~tree_name* value in single quotation marks. If you have multiple trees, separate them using a comma. For example:

   ```
   'SHARE~DEPT1', 'US~DEPT1', 'EU~DEPT1'
   ```

5. Use the $$TREE_STRUCT_ID_LIST parameter to specify the struct IDs for which data needs to be extracted, separating each ID with a comma.

   The default value of this parameter is 'BUSINESS_UNIT','DEPARTMENT'.

6. Repeat steps 3 - 5 for the SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_Extract task.

3.3.2.4 How to Configure Country Codes for PSFT

The Geography Country dimension is used as a roll-up dimension for the Business Location and Geography dimensions.

Oracle Business Intelligence Applications is installed out-of-the-box with a standard set of countries (in the file domainValues_GeoCountry_ISO_Country_Codes_*Type*.csv). If you add a country in your source system that is not in the file domainValues_GeoCountry_ISO_Country_Codes_*Type*.csv, you must modify the file to accommodate this data.

The file domainValues_GeoCountry_ISO_Country_Codes_*Type*.csv is a domain value lookup for the W_GEO_COUNTRY_D table for ISO Country Codes. It contains the following columns: COUNTRY_CODE, COUNTRY_NAME, ISO_COUNTRY_CODE, ISO_COUNTRY_NAME, ISO_NUMERICAL_CODE, ISO_ALPHA3_CODE, NLS_TERRITORY.

Note: The COUNTRY_CODE in domainValues_GeoCountry_ISO_Country_Codes_psft.csv should match the values from the PS_COUNTRY_TBL in the PeopleSoft application database.

To configure Country Codes for PeopleSoft:

1. Using a text editor, edit the domainValues_GeoCountry_ISO_Country_Codes_psft.csv located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
2. Add a new row to the CSV file with appropriate values.
3. Save and close the file.

3.4 Oracle Siebel-Specific Configuration Steps

There are no mandatory or additional configuration steps that apply to Oracle BI Applications deployed with Siebel source systems.

3.5 Oracle's JD Edwards-Specific Configuration Steps

This section contains configuration steps that apply to Oracle BI Applications deployed with Oracle's JD Edwards EnterpriseOne or JD Edwards World source systems.

This section contains the following topic:

- Section 3.5.1, "Configuration Required Before a Full Load for Oracle's JD Edwards EnterpriseOne or JD Edwards World"

3.5.1 Configuration Required Before a Full Load for Oracle's JD Edwards EnterpriseOne or JD Edwards World

This section contains configuration steps required before a full data load that apply to Oracle BI Applications deployed with Oracle's JD Edwards EnterpriseOne source systems. It contains the following topics:

- Section 3.5.1.1, "How to Configure Category Codes"
- Section 3.5.1.2, "About Configuring the Code Dimension for Oracle's JD Edwards EnterpriseOne or JD Edwards World UDCs"
- Section 3.5.1.3, "About Configuring Quarters for a Fiscal Pattern and Fiscal Year of Time Dimensions"
- Section 3.5.1.4, "About Mapping JD Edwards EnterpriseOne or JD Edwards World GL Accounts to Group Account Numbers"
- Section 3.5.1.5, "About Configuring GL Account Segment Hierarchies"
- Section 3.5.1.6, "Setting Up $$JDE_RATE_TYPE"

3.5.1.1 How to Configure Category Codes

You use DAC parameters to load category codes from Oracle's JD Edwards EnterpriseOne and JD Edwards World into dimension tables. When you configure the DAC parameters, you choose which category codes to map to the 20 new columns in the dimensions shown in the following list. In addition, you can also choose category codes to map to the existing columns shown in Table 3–20. In DAC, by default NULL is assigned to all DAC parameters in the name of new attributes and configurable columns in the dimension table.

The following dimensions support category codes:

- W_INT_ORG_D
- W_PRODUCT_D
- W_CUSTOMER_ACCOUNT_D
- W_PARTY_ORG_D
- W_GL_ACCOUNT_D

Note: Category codes pertaining to Oracle's JD Edwards EnterpriseOne and JD Edwards World GL Accounts are mapped into the existing Account Segment columns in the W_GL_ACCOUNT_D table. Configuration files (not DAC parameters) are used to configure these category codes (for more information, see Section 5.2.4.8, "How to Configure the file_glacct_segment_config_jde.csv").

Table 3–20 lists the additional fields in the dimension tables to which you can also map category codes.

Table 3–20 Additional Dimension Table Fields to Which You Can Map Category Codes

Table	Column
W_CUSTOMER_ACCOUNT_DS	ACCOUNT_TYPE_CODE
W_CUSTOMER_ACCOUNT_DS	ACCOUNT_CLASS_CODE
W_INT_ORG_DS	STATE_REGION
W_INT_ORG_DS	COUNTRY_REGION
W_PARTY_ORG_DS	LINE_OF_BUSINESS
W_PARTY_ORG_DS	REGION
W_PARTY_ORG_DS	ACCNT_AHA_NUM
W_PARTY_ORG_DS	ACCNT_CLASS
W_PARTY_ORG_DS	ACCNT_HIN_NUM
W_PARTY_ORG_DS	ACCNT_REGION
W_PARTY_ORG_DS	ACCNT_VALUE
W_PARTY_ORG_DS	CUST_CAT_CODE
W_PRODUCT_DS	CONFIG_CAT_CODE
W_PRODUCT_DS	INDUSTRY_CODE
W_PRODUCT_DS	BRAND
W_PRODUCT_DS	COLOR
W_PRODUCT_DS	UNIV_PROD_CODE

To configure category codes in DAC:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab, and perform a query for the task.

   Table 3–21 shows Oracle JD Edwards EnterpriseOne tables that contain the category codes that you can map from and the corresponding DAC tasks that you need to edit.

Table 3–21 Oracle's JD Edwards EnterpriseOne/JD Edwards World Table Names and Their Corresponding DAC Task Names

Oracle's JD Edwards EnterpriseOne/JD Edwards World Table	DAC Task
F0006 Business Unit Master	SDE_JDE_InternalOrganization_BusinessUnits
F0101 Address Book Master, F03012 Customer Master by Line of Business	SDE_JDE_PartyOrganisationDimension
F03012 Customer Master by Line of Business (JD Edwards EnterpriseOne only)	SDE_JDE_Customer_Account_Dimension

Table 3–21 (Cont.) Oracle's JD Edwards EnterpriseOne/JD Edwards World Table Names and Their Corresponding DAC Task Names

Oracle's JD Edwards EnterpriseOne/JD Edwards World Table	DAC Task
F0301 Customer Master (JD Edwards World only)	SDE_JDE_Customer_Account_Dimension
F4101 Item Master	SDE_JDE_ProductDimensions

3. In the Detail area of the selected task, display the Parameters tab. Modify the value in the DAC parameters by overriding the NULL value with the JDE category codes column name and table name. For example:

   ```
   $$FLEX_ATTRIB_2_CHAR = F0006.MCRP24
   ```

3.5.1.2 About Configuring the Code Dimension for Oracle's JD Edwards EnterpriseOne or JD Edwards World UDCs

The file_udc_category_mapping_jde.csv file loads Oracle's JD Edwards EnterpriseOne or JD Edwards World user defined codes (UDCs) into the Code (W_CODE_D) dimension. To avoid loading all UDCs into the Code dimension, use this flat file to specify a particular set of UDCs that you want to load.

Before you configure the CSV file, identify the UDCs that you need depending on the functional module that you are loading. Choose only the UDCs in transactional tables that are involved in the ETL process. Oracle recommends preparing a list of UDCs that you can refer to when configuring the CSV file.

There are three columns in the CSV file. The first two columns are used to identify the system codes and user defined codes. Together, these columns are used to identify the UDCs that will be loaded into W_CODE_D. The third column is the category into which you want to load the codes in W_CODE_D.

Categories in W_CODE_D are used to group together codes intended for a similar purpose. For example, UDC 00||CN stores the country code and description. To store this under the COUNTRY category in W_CODE_D, enter the following row in the CSV file:

```
00 CN COUNTRY
```

In the CSV file, you specify the system code and user defined code and associate it with the category to which you want the UDCs loaded. This data is loaded into UDC_CATEGORY_MAP_TMP table, which leverages the data and loads the relevant codes into the Code dimension.

Table 3–22 contains the UDCs that you can map using the file_udc_category_mapping_jde.csv flat file (in the order listed in the file).

Table 3–22 List of UDCs for the file_udc_category_mapping_jde.csv Flat File

System Code	User Defined Code	Category
00	PY	SUPPLIER_PAYMENT_METHOD
00	CN	COUNTRY
01	GD	GENDER
01	LP	LANGUAGE
00	S	STATE

Table 3–22 (Cont.) List of UDCs for the file_udc_category_mapping_jde.csv Flat File

System Code	User Defined Code	Category
01	PH	FIN_PHONE_USAGE
00	MC	DIVISION_TYPE
00	TS	FIN_ORG_STRUCTURE
01	SC	PROFITCNTR_GRP
H00	TA	TAX_CODE
00	PY	PAYMENT_METHOD
98	IC	ACCT_DOC~STATUS
00	UM	UOM
41	I	STORAGE_TYPE
49	BX	HAZARD_MTL
41B	PG	PROD_GRP
46	EQ	CONTAINER
41	I	PRODUCT_TYPE
42	FR	FRGHT_TERMS
06	G	JOB
07	MS	MARITAL_STATUS
05	HQ	DISABILITY
00	CN	NATIONAL_ID_TYPE

The temporary table in the data warehouse stores the UDC to Category mappings. When the ETL for Code dimension is started, all UDCs that are found in this table, along with their mappings to respective categories, are extracted from the source and loaded into the W_CODE_D table.

Determining how to associate UDCs with categories requires looking at the SILO mappings, where the descriptions are resolved through lookups in W_CODE_D. The category is hard coded in the lookups. To resolve descriptions from W_CODE_D, you must ensure that the UDCs are loaded into the proper categories.

For information on creating categories and configuring the Code dimension for Oracle's JD Edwards EnterpriseOne or JD Edwards World category codes, see the Knowledge Article on My Oracle Support.

3.5.1.3 About Configuring Quarters for a Fiscal Pattern and Fiscal Year of Time Dimensions

Oracle's JD Edwards EnterpriseOne and JD Edwards World do not have a concept of defining the quarters for a fiscal pattern or a fiscal year. Therefore, a configurable flat file is provided to populate quarter information. This configuration file enables you to feed quarter information such as Quarter Number for each period, Quarter Start Date, and Quarter End Date.

For information about how to configure this flat file, see Section 5.2.4.9, "How to Configure the file_lkp_fiscal_period_Qtr_Config_jde.csv." Each fiscal pattern can have a varying number of periods as supported by the OLTP. Therefore, the quarter configuration is required for each fiscal year and for each fiscal pattern. Table 3–23 shows an example of the file_lkp_fiscal_period_Qtr_Config_jde.csv opened in a text editor.

Table 3–23 Example of file_lkp_fiscal_period_Qtr_Config_jde.csv

Fiscal Pattern	Year	Period	QuarterNo	QuarterStart	QuarterEnd
F	4	1	1	6/1/2004	8/30/2004
F	4	2	1	6/1/2004	8/30/2004
F	4	3	1	6/1/2004	8/30/2004
F	4	4	2	9/1/2004	11/30/2004
F	4	5	2	9/1/2004	11/30/2004
F	4	6	2	9/1/2004	11/30/2004
F	4	7	3	12/1/2004	2/28/2005
F	4	8	3	12/1/2004	2/28/2005
F	4	9	3	12/1/2004	2/28/2005
F	4	10	4	3/1/2005	3/31/2005
F	4	11	4	3/1/2005	3/31/2005
F	4	12	4	3/1/2005	3/31/2005
F	4	13	4	3/1/2005	3/31/2005
F	4	14	4	3/1/2005	3/31/2005

For each fiscal year in the F0008 table, you must define the quarters for each fiscal period. The quarter information is used in the calculation of aggregates by quarter.

The W_MCAL_CONTEXT_G table in the Oracle Business Analytics Warehouse stores calendars associated with the ORG ID, Ledger ID, and Operating Unit columns. In Oracle's JD Edwards EnterpriseOne and JD Edwards World, the fiscal date patterns are associated with the company which forms the ORG_ID and LEDGER_ID.

The W_MCAL_CAL_D table stores the calendar information. Every distinct Fiscal Date Pattern stored in the Fiscal Date Pattern table (F0008) has an entry in this table. The grain of this dimension is the Date Pattern Type, which identifies the Calendar in the Oracle Business Analytics Warehouse. This dimension does not have an association with the Fiscal year for that pattern. The MCAL_CAL_WID column is a four digit number that is reset to 1000 each time the ETL is run and incremented by one for each date pattern type stored in W_MCAL_CAL_D.

3.5.1.4 About Mapping JD Edwards EnterpriseOne or JD Edwards World GL Accounts to Group Account Numbers

This section explains how to map Oracle JD Edwards EnterpriseOne or JD Edwards World General Ledger Accounts to Group Account Numbers, and includes the following topics:

- An overview of setting up General Ledger Accounts - see Section 3.5.1.4.1, "Overview of Mapping GL Accounts to Group Account Numbers."
- A description of how to map General Ledger Accounts to Group Account Numbers - see Section 3.5.1.4.2, "How to Map GL Account Numbers to Group Account Numbers."

- An example that explains how to add Group Account Number metrics to the Oracle BI Repository - see Section 3.5.1.4.3, "Example of Adding Group Account Number Metrics to the Oracle BI Repository."

> **Note:** It is critical that the GL account numbers are mapped to the group account numbers (or domain values) because the metrics in the GL reporting layer use these values. For a list of domain values for GL account numbers, see *Oracle Business Analytics Warehouse Data Model Reference*.

See also Section 3.1.5, "About Mapping GL Accounts to Group Account Numbers for All Source Systems" for additional information.

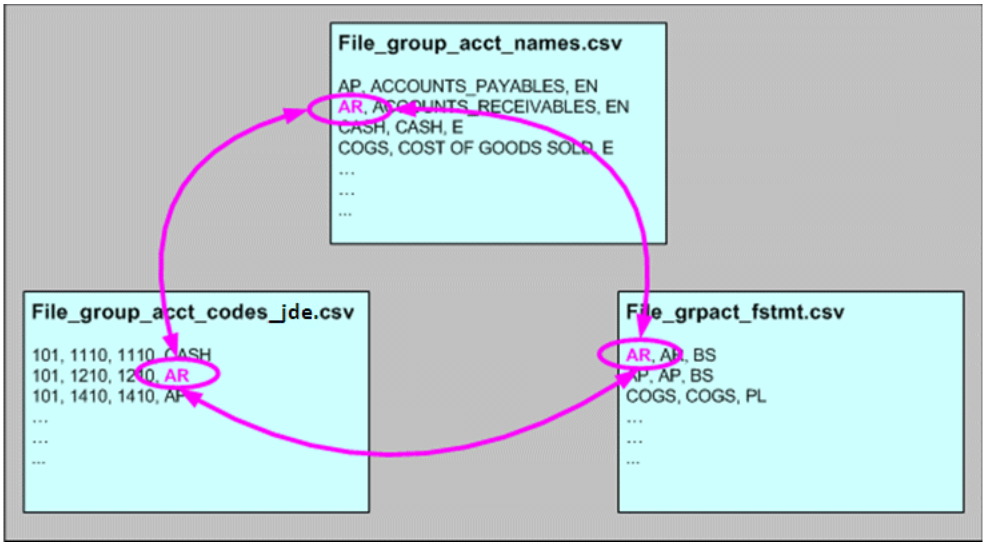
3.5.1.4.1 Overview of Mapping GL Accounts to Group Account Numbers The Group Account Number Configuration is an important step in the configuration of Financial Analytics, as it determines the accuracy of the majority of metrics in the General Ledger and Profitability module.

You set up General Ledger accounts using the following configuration files that are located in the $PMServer\SrcFiles directory:

- file_group_acct_names.csv - this file specifies the group account names and their corresponding group account codes.
- file_group_acct_codes_jde.csv - this file maps General Ledger accounts to group account codes.
- file_grpact_fstmt.csv - this file maps Financial Statement Item Codes to group account codes.

Before you load your data, you must ensure that the account values are mapped consistently across these three configuration files. Figure 3–13 shows how the GROUP_ACCOUNT_NUM field in file_group_acct_names.csv must map to the GROUP_ACCT_NUM field in file_group_acct_codes_ora.csv and the GROUP_ACCT_NUM field in file_grpact_fstmt.csv.

Figure 3–13 Configuration Files for Configuring Group Accounts for JD Edwards



File_group_acct_names.csv

```
AP, ACCOUNTS_PAYABLES, EN
AR, ACCOUNTS_RECEIVABLES, EN
CASH, CASH, E
COGS, COST OF GOODS SOLD, E
...
...
...
```

File_group_acct_codes_jde.csv

```
101, 1110, 1110, CASH
101, 1210, 1210, AR
101, 1410, 1410, AP
...
...
...
```

File_grpact_fstmt.csv

```
AR, AR, BS
AP, AP, BS
COGS, COGS, PL
...
...
...
```

You can categorize the General Ledger accounts in Oracle's JD Edwards EnterpriseOne or JD Edwards World 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 of the General Ledger accounts (for example, Cash account, AR account, Long Term Debt account Payroll account). 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).

Using the file_group_account_codes_jde.csv, you can specify which group account (among the available group accounts) the object account is associated with. The Company column in this CSV file is the actual company the object account belongs to. In addition to the From Account and To Account range, the system uses the incoming company as a parameter for the association. If the incoming company has not been configured in the group account flat file, the system inserts 00000 as the default value for Company for lookups. You can choose to not configure group accounts for any company other than 00000 if you are using a single global chart of accounts. However, if you configure group accounts for additional companies, you must configure all possible From Account and To Account ranges for these companies. In addition, you must always configure the entire range of accounts for company 00000.

Table 3–24 shows an example of the file_group_account_codes_jde.csv opened in a text editor.

Table 3–24 Example of file_group_account_codes_jde.csv

COMPANY	FROM ACCT	TO ACCT	GROUP_ACCT_NUM
00000	4100	4190	AP
00000	1200	1299	AR
00000	2120	2195	ACC DEPCN
00000	4200	4211	ACC LIAB
00000	1100	1121	CASH
00000	4900	4910	CMMN STOCK
00000	1401	1469	FG INV
00000	3990	3990	GOODWILL
00000	4690	4690	LT DEBT
00000	3900	3940	OTHER ASSET
00000	1310	1400	OTHER CA
00000	4212	4550	OTHER CL
00000	4950	4950	OTHER EQUITY
00000	4610	4685	OTHER LIAB

The file_grpact_fstmt.csv file specifies the relationship between a group account number and a Financial Statement Item code. Table 3–25 shows the Financial Statement Item codes to which Group Account Numbers must map, and their associated base fact tables.

Table 3–25 Financial Statement Item Codes and Associated Base Fact Tables

Financial Statement Item Codes	Base Fact Tables
AP	AP base fact (W_AP_XACT_F)
AR	AR base fact (W_AR_XACT_F)
COGS	Cost of Goods Sold base fact (W_GL_COGS_F)
REVENUE	Revenue base fact (W_GL_REVN_F)
TAX	Tax base fact (W_TAX_XACT_F)[1]
OTHERS	GL Journal base fact (W_GL_OTHER_F)

[1] Oracle's JD Edwards EnterpriseOne and JD Edwards World adapters for Financial Analytics do not support the Tax base fact (W_TAX_XACT_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.

3.5.1.4.2 How to Map GL Account Numbers to Group Account Numbers This section explains how to map General Ledger Account Numbers to Group Account Numbers.

> **Note:** If you add new Group Account Numbers to the file_group_acct_codes_jde.csv file, you must also add metrics to the Oracle BI repository. For more information, see Section 3.5.1.4.3, "Example of Adding Group Account Number Metrics to the Oracle BI Repository."

To map Oracle GL account numbers to group account numbers:

1. Open the file_group_acct_codes_jde.csv file with a text editor in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).
2. For each GL account number that you want to map, create a new row in the file containing the following fields:

Field Name	Description
COMPANY	The ID of the COMPANY.
FROM ACCT	The lower limit of the natural account range. This is based on the natural account segment of your GL accounts.
TO ACCT	The higher limit of the natural account range. This is based on the natural account segment of your GL accounts.
GROUP_ACCT_NUM	This field denotes the group account number of the General Ledger account, as specified in the file_group_acct_names.csv file. For example, 'AP' for Accounts Payables, 'CASH' for cash account, 'GEN PAYROLL' for payroll account, and so on.

For example:

```
10000, 1110, 1110, CASH
10000, 1210, 1210, AR
```

```
10000, 1220, 1220, AR
```

> **Note:** You can optionally remove the unused rows in the CSV file.

3. Ensure that the values that you specify in the file_group_acct_codes_jde.csv file are consistent with the values that are specified in the file_group_acct_names.csv file and the file_grpact_fstmt.csv file.

   See Figure 3–13, "Configuration Files for Configuring Group Accounts for JD Edwards" for an example.

4. Save and close the CSV file.

3.5.1.4.3 Example of Adding Group Account Number Metrics to the Oracle BI Repository If you add new Group Account Numbers to the file_group_acct_codes_jde.csv file (as described in Section 3.5.1.4.2, "How to Map GL Account Numbers to Group Account Numbers"), you must also add metrics to the Oracle BI repository to expose the new Group Account Numbers. This example illustrates how you can add Group Account Number metrics using the Oracle BI Administration Tool.

This example assumes that you have a new Group Account Number called 'Payroll,' and you want to add a new metric to the Presentation layer called 'Payroll Expense.'

To add a new metric in the logical table Fact - Fins - GL Other Posted Transaction:

1. Using the Administration Tool, open OracleBIAnalyticsApps.rpd.

   The OracleBIAnalyticsApps.rpd file is located at:

   ```
   ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_
   obisn\repository
   ```

2. In the Business Model and Mapping layer:

   a. Create a logical column named 'Payroll Expense' in the logical table 'Fact - Fins - GL Other Posted Transaction.'

      For example, right-click the Core\Fact - Fins - GL Other Posted Transaction object and choose **New Object**, then **Logical Column**, to display the Logical Column dialog. Specify Payroll Expense in the **Name** field.

   b. Display the Aggregation tab, and then choose 'Sum' in the **Default aggregation rule** drop-down list.

   c. Click **OK** to save the details and close the dialog.

   d. Expand the Core\Fact - Fins - GL Other Posted Transaction\Sources folder and double click the Fact_W_GL_OTHER_GRPACCT_FSCLYR_A source to display the Logical Table Source dialog.

   e. Display the Column Mapping tab.

   f. Select **Show unmapped columns**.

   g. Locate the Payroll Expense expression, and click the Expression Builder button to open Expression Builder.

   h. Use Expression Builder to specify the following SQL statement:

      ```
      CASE WHEN "Oracle Data Warehouse"."Catalog"."dbo".
      "Dim_W_GL_GROUP_ACCOUNT_D"."GROUP_ACCOUNT_NUM" = 'PAYROLL'
      THEN "Oracle Data Warehouse"."Catalog"."dbo".
      "Fact_Agg_W_GL_OTHER_GRPACCT_FSCLYR_A"."OTHER_GLOBAL1_AMT" ELSE NULL END
      ```

The case condition refers to the new Group Account Number 'Payroll' and uses it as a filter.

i. Repeat steps d to h for each Logical Table Source. Modify the expression in step h appropriately for each LTS by using the appropriate fact table that corresponds to the Logical Table Source.

   Steps d to h must be repeated for each Logical Table Source because in this example, there are multiple Logical Table Sources for fact table and aggregation tables in this logical table. Modify the expression in step h appropriately for each Logical Table Source by using the appropriate fact table to which it corresponds.

3. Save the details.
4. To expose the new repository objects in end users' dashboards and reports, drag the new objects from the Business Model and Mapping layer to an appropriate folder in the Presentation layer.

To add a new metric in the logical table Fact - Fins - GL Balance:

1. Using the Administration Tool, open OracleBIAnalyticsApps.rpd.

   The OracleBIAnalyticsApps.rpd file is located at:

   ```
   ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_
   obisn\repository
   ```

2. In the Business Model and Mapping layer:

   a. Create a logical column named 'Payroll Expense' in the logical table 'Fact - Fins - GL Balance.'

      For example, right-click the Core\Fact - Fins - GL Balance object and choose **New Object**, then **Logical Column**, to display the Logical Column dialog. Specify Payroll Expense in the **Name** field.

   b. In the Column Source tab, select **Derived from existing columns using an expression**.

   c. Click the Expression Builder button to display Expression Builder.

   d. Use Express Builder to specify the following SQL statement:

      ```
      FILTER("Core"."Fact - Fins - GL Balance"."Activity Amount"
      USING "Core"."Dim - GL Account"."Group Account Number" = 'Payroll')
      ```

      The case condition refers to the new Group Account Number 'Payroll.'

3. Save the details.
4. To expose the new repository objects in end users' dashboards and reports, drag the new objects from the Business Model and Mapping layer to an appropriate folder in the Presentation layer.

3.5.1.5 About Configuring GL Account Segment Hierarchies

In Oracle's JD Edwards EnterpriseOne adapter, the account segments are mapped as:

```
CO.BU.OBJ.SUB.Sub Ledger.Sub Ledger type.Category Code(x).Category Code(n).
Category Code(y)
```

Segment hierarchies are only supported for the Business Unit (BU) segment. Note that because the Business Unit hierarchy is already captured in W_INT_ORG_DH, there is no extraction (ETL) for populating W_HIERARCHY_D.

3.5.1.6 Setting Up $$JDE_RATE_TYPE

The concept of Rate Type in Oracle's JD Edwards EnterpriseOne and JD Edwards World is different than how it is defined in the Oracle Business Analytics Warehouse. In Oracle's JD Edwards EnterpriseOne and JD Edwards World, the rate type is an optional key; it is not used during exchange rate calculations.

DAC uses the $$JDE_RATE_TYPE source system parameter to populate the Rate_ Type field in the W_EXCH_RATE_GS table. By default, the $$JDE_RATE_TYPE source system parameter in DAC has a value of "Actual."

The query and lookup on W_EXCH_RATE_G will fail if the RATE_TYPE field in the W_EXCH_RATE_G table does not contain the same value as the GLOBAL1_RATE_ TYPE, GLOBAL2_RATE_TYPE 2 and GLOBAL3_RATE_TYPE fields in the W_ GLOBAL_CURR_G table.

Part III

Configuring Functional Areas

Part III contains mandatory and additional tasks required to configure the different Oracle BI Applications functional areas.

Note: Before you perform the application-specific configuration tasks described in this Part, you must first perform the steps in Chapter 3, "Configuring Common Areas and Dimensions."

Perform the tasks in the following sections for the applications you want to configure:

- Chapter 4, "Configuring Oracle Procurement and Spend Analytics"
- Chapter 5, "Configuring Oracle Financial Analytics"
- Chapter 6, "Configuring Oracle Supply Chain and Order Management Analytics"
- Chapter 7, "Configuring Oracle Human Resources Analytics"
- Chapter 8, "Configuring Oracle Sales Analytics"
- Chapter 9, "Configuring Oracle Contact Center Telephony Analytics"
- Chapter 10, "Configuring Oracle Price Analytics"
- Chapter 11, "Configuring Oracle Service Analytics"
- Chapter 12, "Configuring Oracle Marketing Analytics"
- Chapter 13, "Configuring Oracle Loyalty Analytics"
- Chapter 14, "Configuring Oracle Pharma Analytics"
- Chapter 15, "Configuring Oracle Project Analytics"

Note: Before you run Execution Plans for your Subject Areas, ensure that you have used DAC to set the appropriate Source System Parameters. To do so, in DAC, go to the Design view, and select the appropriate custom container from the drop-down list. Then, display the Source System Parameters tab.

4 Configuring Oracle Procurement and Spend Analytics

This section describes how to configure Oracle Procurement and Spend Analytics. It contains the following topics:

- Section 4.1, "Overview of Oracle Procurement and Spend Analytics"
- Section 4.2, "Configuration Required Before a Full Load for Oracle Procurement and Spend Analytics"
- Section 4.3, "Configuration Steps for Controlling Your Data Set"

4.1 Overview of Oracle Procurement and Spend Analytics

Oracle Procurement and Spend Analytics comprises Oracle Procurement and Spend Analytics and Oracle Supplier Performance Analytics.

Oracle Procurement and Spend Analytics enables 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 spend and procurement processes, including comprehensive supplier performance analysis, supplier payables analysis and employee expenses 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 integrates with the other applications in the Oracle Business Intelligence Applications product line, such as Oracle Financial Analytics. They deliver this insight across the organization to increase the company's effectiveness in managing its customers, suppliers, and financial decisions.

Oracle Procurement and Spend Analytics provides visibility into direct and indirect spending across the enterprise, payment, and employee expenses. Oracle Procurement and Spend Analytics comprises the following Subject Areas:

- Procure to Pay: This is a summary Subject Area that provides the ability to do comparative analysis and report on requested spend, committed spend and actual spend and receipts across business units, buying locations, suppliers, products, item categories 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 combines the information from Purchase Order Costs and Purchase Schedules with the ability to report on

committed spend, contract compliance and Purchase orders of the suppliers of an organization across suppliers, company, products, item categories and associated hierarchies at purchase order line level.

- Purchase Cycle Lines: This is a summary Subject Area that provides the ability to report cycle time performance, such as requisition to purchase order lead time, purchase order to receipt lead time, P2P lead time of the suppliers of an organization.
- Purchase Requisitions: This is a detailed Subject Area that provides the ability to report on requested spend and purchase requisitions (including cyclic requisitions) of the suppliers of an organization across suppliers, company, products, item categories and associated hierarchies at purchase requisition line level.
- 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, item categories and associated hierarchies at purchase receipt line level, including reporting based on receiving time.
- Employee Expenses: 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, including Approvers and cycle time measurements related to Approval, and Employee Expenses by various expense types. This Subject Area is only applicable for Oracle EBS 11.5.10 (family Pack M) and R12, PeopleSoft 8.9 and 9.0, and the Universal adapter.

  Note: For Oracle Employee Expenses (OIE), Release 7.9.6.3 support comes in two different adapters: SDE_ORA11510_Adaptor supports Oracle EBS release 11i.FIN_PF.F/11i.OIE.I minipack, and SDE_ORAR12_Adaptor supports Oracle EBS release R12.
- Invoice Lines: This is a detailed Subject Area that provides the ability to report on total spend of an organization across suppliers, products, item categories, business units, cost centers, buying locations, supplier locations and associated hierarchy. In addition, this Subject Area also provides detailed information at invoice distribution level.
- Supplier Performance. The Suppliers functional area (available with Supplier Performance Analytics) 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 (available with Supplier Performance Analytics) 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 enables organizations to maintain better relationships with their best suppliers.

  Note: In order to populate Supplier Payables component, you must implement the Accounts Payables module of Oracle Financial Analytics. If you do not implement the Accounts Payables module, then some of the Supplier Payables reports will not be populated. For more information about configuring Oracle Financial Analytics, see Chapter 5, "Configuring Oracle Financial Analytics".

Oracle Procurement and Spend Analytics provides the following dashboards:

- Spend Analyzer - enables organizations in getting complete visibility into the spend patterns of the company by consolidating the data into a single source of truth. By gaining insight, business users are empowered in identifying saving opportunities and meeting organizational goals. Spend Analyzer dashboards provides different views such as Spend by Supplier, Spend by Category, Spend by Organization with drill downs to identify the savings potential at item level.
- Procurement Performance - a re-designed dashboard that provides insight into the procure to pay process, including current status and procurement effectiveness that helps organizations to monitor the procurement performance on regular basis, identify bottlenecks, and to pro-actively take corrective actions before they become issues.
- Supplier Performance - 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 analysis of payments due and overdue.
- Employee Expenses - Enables organizations to have a complete picture of the Employee Expenses, including approval cycle times, expenses by expense type, expense reports by status. This dashboard provides different views such as Manager view, Cost center view and submissions. While Spend Analyzer provides high level spend at Invoice level, Employee expenses provides detailed analysis of how the money related to travel and expenses is being spent. This Dashboard is only applicable for Oracle EBS 11.5.10 (family Pack J) and R12, and Universal adapter.

4.2 Configuration Required Before a Full Load for Oracle Procurement and Spend Analytics

This section contains configuration steps that you must perform on Oracle Procurement and Spend Analytics before you do a full data load. It contains the following topics:

- Section 4.2.1, "Configuration Steps for Oracle Procurement and Spend Analytics for All Source Systems"
- Section 4.2.2, "Configuration Steps for Oracle Procurement and Spend Analytics for Oracle EBS"
- Section 4.2.3, "Configuration Steps for Oracle Procurement and Spend Analytics for PeopleSoft"

4.2.1 Configuration Steps for Oracle Procurement and Spend Analytics for All Source Systems

This section contains configuration steps that apply to all source systems before you do a full data load. It contains the following topic:

- Section 4.2.1.1, "About Integrating Oracle Procurement with Spend Analytics with Oracle Spend Classification"

Note: For configuration steps that apply to all BI Applications modules, see Chapter 3, "Configuring Common Areas and Dimensions."

4.2.1.1 About Integrating Oracle Procurement with Spend Analytics with Oracle Spend Classification

This section contains configuration steps that apply to Oracle Procurement and Spend Analytics when deployed with Oracle Spend Classification. For implementing Oracle Spend Classification and required patches, refer to the Oracle Spend Classification product documentation.

If you are not implementing Oracle Spend Classification, you might choose to remove or hide the Oracle Spend Classification integration metadata from the Presentation layer of the BI repository (for more information, see Section 4.2.1.1.3, "How to Remove or Hide Oracle Spend Classification Integration Metadata").

Note: Oracle Spend Classification is not part of the core Oracle BI Applications product suite, and is not packaged with any module of Oracle BI Applications. It is a separate solution offered by Oracle, and a separate license is required. If you are interested in licensing and implementing Oracle Spend Classification, contact your Oracle Sales Representative.

4.2.1.1.1 Overview to Oracle Spend Classification Integration Oracle Spend Classification is a complementary product that can be used in conjunction with Oracle Procurement and Spend Analytics to improve the accuracy of Spend by converting 'unclassified' Spend into item categories. Oracle Procurement and Spend Analytics is designed to work with or without Oracle Spend Classification.

Typical procurement systems will have many PO, Invoice, and Expense Transactions without reference to item and item categories, and in most cases they might have item descriptions in a free text format. When you implement Oracle Procurement and Spend Analytics, these transactions will come into the system as 'Unclassified' because they do not have corresponding items and/or item categories. This issue is more prominent if your organization’s Spend constitutes a major portion of Indirect Spend.

Oracle Procurement and Spend Analytics is installed with infrastructure required to feed data from the data warehouse to Oracle Spend Classification, and feed the classified data back into the data warehouse. This Infrastructure is provided as an additional feature for those customers who would like to take the advantage of both Oracle Procurement and Spend Analytics and Oracle Spend Classification.

If you choose not to use Oracle Spend Classification, Oracle Procurement and Spend Analytics can be deployed as a stand alone solution, and all the features of Procurement and Spend Analytics can be deployed without any dependency on Oracle Spend Classification.

4.2.1.1.2 About the Oracle Spend Classification Metadata This section describes the Oracle Spend Classification metadata and repository metadata that is available for use with Oracle Spend Classification.

The following facts are integrated with Oracle Data Classification to enrich and automatically assign category codes:

- W_AP_INV_DIST_F
- W_PURCH_COST_F
- W_RQSTN_LINE_COST_F

There are five types of taxonomy supported: UNSPSC, EBS Purchasing Categories, and three custom categories. The classification results are stored in these columns:

- AUTO_UNSPSC_WID
- AUTO_PURCHASING_CATEGORY_WID
- AUTO_CUSTOM_CATEGORY1_WID
- AUTO_CUSTOM_CATEGORY2_WID
- AUTO_CUSTOM_CATEGORY3_WID

In the Analytics metadata repository (RPD), the following is configured by default:

- UNSPSC, EBS Purchasing Categories, and Custom Category1 are configured up to the Business Model and Mapping layer. The facts and dimension names are as follows:
  - Fact - Spend and AP Invoice Distribution
  - Fact - Purchasing – Order
  - Fact - Purchasing – Requisition
  - Dim - Auto UNSPSC
  - Dim - Auto Purchasing Category
  - Dim - Auto Custom Category1
- In the Presentation layer, 'Procurement and Spend - Invoice Lines' contains the columns for data classification, under the following folders:
  - Data Classification
  - Auto UNSPSC
  - Auto Purchasing Category
  - Auto Custom Category 1

4.2.1.1.3 How to Remove or Hide Oracle Spend Classification Integration Metadata If you are not implementing Oracle Spend Classification, Oracle recommends that you remove or hide the Oracle Spend Classification integration metadata that is included in the Presentation layer of the BI repository as installed out-of-the-box. Hiding or deleting this metadata avoids potential confusion among business end users.

To remove or hide Oracle Spend Classification Integration Metadata:

1. Using the Administration Tool, open OracleBIAnalyticsApps.rpd.

   The OracleBIAnalyticsApps.rpd file is located at:

   ```
   ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_
   obisn\repository
   ```

2. In the Presentation layer pane, expand the folder 'Procurement and Spend - Invoice Lines.'

   The Oracle Spend Classification metadata in the Presentation layer consists of the following objects:

   Data Classification

   Auto UNSPSC

   Auto Purchasing Category

Auto Custom Category 1

3. To remove the metadata objects listed above, right-click the objects and select **Delete**.

> **Note:** If you decide later to implement Oracle Spend Classification, you need to do the following:
>
> 1. In the Business Model and Mapping layer, drag and drop the following dimensions to the 'Procurement and Spend - Invoice Lines' folder in the Presentation layer:
>
>   - Dim - Auto UNSPSC
>
>   - Dim - Auto Purchasing Category
>
>   - Dim - Auto Custom Category1

4. To hide these objects from end users, right-click each object and select **Properties**, then **Permissions**, and then select **No Access** for the appropriate application roles.

> **Note:** If you decide later to implement Oracle Spend Classification, you need to do the following:
>
> 1. To display the following objects to end users, right-click each object and select Properties, then Permissions, and then select **Read** for the appropriate application roles:
>
>   - Data Classification
>
>   - Auto UNSPSC
>
>   - Auto Purchasing Category
>
>   - Auto Custom Category 1

5. Save and close the repository.

4.2.1.1.4 How to Deploy UNSPSC, EBS Purchasing Categories, and Custom Category1

Follow these steps if you want to expose UNSPSC, EBS Purchasing Categories, and Custom Category1 for your Purchase Order and Purchase Requisition Subject Area.

To deploy UNSPSC, EBS Purchasing Categories, and Custom Category1:

1. Using the Administration Tool, open OracleBIAnalyticsApps.rpd.

   The OracleBIAnalyticsApps.rpd file is located at:

   ```
   ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_
   obisn\repository
   ```

2. In the Presentation layer, do the following:

   a. Expand the folder "Procurement and Spend - Invoice Lines".

   b. Multi-select the following folders and right-click to copy:

      Data Classification

      Auto UNSPSC

      Auto Purchasing Category

      Auto Custom Category 1

   c. To implement Oracle Spend Classification in Purchase Orders, select the folder "Procurement and Spend - Purchase Orders" and right-click to paste in the folders.
   d. To implement Oracle Spend Classification in Purchase Requisitions, select the folder "Procurement and Spend - Purchase Requisitions" and right-click to paste in the selected folders.
   e. Verify the new folders.
   f. If required, reorder the folders as you would like the folders to be displayed to business users in the Presentation Services catalog.

3. Save and close the repository.

4.2.1.1.5 How to Deploy the Additional Custom Category2 and Custom Category3 This section explains how to deploy Custom Category2 and Custom Catogory3.

Note: This task uses the Fact_W_AP_INV_DIST_F fact as an example, though you can also apply the steps to deploy other facts.

To deploy Custom Category2 and Custom Catogory3:

1. Using the Administration Tool, open OracleBIAnalyticsApps.rpd.

   The OracleBIAnalyticsApps.rpd file is located at:

   ```
   ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_
   obisn\repository
   ```

2. In the Physical layer, do the following:
   a. Right-click "Dim_W_PROD_CAT_DH_AUTO_CUSTOM_CATEGORY1" under "Oracle Data Warehouse" and select Duplicate.
   b. Rename it to "Dim_W_PROD_CAT_DH_AUTO_CUSTOM_CATEGORY2".
   c. Join dimension "Dim_W_PROD_CAT_DH_AUTO_CUSTOM_CATEGORY2" and fact "Fact_W_AP_INV_DIST_F" using the following condition:

      ```
      Dim_W_PROD_CAT_DH_AUTO_CUSTOM_CATEGORY2.ROW_WID = Fact_W_AP_INV_DIST_
      F."AUTO_CUSTOM_CATEGORY2_WID
      ```

3. In the Business Model and Mapping layer, do the following:
   a. Immediately below table "Dim - Auto Custom Category1", create "Dim - Auto Custom Category2".
   b. Immediately below hierarchy "Auto Custom Category1", create "Dim - Auto Custom Category2" based on the physical table "Dim_W_PROD_CAT_DH_AUTO_CUSTOM_CATEGORY2".
   c. Join "Dim - Auto Custom Category1" to "Fact - Spend and AP Invoice Distribution".
   d. Edit "Fact - Spend and AP Invoice Distribution". Fact_W_AP_INV_DIST_F. Display the Content tab, and set the level of "Auto Custom Category2" to "Custom Hierarchy Base Level".

4. In the Presentation layer, do the following:
   a. Create a sub-folder called "Auto Custom Category 2" in the "Procurement and Spend - Invoice Lines" folder. Edit folder and add this exact string to the Description box.

      Auto Custom Category2 becomes a sub-folder of Data Classification.

   b. Order this folder so that it is after "Auto Custom Category 1".
   c. Drag the "Dim - Auto Custom Category1" columns from the Business Model and Mapping layer into the "Auto Custom Category 2" folder in the Presentation layer.
5. Save and close the repository.
6. Repeat steps 2 - 5 for Custom Category3.

4.2.2 Configuration Steps for Oracle Procurement and Spend Analytics for Oracle EBS

This section contains configuration steps that apply to Oracle EBS before you do a full data load. It contains the following topics:

- Section 4.2.2.1, "How to Configure the DAC Parameter for Purchase Cycle Line"
- Section 4.2.2.2, "Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics"
- Section 4.2.2.3, "How to Configure AP Invoice Types for Spend Analysis"
- Section 4.2.2.4, "Configuration Specific to Certain Versions of Oracle EBS"

4.2.2.1 How to Configure the DAC Parameter for Purchase Cycle Line

To load the purchase cycle line table (W_PURCH_CYCLE_LINE_F), the ETL tasks need to distinguish between data coming from Oracle EBS applications and data that might be coming from other applications such as PeopleSoft. This section applies to all versions of Oracle EBS.

To configure the DAC parameter for purchase cycle line:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab, and query for task SIL_PurchaseCycleLinesFact.
3. Display the Parameters tab in the lower pane.
4. Edit the value of the parameter $$ORA_DATASOURCE_NUM_ID_LIST and specify a comma-separated list of Data Source NUM IDs that you defined for your Oracle data sources.

   You specify Data Source NUM IDs on the DAC\Setup\Physical Data Sources tab in the **Data Source Number** field for each data source.
5. Repeat the same steps for task SIL_PurchaseCycleLinesFact_Extract.

For more information about DAC, see *Oracle Business Intelligence Data Warehouse Administration Console Guide*.

4.2.2.2 Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics

If you have modified or extended 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 from the seeded values, you need to follow the instructions for configuring the domain values and CSV worksheet files in this section.

This section applies to all versions of Oracle EBS. See also Section 4.2.2.4, "Configuration Specific to Certain Versions of Oracle EBS," which describes CSV files not included in this list.

4.2.2.2.1 List of Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics

Table 4–1 lists the domain values for Oracle Procurement and Spend Analytics in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles). The table also lists the CSV worksheet source files for Oracle Procurement and Spend Analytics in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).

Table 4–1 Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics

Worksheet File Name	Description	Session
domainValues_Employee_Expense_Status.csv	Used to configure the Employee Expense Approval Status and the corresponding domain values for the Oracle application. For information about how to edit this file, see Section 4.2.2.2.13, "How to Configure domainValues_Employee_Expense_Status.csv."	SDE_ORA_EmployeeExpenseStatusDimension
domainValues_Employee_Expense_Type_ora11i.csv	Used to identify the expense transaction type, such as AIRFARE, ACCOMODATION, and so on. For information about how to edit this file, see Section 4.2.2.2.10, "How to Configure domainValues_Employee_Expense_Type_ora11i.csv."	SDE_ORA_EmployeeExpenseTypeDimension
domainValues_ShipmentType_ora12.csv	List the Shipment Type and the corresponding domain values for the Oracle 11i application. For information about how to edit this file, see Section 4.2.2.2.8, "How to Configure domainValues_ShipmentType_ora12.csv."	SDE_ORA_TransactionTypeDimension_PO_Shipment_Type
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 4.2.2.2.2, "How 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 4.2.2.2.3, "How to Configure domainValues_Status_Purch_Cycle_ora11i.csv."	SDE_ORA_StatusDimension_PurchaseCycle
domainValues_Xact_Sources_APSources_ora.csv	Used to identify whether an AP invoice source is considered manual or electronic. For information about how to edit this file, see Section 4.2.2.2.12, "How to Configure domainValues_Xact_Sources_APSources_ora.csv."	SDE_ORA_TransactionSourceDimension_AP_CC_Extract, SDE_ORA_TransactionSourceDimension_AP_LKP_Extract
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 4.2.2.2.7, "How 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 4.2.2.2.6, "How to Configure domainValues_Xact_Types_PO_Line_Type_ora11i.csv."	SDE_ORA_TransactionTypeDimension_PO_Line_Type

Table 4–1 (Cont.) Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics

Worksheet File Name	Description	Session
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 4.2.2.2.4, "How 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 4.2.2.2.5, "How to Configure domainValues_Xact_Types_Purch_Requisitions_ora11i.csv."	SDE_ORA_TransactionTypeDimension_PurchaseRequest
domainValues_Xact_Types_PurchaseRequisition.csv	Used to identify whether the purchase order creation method is manual or auto. For information about how to edit this file, see Section 4.2.2.2.9, "How to Configure domainValues_Xact_Types_PurchaseRequisition.csv." **Note**: See also the file domainValues_Xact_Types_PurchaseReceipt.csv, which is used to configure Receipt transaction types.	SDE_ORA_TransactionTypeDimension_PurchaseRequisition
file_employee_expense_payment_method.csv	Lists the Employee Expenses Payment Method codes and the corresponding domain values for the Oracle application. This file contains both source values and domain values for the Employee Expenses Payment Method. The values of CASH and CREDIT_CARD have been defined for the Cash Payment and Payment through Credit Card. No configuration should be required for this file.	SDE_ORA_EmployeeExpensePaymentMethodDimension
file_Expense_pay_type.csv	Lists the Employee Expenses type codes and the corresponding domain values for the Oracle application. This file contains both source value and domain value for the Employee Expenses Payment Type. The value 'E' has been defined for the Actual Expense Incurred By The Requestor. No configuration should be required for this file.	SDE_ORA_TransactionTypeDimension_Expense_Pay_Type
File_StatusDimension_PurchaseRequisition.csv	Lists the Purchasing Requisition Status column and the corresponding domain values for the Oracle EBS application.	SDE_ORA_StatusDimension_PurchaseRequisition
file_xact_type_AgreementLeverageType.csv	Used to identify the agreement leverage types. For information about how to edit this file, see Section 4.2.2.2.11, "How to Configure file_xact_type_AgreementLeverageType.csv."	SDE_ORA_TransactionTypeDimension_AgreementLeverageType
domainValues_Status_APInvoicePaymentStatus_ora11i.csv	Lists the AP Invoice Payment statuses and the corresponding domain values for the Oracle EBS Application.The values of NOT PAID,PARTIALLY PAID,FULLY PAID have been defined for the source values of 'N','P','Y' respectively. No configuration should be required for this file.	SDE_ORA_StatusDimension_APInvoicePaymentStatus
file_xact_type_AgreementLeverageType_2.csv	Configures the value of W_XACT_TYPE_CODE2 to identify types of non-agreement purchase. The values Others, Leakage, and Potential Leakage have been defined for Non-agreement. No configuration should be required for this file.	SDE_ORA_TransactionTypeDimension_AgreementLeverageType

Table 4–1 (Cont.) Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics

Worksheet File Name	Description	Session
file_Consigned_Type.csv	Lists the Purchase Order Consigned Code Transaction Type column and the corresponding domain values for the Oracle EBS application. The Consigned Code CONSIGNED, REGULAR, and CONSIGNED-CONSUMED have been identified as different consigned types. No configuration should be required for this file.	SDE_ORA_TransactionType Dimension_ConsignedCode
file_APInvoice_ Approval_Status.csv	Lists the AP invoice approval status codes and the corresponding domain values for the Oracle EBS Application. This file contains both source values and domain values for AP Invoice Approval status. The values of CANCELLED, ON HOLD, REQUIRES APPROVAL, and APPROVED have been defined for the payables invoices. No configuration should be required for this file.	SDE_ORA_ StatusDimension_ APInvoiceApproval Status
file_APInvoice_spend_ type.csv	Lists the AP Invoice distribution types and the corresponding domain values for the spend types for the Oracle EBS Application. This file contains both source values and domain values for the AP Invoice distribution spend types. The values of PO MATCHED, PAYABLES LEAKAGE, and PO NOT REQUIRED have been defined for the distribution types of ITEM, TAX, FREIGHT, and MISCELLANEOUS. No configuration should be required for this file.	SDE_ORA_TransactionType Dimension_APSpend
domainValues_Xact_ Types_DocTypes_ ora11i.csv	The file lists the Transaction types and the corresponding domain values for the Oracle EBS application. The PAYABLES related values are considered. The source invoice type and source invoice distribution type is defined, and corresponding domain values are mapped. No configuration should be required for this file.	SDE_ORA_TransactionType Dimension_APDerive
file_ap_invoice_config_ spend_ora12.csv	This file lists the AP invoice types and corresponding designation on whether to include in the spend calculation. For information about how to edit this file, see Section 4.2.2.3, "How to Configure AP Invoice Types for Spend Analysis."	SDE_ORA_APInvoice DistributionFact

4.2.2.2.2 How to Configure domainValues_Status_Purch_Approve_ora11i.csv This section explains how to configure the domainValues_Status_Purch_Approve_ora11i.csv file.

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. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), 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.

4.2.2.2.3 How to Configure domainValues_Status_Purch_Cycle_ora11i.csv

This section explains how to configure the domainValues_Status_Purch_Cycle_ora11i.csv file.

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. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open domainValues_Status_Purch_Cycle_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 Cycle (PURCH_CYCLE) domain value.

   Use commas to separate the entries.

5. Save and close the file.

4.2.2.2.4 How to Configure domainValues_Xact_Types_Purch_Orders_ora11i.csv

This section explains how to configure the domainValues_Xact_Types_Purch_Orders_ora11i.csv file.

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. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open domainValues_Xact_Types_Purch_Orders_ora11i.csv file in a text editor.

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.

4.2.2.2.5 How to Configure domainValues_Xact_Types_Purch_Requisitions_ora11i.csv

This section explains how to configure the domainValues_Xact_Types_Purch_Requisitions_ora11i.csv file.

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. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open domainValues_Xact_Types_Purch_Requisitions_ora11i.csv file in a text editor.
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.

4.2.2.2.6 How to Configure domainValues_Xact_Types_PO_Line_Type_ora11i.csv This section explains how to configure the domainValues_Xact_Types_PO_Line_Type_ora11i.csv file.

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. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open domainValues_Xact_Types_PO_Line_Type_ora11i.csv file in a text editor.
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.

4.2.2.2.7 How to Configure domainValues_Xact_Types_PO_Line_Type_CODE1_ora11i.csv This section explains how to configure the domainValues_Xact_Types_Code1_PO_Line_Type_ora11i.csv file.

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. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open domainValues_Xact_Types_Code1_PO_Line_Type_ora11i.csv file in a text editor.

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.

4.2.2.2.8 How to Configure domainValues_ShipmentType_ora12.csv

This section explains how to configure the domainValues_ShipmentType_ora12.csv file. This file is only applicable to Oracle R12.

1. Identify the Purchase Order Shipment Type in your Oracle EBS source system by using the following SQL:

```
SELECT DISTINCT PO_LOOKUP_CODES.LOOKUP_CODE
FROM PO_LOOKUP_CODES
WHERE PO_LOOKUP_CODES.LOOKUP_TYPE = 'SHIPMENT TYPE'
AND PO_LOOKUP_CODES.LOOKUP_CODE
IN ('STANDARD', 'PREPAYMENT', 'SCHEDULED', 'BLANKET')
```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open domainValues_PaymentType_ora12.csv file in a text editor.

3. Copy the LOOKUP_CODE to the XACT_TYPE_CODE column in the file. The data must be copied starting from the second line.

4. Map each LOOKUP_CODE to one PO Shipment Type (PO_SHIPMENT_TYPE) domain value. Use commas to separate the entries.

5. Save and close the file.

4.2.2.2.9 How to Configure domainValues_Xact_Types_PurchaseRequisition.csv

This section describes how to configure the domainValues_Xact_Types_PurchaseRequisition.csv file.

1. Identify the different PO creation methods in your Oracle EBS source system by using the following SQL:

```
SELECT DISTINCT FND_LOOKUP_VALUES.LOOKUP_CODE FROM FND_LOOKUP_VALUES
WHERE FND_LOOKUP_VALUES.LOOKUP_TYPE = 'PO_DOCUMENT_CREATION_METHOD' AND
FND_LOOKUP_VALUES.LANGUAGE = 'US'
```

   Note: The language code must be set for non-English deployments.

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the file domainValues_Xact_Types_PurchaseRequisition.csv in a text editor.

3. Copy the columns LOOKUP_CODE to XACT_TYPE_CODE starting from the second line of data.

4. Map each LOOKUP_CODE to either the MANUAL or AUTO domain value. Use commas to separate the entries.

5. Save and close the file.

4.2.2.2.10 How to Configure domainValues_Employee_Expense_Type_ora11i.csv

This section describes how to configure the domainValues_Employee_Expense_Type_ora11i.csv file.

1. Identify the different transaction types of purchase receipts in your Oracle EBS source system by using the following SQL:

   ```
   SELECT DISTINCT FND.LOOKUP_CODE FROM FND_LOOKUP_VALUES FND
   WHERE LOOKUP_TYPE = 'OIE_EXPENSE_CATEGORY' AND
   LANGUAGE = USERENV('LANG')
   ```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the file domainValues_Employee_Expense_Type_ora11i.csv in a text editor.
3. Copy the columns LOOKUP_CODE to XACT_TYPE_CODE starting from the second line of data.
4. Map each LOOKUP_CODE to one of the respective EXPENSE_TYPE domain values. Use commas to separate the entries.
5. Save and close the file.

4.2.2.2.11 How to Configure file_xact_type_AgreementLeverageType.csv This section describes how to configure the file_xact_type_AgreementLeverageType.csv file.

1. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the file file_xact_type_AgreementLeverageType.csv in a text editor.
2. Configure the value of W_XACT_TYPE_CODE1.

   XACT_TYPE_CODE and XACT_TYPE _DESC list the seven cases of business scenarios. Scenarios one through six use some kind of negotiation. Scenario seven does not use negotiation.

XACT_TYPE_CODE	XACT_TYPE_DESC
1. Purchase releases	Orders or releases against local blanket agreement, global blanket agreement or planned orders.
2. Punch-out and catalog purchase	Purchases through the catalog or through a "punch-out" from the catalog to an external supplier site.
3. Contract reference	Orders against a contract agreement.
4. Sourcing document reference	Orders against a sourcing document.
5. Complex work	Complex work procurement.
6. Other negotiated	Other negotiated purchases as indicated by the buyer.
7. Non-negotiated	Non-agreement purchases without existing matching blanket purchase agreement.

3. In the preconfigured application, scenarios one through six are considered to be agreement purchases (W_XACT_TYPE_CODE1='Agreement'). If your business rule does not consider one of these scenarios as agreement purchases, you can change the value to 'Non-Agreement'.

4.2.2.2.12 How to Configure domainValues_Xact_Sources_APSources_ora.csv This section describes how to configure the domainValues_Xact_Sources_APSources_ora.csv file.

> **Note:** Oracle Procurement and Spend Analytics and Oracle Financial Analytics share the same configuration file.

1. Identify the different transaction types of payable transaction sources in your Oracle EBS source system by using the following SQL:

   ```
   SELECT DISTINCT UPPER(SOURCE) FROM AP_INVOICES_ALL
   ```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the file domainValues_Xact_Sources_APSources_ora.csv in a text editor.
3. Copy the results of the query in step 1 to the SOURCE_NAME column in the file, starting from the second line of data.
4. Map each source name to the respective domain values.

   Map each source name to the respective domain values. There are two domain values for each source. The first domain value specifies whether the source is a CREDIT_CARD application. Values are CREDIT_CARD or OTHER. The second domain value specifies whether the application is for MANUAL or ELECTRONIC entry. Values are MANUAL or ELECTRONIC.
5. Save and close the file.

4.2.2.2.13 How to Configure domainValues_Employee_Expense_Status.csv This section describes how to configure the domainValues_Employee_Expense_Status.csv file.

1. Identify the Approval Status codes in your Oracle EBS source system by using the following SQL:

   ```
   SELECT LOOKUP_CODE FROM FND_LOOKUP_VALUES
   WHERE FND_LOOKUP_VALUES.LOOKUP_TYPE= 'EXPENSE REPORT STATUS'
   AND FND_LOOKUP_VALUES.LANGUAGE = USERENV('LANG')
   ```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the file domainValues_Employee_Expense_Status.csv in a text editor.
3. Copy the columns LOOKUP_CODE to STATUS_CODE.

   The data must be copied starting from the 8th line.
4. Map each LOOKUP_CODE to one W_STATUS_CODE domain value.

   Use commas to separate the entries.
5. Save and close the file.

4.2.2.3 How to Configure AP Invoice Types for Spend Analysis

The configuration file for AP invoice types for Spend Analysis is file_ap_invoice_config_spend_ora12.csv in $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles). This file lists the AP invoice types and corresponding designation on whether to include in the spend calculation. The lists show all types for Oracle EBS R12, which are a superset of oracle EBS 11i. If an invoice type is not listed, it will not be included in spend calculation. Prepayment and AWT invoice types are not included in spend calculation, which is not configurable.

This section applies to all versions of Oracle EBS.

To configure file_ap_invoice_config_spend_ora12.csv:

1. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open file_ap_invoice_config_spend_ora12.csv file in a text editor.

2. Update the Y/N flag to designate whether to include invoice types in spend calculations.
3. Save and close the file.

4.2.2.4 Configuration Specific to Certain Versions of Oracle EBS

This section describes configuration that must be performed only for certain Oracle EBS versions. This section contains the following topics:

- Section 4.2.2.4.1, "How to Configure domainValues_PaymentType_ora12.csv"
- Section 4.2.2.4.2, "How to Configure Oracle Employee Expense Analytics for Oracle EBS 11.5.10"

4.2.2.4.1 How to Configure domainValues_PaymentType_ora12.csv This section explains how to configure the domainValues_PaymentType_ora12.csv file. This file lists the Payment Type and the corresponding domain values for the Oracle EBS R12 application. This file is only applicable to Oracle R12.

Note that the session for this file is SDE_ORA_TransactionTypeDimension_PO_Payment_Type.

1. Identify the Purchase Order Payment Type in your Oracle EBS source system by using the following SQL:

```
SELECT PO_LOOKUP_CODES.LOOKUP_CODE
FROM PO_LOOKUP_CODES
WHERE PO_LOOKUP_CODES.LOOKUP_TYPE = 'PAYMENT TYPE'
```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the domainValues_PaymentType_ora12.csv file in a text editor.
3. Copy the LOOKUP_CODE to the XACT_TYPE_CODE column in the file.

   The data must be copied starting from the second line.
4. Map each LOOKUP_CODE to one PO Payment Type (PO_PAYMENT_TYPE) domain value.

   Use commas to separate the entries.
5. Save and close the file.

4.2.2.4.2 How to Configure Oracle Employee Expense Analytics for Oracle EBS 11.5.10 If you want to implement Oracle Employee Expense Analytics with Oracle EBS 11.5.10, you need to have at least 11i.FIN_PF.F/11i.OIE.I family pack and above patch set level. In addition, if your 11.5.10 patch set level is on or above 11i.OIE.K family pack, then you need to follow these configuration steps.

To configure Oracle Employee Expenses for Oracle EBS 11.5.10:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab, and query for task SDE_ORA_EmployeeExpenseFact.
3. In the lower pane, display the Edit tab, and change the value for 'Command for Incremental Load' from SDE_ORA_EmployeeExpenseFact to SDE_ORA_EmployeeExpenseFact_FP.
4. On the Edit tab, change the value for 'Command for Full Load' from SDE_ORA_EmployeeExpenseFact_Full to SDE_ORA_EmployeeExpenseFact_FP_Full.

5. Save the changes.
6. Display the Subject Areas tab, and query for the 'Employee Expenses' Subject Area.
7. Select the queried record for 'Employee Expenses', and click Assemble to assemble the Subject Area.
8. Go to Execute view, select your Execution Plan that contains the 'Employee Expenses' Subject Area, then click 'Build' to build the Execution Plan.

   The Execution Plan is now ready to Execute.

4.2.3 Configuration Steps for Oracle Procurement and Spend Analytics for PeopleSoft

This section contains configuration steps that apply to PeopleSoft before you do a full data load. It contains the following topics:

- Section 4.2.3.1, "Domain Values and CSV Worksheet Files for PeopleSoft Procurement and Spend Analytics"
- Section 4.2.3.2, "CSV Worksheet Source Files for PeopleSoft Procurement and Spend Analytics"
- Section 4.2.3.3, "Enabling Reporting on Product/Purchasing Categories for PeopleSoft Procurement and Spend Analytics"

4.2.3.1 Domain Values and CSV Worksheet Files for PeopleSoft Procurement and Spend Analytics

If you have to modify or extend a seeded list of values, you must configure the CSV files for PeopleSoft 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. Compare the lists of values with the seeded values and if the lists of values are different than the seeded values, you need to follow the instructions to configure the Domain Values and CSV Worksheet Files.

> **Note:** Some sessions may fail if these procedures are not compiled in the database before running the workflows. If you have problems deploying the stored procedures, see your database reference guide, or contact your database administrator.

Table 4–2 lists the domain value csv files for PeopleSoft Procurement and Spend Analytics in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

Table 4–2 Domain Values Files for PeopleSoft Procurement and Spend Analytics

Worksheet File Name	Description	Session
domainValues_Status_Purch_Approve_psft.csv	Lists the Purchasing Approval Status column and the corresponding domain values for the PeopleSoft application. For information about how to edit this file, see Section 4.2.3.1.1, "How to Configure domainValues_Status_Purch_Approve_psft.csv."	SDE_PSFT_StatusDimension_PurchApprove
domainValues_Status_Purch_Cycle_psft.csv	Lists the Purchasing Cycle Status column and the corresponding domain values for the PeopleSoft application. For information about how to edit this file, see Section 4.2.3.1.2, "How to Configure domainValues_Status_Purch_Cycle_psft.csv."	SDE_PSFT_StatusDimension_PurchCycle
domainValues_Xact_Types_Procurement_psft.csv	Lists the Purchasing Line Type and the corresponding domain Values for the PeopleSoft application. For information about how to edit this file, see Section 4.2.3.1.3, "How to Configure domainValues_Xact_Types_Procurement_psft.csv - PO Line Type."	SDE_PSFT_Transaction TypeDimension_PO_Line_Type
domainValues_Xact_Types_Procurement_psft.csv	Lists the Purchase Order Transaction Type column and the corresponding domain values for the PeopleSoft application. For information about how to edit this file, see Section 4.2.3.1.4, "How to Configure domainValues_Xact_Types_Procurement_psft.csv - Purchase Order."	SDE_PSFT_Transaction TypeDimension_PurchaseOrder
domainValues_Xact_Types_Procurement_psft.csv	Lists the Purchase Requisition Transaction Type column and the corresponding domain values for the PeopleSoft application. The Purchase Requisition in PeopleSoft does not have a type defined. Two records have been added to the domain values to designate whether the requisition is on hold. No configuration should be needed for this file.	SDE_PSFT_Transaction TypeDimension_PurchaseRequestion
domainValues_Xact_Types_Procurement_psft.csv	Lists the Purchase Receipt Transaction Type column and the corresponding domain values for the PeopleSoft application. The Purchase Receipt in PeopleSoft does not have a type defined. Two records have been added to the domain values file to designate whether the Receipt is a receipt or is a Return to Vendor. No configuration should be needed for this file.	SDE_PSFT_Transaction TypeDimension_PurchaseReceipt
domainValues_Xact_Types_Procurement_psft.csv	Lists the Purchase Order Consigned Code Transaction Type column and the corresponding domain values for the PeopleSoft application. The Consigned Code in PeopleSoft does not have a type defined. Two records have been added to the domain values file to designate whether consigned or not, Y/N. No configuration should be needed for this file.	SDE_PSFT_Transaction TypeDimension_Consigned_Code
domainValues_Status_APInvoice_psft.csv	Lists the AP Invoice Approval Status column and the corresponding domain values for the PeopleSoft application. For information about how to edit this file, see Section 4.2.3.1.5, "How to Configure domainValues_Status_APInvoice_psft.csv - Approval Status."	SDE_PSFT_StatusDimension_APInvoiceApproval
domainValues_Xact_Source_AP_Invoice_psft.csv	Lists the AP Invoice Transaction Source column and the corresponding domain values for the PeopleSoft application. For information about how to edit this file, see Section 4.2.3.1.6, "How to Configure domainValues_Xact_Source_AP_Invoice_psft.csv."	SDE-PSFT_Transaction SourceDimension_APInvoice

Table 4–2 (Cont.) Domain Values Files for PeopleSoft Procurement and Spend Analytics

Worksheet File Name	Description	Session
domainValues_Xact_Types_APInvoice_DistributionTypes_psft.csv	Lists the AP Invoice Distribution Type Transaction Type column and the corresponding domain values for the PeopleSoft application. Distribution types of Freight, AWT, Item, Miscellaneous, Tax, and Prepay have been added to the domain values file. No configuration should be needed for this file.	SDE_PSFT_TransactionTypeDimension_APInvoiceDistributionType
file_xact_type_AgreementLeverageType_psft.csv	Lists the agreement leverage types and the corresponding domain values for the PeopleSoft application. For more information about how to edit this file, see Section 4.2.3.1.7, "How to Configure file_xact_type_AgreementLeverageType_psft.csv."	SDE_PSFT_TransactionTypeDimension_AgreementLeverageType
domainValues_Xact_Types_Expense_psft.csv	Lists the different expense types and the corresponding domain values for the PeopleSoft application. For more information, see Section 4.2.3.1.8, "How to Configure the domainValues_Xact_Types_Expense_psft.csv."	SDE_PSFT_TransactionTypeDimension_EmployeeExpense
domainValues_Xact_Types_ExpensePay_psft.csv	Lists the expense pay types and the corresponding domain values for the PeopleSoft application. For information about how to edit this file, see Section 4.2.3.1.9, "How to Configure the domainValues_Xact_Types_ExpensePay_psft.csv."	SDE_PSFT_TransactionTypeDimension_ExpensePayType
domainValues_Status_Expense_psft.csv	Lists the expense statuses and the corresponding domain values for the PeopleSoft application. For information about how to edit this file, see Section 4.2.3.1.10, "How to Configure the domainValues_Status_Expense_psft.csv."	SDE_PSFT_StatusDimension_EmployeeExpense
domainValues_Payment_Method_psft.csv	Lists the expense payment methods and the corresponding domain values for the PeopleSoft application. For information about how to edit this file, see Section 4.2.3.1.11, "How to Configure the domainValues_Payment_Method_psft.csv."	SDE_PSFT_PaymentMethodDimension_EmployeeExpense

4.2.3.1.1 How to Configure domainValues_Status_Purch_Approve_psft.csv This section explains how to configure the domainValues_Status_Purch_Approve_psft.csv file.

1. Identify the Purchase Approval Status for Purchase Orders and Purchase Requisitions in your PeopleSoft source system by using the following SQL:

```
SELECT A.FIELDVALUE, A.XLATLONGNAME
FROM PSXLATITEM A
WHERE A.EFFDT = (SELECT MAX (C.EFFDT) FROM PSXLATITEM C WHERE
C.FIELDNAME = A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE)
AND A.FIELDNAME IN('PO_STATUS','CURR_STATUS')
ORDER BY 1
```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open domainValues_Status_Purch_Approve_psft.csv file in a text editor.
3. Copy the FIELDVALUE to the STATUS_CODE column in the file. The data must be copied starting from the 9th line.
4. Map each FIELDVALUE to one Purchase Approval (PURCH_APPROVAL) domain value. Use commas to separate the entries.
5. Save and close the file.

4.2.3.1.2 How to Configure domainValues_Status_Purch_Cycle_psft.csv

This section explains how to configure the domainValues_Status_Purch_Cycle_psft.csv file.

1. Identify the Purchase Cycle Status in your PeopleSoft source system by using the following SQL:

   ```
   SELECT A.FIELDVALUE, A.XLATLONGNAME
   FROM PSXLATITEM A
   WHERE A.EFFDT = (SELECT MAX (C.EFFDT) FROM PSXLATITEM C WHERE
   C.FIELDNAME = A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE)
   AND A.FIELDNAME IN('PO_STATUS','CURR_STATUS')
   ORDER BY 1
   ```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open domainValues_Status_Purch_Cycle_psft.csv file in a text editor.
3. Copy the FIELDVALUE to the STATUS_CODE column in the file. The data must be copied starting from the 9th line.
4. Map each FIELDVALUE to one Purchase Cycle (PURCH_CYCLE) domain value. Use commas to separate the entries.
5. Save and close the file.

4.2.3.1.3 How to Configure domainValues_Xact_Types_Procurement_psft.csv - PO Line Type

This section explains how to configure the domainValues_Xact_Types_Procurement_psft.csv file for the PO Line Type transaction type.

1. Identify the Purchase Order Line Type in your PeopleSoft source system by using the following SQL:

   ```
   SELECT A.FIELDVALUE, A.XLATLONGNAME
   FROM PSXLATITEM A
   WHERE A.EFFDT = (SELECT MAX (C.EFFDT) FROM PSXLATITEM C WHERE
   C.FIELDNAME = A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE)
   AND A.FIELDNAME =('POAMOUNT_YN')
   ORDER BY 1
   ```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open domainValues_Xact_Types_Procurement_psft.csv file in a text editor.
3. Copy the FIELDVALUE to the XACT_SUBTYPE_CODE column in the file. The data must be copied starting from the 11th line.
4. Map each FIELDVALUE to one PO Line Type transaction type (PO_LINE_TYPE) domain value. Use commas to separate the entries.
5. Save and close the file.

4.2.3.1.4 How to Configure domainValues_Xact_Types_Procurement_psft.csv - Purchase Order

This section explains how to configure the domainValues_Xact_Types_Procurement_psft.csv file for the Purchase Order transaction type.

1. Identify the Purchase Order Types in your PeopleSoft source system by using the following SQL:

   ```
   SELECT A.FIELDVALUE, A.XLATLONGNAME
   FROM PSXLATITEM A
   WHERE A.EFFDT = (SELECT MAX (C.EFFDT) FROM PSXLATITEM C WHERE
   C.FIELDNAME = A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE)
   AND A.FIELDNAME =('PO_TYPE')
   ```

```
ORDER BY 1
```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open domainValues_Xact_Types_Procurement_psft.csv file in a text editor.
3. Copy the FIELDVALUE to the XACT_SUBTYPE_CODE column in the file. The data must be copied starting from the 11th line.
4. Map each FIELDVALUE to one Purchase Order transaction type (PURCH_ORDERS) domain value. Use commas to separate the entries.
5. The delivered domain values file has values for GEN, defined as STANDARD, and KAN, defined as PLANNED, purchase order types and an additional type for BLANKET.
6. Save and close the file.

4.2.3.1.5 How to Configure domainValues_Status_APInvoice_psft.csv - Approval Status This section explains how to configure the domainValues_Status_APInvoice_psft.csv file.

1. Identify the AP Invoice Approval Status in your PeopleSoft source system by using the following SQL:

   ```
   SELECT A.FIELDVALUE, A.XLATLONGNAME
   FROM PSXLATITEM A
   WHERE A.EFFDT = (SELECT MAX (C.EFFDT) FROM PSXLATITEM C WHERE
   C.FIELDNAME = A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE)
   AND A.FIELDNAME =('APPR_STATUS')
   ORDER BY 1
   ```

   In addition, the domain values file contains an entry for a cancelled status. This entry is defined as 'X' which is an 'ENTRY_STATUS' value.
2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the domainValues_Status_Purch_Approve_psft.csv file in a text editor.
3. Copy the FIELDVALUE to the STATUS_CODE column in the file. The data must be copied starting from the 10th line.
4. Map each FIELDVALUE to one Approval (APPROVAL) domain value. Use commas to separate the entries.
5. Save and close the file.

4.2.3.1.6 How to Configure domainValues_Xact_Source_AP_Invoice_psft.csv This section explains how to configure the domainValues_Xact_Types_Procurement_psft.csv file for the AP Invoice transaction sources.

1. Identify the AP Invoice source type in your PeopleSoft source system by using the following SQL:

   ```
   SELECT A.FIELDVALUE, A.XLATLONGNAME
   FROM PSXLATITEM A
   WHERE A.EFFDT = (SELECT MAX (C.EFFDT) FROM PSXLATITEM C WHERE
   C.FIELDNAME = A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE)
   AND A.FIELDNAME IN('VCHR_SRC')
   ORDER BY 1
   ```
2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open domainValues_Xact_Source_AP_Invoice_psft.csv file in a text editor.

3. Copy the FIELDVALUE to the SOURCE_CODE column in the file. The data must be copied starting from the 10th line.
4. Map each FIELDVALUE to one AP Invoice domain value. Use commas to separate the entries.
5. There are two domain values for each source. The first domain value specifies if the source is a CREDIT_CARD application. Values are CREDIT_CARD or OTHER. The second domain value specifies whether the application is for MANUAL or ELECTRONIC entry. Values are MANUAL or ELECTRONIC.
6. Save and close the file.

4.2.3.1.7 How to Configure file_xact_type_AgreementLeverageType_psft.csv This section explains how to configure the file_xact_type_AgreementLeverageType_psft.csv file.

1. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the file_xact_type_AgreementLeverageType_psft.csv in a text editor.
2. Configure the value of W_XACT_TYPE_CODE1.

   The following table lists the seven business scenarios for XACT_TYPE_CODE and XACT_TYPE_DESC. Scenarios one through six use a type of negotiation. Scenario seven does not use negotiation. Only Oracle EBS supports all seven scenarios.

   > **Note:** PeopleSoft does not support these scenarios: 5. Complex work and 6. Other Negotiated.

XACT_TYPE_CODE	XACT_TYPE_DESC
1. Purchase releases	Orders or releases against local blanket agreement, global blanket agreement, or planned orders.
2. Punch-out and catalog purchase	Purchases through the catalog or through a "punch-out" from the catalog to an external supplier site.
3. Contract reference	Orders against a contract agreement.
4. Sourcing document reference	Orders against a sourcing document.
5. Complex work	Complex work procurement.
6. Other negotiated	Other negotiated purchases as indicated by the buyer.
7. Non-negotiated	Non-agreement purchases without existing matching blanket purchase agreement.

3. In the preconfigured application, scenarios one through six are considered agreement purchases, for example W_XACT_TYPE_CODE1=Agreement. If your business rule does not consider one of these scenarios as an agreement purchase, change the value to "Non-Agreement."

4.2.3.1.8 How to Configure the domainValues_Xact_Types_Expense_psft.csv This section describes how to configure the domainValues_Xact_Types_Expense_psft.csv file.

1. Identify the different expense types of employee expense reports in your PeopleSoft source system by using the following SQL:

   ```
   SELECT DISTINCT EXPENSE_TYPE FROM PS_EX_TYPES_TBL
   ```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open domainValues_Xact_Types_Expense_psft.csv in a text editor.
3. Copy the EXPENSE_TYPE to XACT_TYPE_CODE column. The data must be copied starting from the second line.
4. Map each expense type to one of the respective EXPENSE_TYPE domain values. Use commas to separate the entries.
5. Save and close the file.

4.2.3.1.9 How to Configure the domainValues_Xact_Types_ExpensePay_psft.csv This section describes how to configure the domainValues_Xact_Types_ExpensePay_psft.csv file.

1. Identify the different expense pay types of employee expense reports in your PeopleSoft source system by using the following SQL:

```
SELECT A.FIELDVALUE FIELDVALUE, A.XLATLONGNAME XLATLONGNAME
FROM PSXLATITEM A LEFT OUTER JOIN PSXLATITEMLANG B ON A.FIELDNAME=B.FIELDNAME
AND A.FIELDVALUE=B.FIELDVALUE AND B.LANGUAGE_CD='ENG'
WHERE A.EFFDT=(SELECT MAX(C.EFFDT) FROM PSXLATITEM C
WHERE C.FIELDNAME=A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE )
AND A.FIELDNAME='EX_PYMNT_TYPE'
```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the domainValues_Xact_Types_ExpensePay_psft.csv in a text editor.
3. Copy the FIELDVALUE and XLATLONGNAME to the XACT_TYPE_CODE and XACT_TYPE_DESC columns in the file respectively. The data must be copied starting from the second line.
4. Map each expense pay type to one of the respective EXPENSE_PAY_TYPE domain values. Use commas to separate the entries.
5. Save and close the file.

4.2.3.1.10 How to Configure the domainValues_Status_Expense_psft.csv This section describes how to configure the domainValues_Status_Expense_psft.csv file.

1. Identify the different expense statuses of employee expense reports in your PeopleSoft source system by using the following SQL:

```
SELECT A.FIELDVALUE FIELDVALUE, A.XLATSHORTNAME XLATSHORTNAME,
A.XLATLONGNAME XLATLONGNAME FROM PSXLATITEM A LEFT OUTER JOIN PSXLATITEMLANG B
ON A.FIELDNAME=B.FIELDNAME AND A.FIELDVALUE=B.FIELDVALUE
AND B.LANGUAGE_CD='ENG'
WHERE A.EFFDT = (SELECT MAX(C.EFFDT) FROM PSXLATITEM C
  WHERE C.FIELDNAME = A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE )
    AND A.FIELDNAME='SHEET_STATUS'
```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the file domainValues_Status_Expense_psft.csv in a text editor.
3. Copy the FIELDVALUE to STATUS_CODE, XLATESHORTNAME to STATUS_NAME, and XLATLONGNAME to STATUS_DESC. The data must be copied starting from the second line.
4. Map each expense status to one of the respective EXPENSE REPORT STATUS domain values. Use commas to separate the entries.

5. Save and close the file.

4.2.3.1.11 How to Configure the domainValues_Payment_Method_psft.csv This section describes how to configure the domainValues_Payment_Method_psft.csv file.

1. Identify the different expense payment methods of employee expense in your PeopleSoft source system by using the following SQL:

   ```
   SELECT DISTINCT EXPEND_MTHD, DESCR FROM PS_EX_EXP_MTHD_TBL
   ```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the file domainValues_Payment_Method_psft.csv in a text editor.
3. Copy the EXPEND_MTHD and DESCR to the PAYMENT_METHOD_CODE and PAYMENT_METHOD_DESC columns in the file respectively. The data must be copied starting from the second line.
4. Map each expense payment method to one of the respective expense payment method domain values. Use commas to separate the entries.
5. Save and close the file.

4.2.3.2 CSV Worksheet Source Files for PeopleSoft Procurement and Spend Analytics

Table 4–3 lists the CSV worksheet source files for PeopleSoft Procurement and Spend Analytics in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).

Table 4–3 CSV Worksheet Source Files for PeopleSoft Procurement and Spend Analytics

Worksheet File Name	Description	Session
File_StatusDimension_PurchaseRequisition_psft.csv	Lists the Purchasing Requisition Status column and the corresponding domain values for the PeopleSoft application. This file contains both source values and domain values for the cycle status of the Purchase Requisition. The values of UNAPPROVED, UNPROCESSED, UNFULFILLED, FULFILLED, and NOT APPLICABLE have been defined. No configuration should be needed for this file.	SDE_PSFT_StatusDimension_PurchaseRequisition
File_TransactionTypeDimension_PO_CreationMethod_psft.csv	Lists the Purchase Order Creation Method transaction type column and the corresponding domain values for the PeopleSoft application. This file contains both source values and domain values for the purchase order creation method. The values of BATCH and ONLINE for one domain value have been defined. For the other domain value the values of AUTO and MANUAL have been defined. No configuration should be needed for this file.	SDE_PSFT_TransactionTypeDimension_PO_Creation_Method
file_purchasing_transaction_source_psft.csv	Lists the Purchase Order Transaction Source column and the corresponding domain values for the PeopleSoft application. This file contains both source values and domain values for the purchase order receipt source. The value of PO with a domain value of PO for RECEIVING type has been defined. No configuration should be needed for this file.	SDE_PSFT_TransactionSourceDimension_Purchasing

Table 4–3 (Cont.) CSV Worksheet Source Files for PeopleSoft Procurement and Spend Analytics

Worksheet File Name	Description	Session
file_ap_invoice_config_spend_voucher_psft.csv	Lists the AP Invoice document types and the corresponding designation on whether to include in spend calculations for the PeopleSoft application. For information about how to edit this file, see Section 4.2.3.2.1, "How to Configure file_ap_invoice_config_spend_voucher_psft.csv."	SDE_PSFT_APInvoiceLineFact
file_ap_invoice_spendtypes_psft.csv	Lists the AP Invoice distribution types and the corresponding domain values for the spend types for the PeopleSoft application. This file contains both source values and domain values for the AP Invoice distribution spend types. The values of PO MATCHED, PAYABLES LEAKAGE, and PO NOT REQUIRED have been defined for the distribution types of ITEM, TAX, FREIGHT, and MISCELLANEOUS. No configuration should be needed for this file.	SDE_PSFT_TransactionTypeDimension_APInvoiceSpendType
PROCUREMENT_PO_CONSIGNED_CODE_PSFT.csv	Lists the Purchase Order Consigned Code values for the PeopleSoft application. This file defines the purchase order consigned code source value for the transaction type and the code dimensions. The values of Y - Consigned and N-Regular have been defined. No configuration should be needed for this file.	SDE_PSFT_TransactionTypeDimension_Consigned_Code, SDE_PSFT_CodeDimension_Supply_Chain_Purchasing_Consigned_Code
file_xact_type_AgreementLeverageType_psft.csv	Lists the agreement leverage types and the corresponding domain values for the PeopleSoft application. For more information about how to edit this file, see Section 4.2.3.2.2, "How to Configure file_xact_type_AgreementLeverageType_psft.csv."	SDE_PSFT_TransactionTypeDimension_AgreementLeverageType

4.2.3.2.1 How to Configure file_ap_invoice_config_spend_voucher_psft.csv This section explains how to configure the file_ap_invoice_config_spend_voucher_psft.csv file for the AP Invoice spend calculation.

This file defines AP Invoice types and has a flag to indicate whether the type should be included in spend calculations.

1. From the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles), open file_ap_invoice_config_spend_voucher_psft.csv file in a text editor.
2. Update the Y/N flag to designate whether to include the type in spend calculations.
3. Save and close the file.

4.2.3.2.2 How to Configure file_xact_type_AgreementLeverageType_psft.csv This section describes how to configure the file_xact_type_AgreementLeverageType_psft.csv file.

1. From the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles), open the file_xact_type_AgreementLeverageType_psft.csv file in a text editor.
2. Configure the value of W_XACT_TYPE_CODE1.

   The following table lists the seven business scenarios for XACT_TYPE_CODE and XACT_TYPE_DESC. Scenarios one through six use a type of negotiation. Scenario seven does not use negotiation. Only Oracle EBS supports all seven scenarios.

Note: PeopleSoft does not support these scenarios: 5. Complex work and 6. Other negotiated.

XACT_TYPE_CODE	XACT_TYPE_DESC
1. Purchase releases	Orders or releases against local blanket agreement, global blanket agreement or planned orders.
2. Punch-out and catalog purchase	Purchases through the catalog or through a "punch-out" from the catalog to an external supplier site.
3. Contract reference	Orders against a contract agreement.
4. Sourcing document reference	Orders against a sourcing document.
5. Complex work	Complex work procurement.
6. Other negotiated	Other negotiated purchases as indicated by the buyer.
7. Non-negotiated	Non-agreement purchases without existing matching blanket purchase agreement.

3. In the preconfigured application, scenarios one through six are considered agreement purchases, for example W_XACT_TYPE_CODE1=Agreement. If your business rule does not consider one of these scenarios as an agreement purchase, change the value to "Non-Agreement."

4.2.3.3 Enabling Reporting on Product/Purchasing Categories for PeopleSoft Procurement and Spend Analytics

Procurement and Spend Analytics supports reporting by Product/Purchasing Category hierarchy for organizations that use Product/Purchasing Category to create requisition and purchase orders.

In PeopleSoft Enterprise applications, Product/Purchasing Category hierarchies are stored in trees. A PeopleSoft Enterprise application user chooses a tree and then assigns a category from that tree to an item using the Create Requisition option from the eProcurement menu.

The default value of TREE_NAME in the PeopleSoft source system is ALL_PURCHASE_ITEMS. If your organization has not changed the value of TREE_NAME to another value, and you want to report by this TREE_NAME, then no additional configuration is required to report on Product/Purchasing Category hierarchies.

If your organization has changed the default value of TREE_NAME, then you need to perform the following tasks described in this section.

Use the following SQL query to extract the tree names from the PeopleSoft Enterprise application to confirm the value of TREE_NAME:

```
SELECT * FROM PSTREEDEFN WHERE TREE_STRCT_ID = 'ITEMS'
```

To enable business intelligence reporting on Product/Purchasing Categories, use DAC to configure the following tasks:

- SDE_PSFT_ProductDimension_ItemCategory
- SDE_PSFT_ProductDimension_ItemMaster

In both of these tasks, set the value of the $$TREE_NAME1 variable to the tree name that you want to use for reporting, and then save your changes.

4.3 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle Procurement and Spend Analytics.

- Section 4.3.1, "Configuration Steps for Oracle Procurement and Spend Analytics for All Source Systems"
- Section 4.3.2, "Configuration Steps for Oracle Procurement and Spend Analytics for Oracle EBS"
- Section 4.3.3, "Configuration Steps for Oracle Procurement and Spend Analytics for PeopleSoft"
- Section 4.3.4, "Configuration Steps for Oracle Procurement and Spend Analytics for Universal"

4.3.1 Configuration Steps for Oracle Procurement and Spend Analytics for All Source Systems

This section contains configuration steps that apply to all source systems. It contains the following topics:

- Section 4.3.1.1, "About Configuring the Purchase Receipts Aggregate Table"
- Section 4.3.1.2, "How to Configure the Purchase Receipts Aggregate Table"
- Section 4.3.1.3, "About Configuring the Purchase Cycle Lines Aggregate Table"
- Section 4.3.1.4, "How to Configure the Purchase Cycle Lines Aggregate Table"

4.3.1.1 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 ETL run, you need to configure the GRAIN parameter for the time aggregation level in the Purchase Receipts Aggregate fact table.

For the incremental ETL 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. Valid values are DAY, WEEK, MONTH, QUARTER, YEAR.

The Purchase Receipt Lines aggregate table is fully loaded from the base table in the initial ETL 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 ETL run. Oracle Business Analytics Warehouse minimizes the incremental aggregation effort, by modifying the aggregate table incrementally as the base table is updated.

When the Supply Chain - Purchase Receipts Subject Area is included in an Execution Plan in DAC, the Purchase Receipts data is extracted using the following tasks:

- **SIL_PurchaseReceiptAggregate_Derive_PreSoftDeleteImage** finds the records to be deleted in the base table since the last ETL run, and loads them into the W_PURCH_RCPT_TMP table. The measures in these records are multiplied by (-1). The task is run in the source-specific workflow before the records are deleted from the base table.
- **SIL_PurchaseReceiptAggregate_Derive_PreLoadImage** finds the records to be updated in the base table since the last ETL run, and loads them into the W_PURCH_RCPT_TMP table. The measures in these records are multiplied by (-1). The task is run in the source-specific workflow before the records are updated in the base table.
- **PLP_PurchaseReceiptAggregate_Derive_PostLoadImage** finds the inserted or updated records in the base table since the last ETL run, and loads them into the W_PURCH_RCPT_TMP table, without changing their sign. The task is run in the post load-processing workflow after the records are updated or inserted into the base table.
- **PLP_PurchaseReceiptAggregate_Load** 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.

4.3.1.2 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. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab.
3. For each of the following tasks, display the Parameters subtab and create the parameter name and parameter value pairs as follows:
   - SIL_PurchaseReceiptAggregate_Derive_PreLoadImage $$TIME_GRAIN MONTH
   - SIL_PurchaseReceiptAggregate_Derive_PreSoftDeleteImage $$TIME_GRAIN MONTH
   - PLP_PurchaseReceiptAggregate_Derive_PostLoadImage $$TIME_GRAIN MONTH
   - PLP_PurchaseReceiptAggregate_Load $$GRAIN 'MONTH'

To configure the Purchase Receipts aggregate table for Universal Source:

1. Using a text editor, open the file parameterfileDW.txt, located in the Dac\Informatica\parameters\input folder on the DAC Server computer.

   For example:

   C:\orahome\10g*version*\bifoundation\dac\Informatica\parameters\input
2. Locate the parameter entries for the PLP_PurchaseReceiptAggregate* and SIL_PurchaseReceiptAggregate* tasks.
3. Replace the default parameter values with your new values.
4. Save and close the file.

4.3.1.3 About Configuring the Purchase Cycle Lines Aggregate Table

The Purchase Cycle Lines aggregate table (W_PURCH_CYCLE_LINE_A) is used to capture information about the purchasing cycle lines.

For your initial ETL run, you need to configure the GRAIN parameter for the time aggregation level in the Purchase Cycle Lines Aggregate fact table.

For the incremental ETL 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. Valid values are DAY, WEEK, MONTH, QUARTER, YEAR.

The Purchase Cycle Lines aggregate table is fully loaded from the base table in the initial ETL 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 ETL run. Oracle Business Analytics Warehouse minimize the incremental aggregation effort, by modifying the aggregate table incrementally as the base table gets updated.

When the Supply Chain - Purchase Cycle Lines Subject Area is included in an Execution Plan in DAC, the Purchase Cycle Lines data is extracted using the following tasks:

- **SIL_PurchaseCycleLinesAggregate_Derive_PreSoftDeleteImage** finds the records to be deleted in the base table since the last ETL run, and loads them into the W_PURCH_CYCLE_LINE_TMP table. The task is run in the source-specific window before the records are deleted from the base table.
- **SIL_PurchaseCycleLinesAggregate_Derive_PreLoadImage** finds the records to be updated in the base table since the last ETL run, and loads them into the W_PURCH_CYCLE_LINE_TMP table. The measures in these records are multiplied by (-1). The task is run in the source-specific workflow before the records are updated in the base table.
- **PLP_PurchaseCycleLinesAggregate_Derive_PostLoadImage** finds the inserted or updated records in the base table since the last ETL run, and loads them into the W_PURCH_CYCLE_LINE_TMP table, without changing their sign. The task is run in the post load-processing workflow after the records are updated or inserted into the base table.
- **PLP_PurchaseCycleLinesAggregate_Load** 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.

4.3.1.4 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. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.

2. Display the Tasks tab.
3. For each of the following tasks, display the Parameters subtab and create the parameter name and parameter value pairs as follows:
   - SIL_PurchaseCycleLinesAggregate_Derive_PreLoadImage $$TIME_GRAIN MONTH
   - SIL_PurchaseCycleLinesAggregate_Derive_PreSoftDeleteImage $$TIME_GRAIN MONTH
   - PLP_PurchaseCycleLinesAggregate_Derive_PostLoadImage $$TIME_GRAIN MONTH
   - PLP_PurchaseCycleLinesAggregate_Load $$GRAIN 'MONTH'

4.3.2 Configuration Steps for Oracle Procurement and Spend Analytics for Oracle EBS

This section contains configuration steps that apply to Oracle EBS. It contains the following topics:

- Section 4.3.2.1, "How to Extract Particular Purchase Order Records"
- Section 4.3.2.2, "How to Enable Project Analytics Integration with Procurement and Spend Subject Areas"

4.3.2.1 How to Extract Particular Purchase Order Records

This configuration also applies to the Spend functional area.

You may not want to extract particular types of records from purchase orders in your source system. In these cases, you can modify the filter condition in the Source Qualifier of the mapplet. By default, the filter condition is set to `PLANNED`, `BLANKET`, or `STANDARD`. However, you can change this value to some conditional statement that only allows particular types of records to be extracted.

To extract particular types of purchase order records:

1. In Informatica PowerCenter Designer, open the SDE_ORA*Version*_Adaptor.
2. Open the mapplet mplt_BC_ORA_PurchaseOrderFact.
3. Double-click the Source Qualifier to display the Edit Transformations dialog, and display the Properties tab.
4. Select 'Sql Query' in the **Transformation Attribute** column, and click the down arrow in the **Value** column to display the SQL Editor dialog.
5. In the **SQL** box, replace the prepackaged filter condition with the new filter statement that reflects your business needs.
6. Edit the `WHERE` clause of the statement.
7. Click Apply to save the changes, and click OK to exit.
8. Validate the expression, and save your changes to the repository.
9. Repeat Step 2 to Step 8 for the mapplet mplt_BC_ORA_PurchaseScheduleLinesFact.

4.3.2.2 How to Enable Project Analytics Integration with Procurement and Spend Subject Areas

You can enable Oracle Procurement and Spend Analytics for EBS to use dimension tables in Oracle Project Analytics. You can only perform this integration if you have licensed Oracle Project Analytics.

To enable the integration:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Subject Areas tab.
3. For each Subject Area that you want to deploy, do the following:
   a. Select the Subject Area.
   b. Display the Configuration Tags tab in the upper pane.
   c. Select the Enable Project Dimensions tag.
   d. Clear the **Inactive** check box.
4. Assemble the Subject Areas.
5. Click the Execute button and build the Execution Plan for the Subject Areas that you updated.

4.3.3 Configuration Steps for Oracle Procurement and Spend Analytics for PeopleSoft

This section contains configuration steps that apply to PeopleSoft. It contains the following topic:

- Section 4.3.3.1, "How to Enable Project Analytics Integration with Procurement and Spend Subject Areas"

4.3.3.1 How to Enable Project Analytics Integration with Procurement and Spend Subject Areas

You can enable Oracle Procurement and Spend Analytics for PeopleSoft to use dimensions in Oracle Project Analytics. You can only perform this configuration if you have licensed Oracle Project Analytics.

To enable the integration:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Subject Areas tab.
3. For each Subject Area that you want to deploy, do the following:
   a. Select the Subject Area.
   b. Display the Configuration Tags tab in the upper pane.
   c. Select the Enable Project Dimensions tag.
   d. Clear the **Inactive** check box.
4. Assemble the Subject Areas.
5. Click the Execute button and build the Execution Plan for the Subject Areas that you updated.

4.3.4 Configuration Steps for Oracle Procurement and Spend Analytics for Universal

This section contains configuration steps that apply to Universal. It contains the following topics:

- Section 4.3.4.1, "About Configuring Oracle Procurement and Spend Analytics for Universal Source"
- Section 4.3.4.2, "How to Configure the Preferred Merchant Flag"
- Section 4.3.4.3, "How to Configure the Customer Billable Indicator"
- Section 4.3.4.4, "How to Configure the Receipts Indicator"
- Section 4.3.4.5, "How to Configure the Default Expense Distribution Percentage"
- Section 4.3.4.6, "How to Configure Lookup Dates for Currency Conversion"

4.3.4.1 About Configuring Oracle Procurement and Spend Analytics for Universal Source

Expenses has one fact table (W_EXPENSE_F) that supports metrics and reports for examining employee expenses. Several mappings populate these tables to complete extracts, loads and updates; you may configure these to suit your organization's business rules. The following sections discuss decisions you must make before you begin adapting individual PowerCenter objects, and provide specific configuration procedures for the universal source.

Universal source adapter mapplets extract data from a flat file interface to populate the Oracle Business Analytics Warehouse. In this phase of your project, you can configure the following:

- **System Flags and Indicators**. You may configure various system flags to indicate record rejection settings, as well as to indicate if your employees are using your preferred vendors, if you can forward expenses to your customers, and if receipts are available for expensed items.
- **Currency and Payment Options.** You may configure the date used to establish your exchange rates, determine if you allow expenses to be distributed across multiple cost centers, and define payment types in your data warehouse.

Before you begin, you must make the following decisions:

- **Cash Advances.** Cash advance records have a unique expense item number. If your system allows multiple cash advance records for one expense report, each of these advances must have their own identifiers.
- **Violations.** Many organizations capture violations of company expense policies at the item level (for example, the line item airfare exceeds $2000), cash advance level (for example, cash advance exceeds $500) and at the expense report level (for example, the report's total expenses exceed $5000). Currently the Oracle Business Analytics Warehouse stores item level violations within the corresponding item record, but the cash advance record stores both cash advance and report-level violations. Furthermore, each record has a VIOLATION_WID that can point to W_REASON_D, where violation details are stored. Depending on how you want your analytic system to perform, you must edit your universal business adapter file to reflect the violation counts and keys appropriately. For example:
  - If a requestor violates a cash advance policy, but there are no other violations at the report level, the VIOLATION_ID refers to the cash advance violation only. The violation count equals the cash advance violation counts.

- If a requestor violates company policy with their expense report, but has not taken a cash advance, you must add a dummy record in the flat file for a cash advance and set the cash advance amount to zero, and enter the violation count as the total number of expense report violations. In this scenario, VIOLATION_ID refers to the expense report violation data only.
- If a requestor violates a cash advance policy and an expense report policy, you must total the violation counts and enter them in your flat file record, and the VIOLATION_ID has no value. However, if your organization wants to prioritize the violations and have the VIOLATION_ID point to that which is most important, you may point it to the appropriate entry in W_REASON_D.

4.3.4.2 How to Configure the Preferred Merchant Flag

The Oracle Business Analytics Warehouse provides a preferred merchant flag to indicate whether the requestor used a preferred merchant for an expensed item. The flag can have only one value—`Y` (item acquired from a preferred merchant) or `N` (item acquired from a merchant not recorded). If you use custom logic to determine merchant status, you must include that logic in the Expenses source adapter.

To configure the preferred merchant flag:

1. In Informatica PowerCenter Designer, open the SDE_Universal_Adaptor folder.
2. Open the SDE_Universal_ExpenseFact mapping in the Mapping Designer.
3. Select the Expression transformation to open the Edit Transformations dialog and display the Ports tab.
4. If a port named PREF_MERCHANT_FLG is not present, click the Add Port icon and add a port named PREF_MERCHANT_FLG.
5. Configure the PREF_MERCHANT_FLG port logic as required.
6. Validate the mapplet and click OK to exit.
7. Save your changes to the repository.

4.3.4.3 How to Configure the Customer Billable Indicator

The Oracle Business Analytics Warehouse provides a customer billable indicator that registers whether an expense item is billed to a customer or paid by your organization. The flag can have only one value—`Y` (cost is passed to the customer) or `N` (cost is paid by your organization). If you use custom logic to determine customer billable status, you must include that logic in the expenses source adapter.

To configure the customer billable indicator:

1. In Informatica PowerCenter Designer, open the SDE_Universal_Adaptor folder.
2. Open the SDE_Universal_ExpenseFact mapping in the Mapping Designer.
3. Select the Expression transformation to open the Edit Transformations dialog, and display the Ports tab.
4. If a port named CUST_BILLABLE_FLG is not present, click the Add Port icon. Then, add a port called CUST_BILLABLE_FLG = *your_expression*.
5. Validate the mapplet and click OK to exit.
6. Save your changes to the repository.

4.3.4.4 How to Configure the Receipts Indicator

The Oracle Business Analytics Warehouse provides a receipts indicator that registers whether requestors have submitted a receipt for a line item in their expense report. The flag can have only one value—Y (receipts are available) or N (receipts are not available). If you use custom logic to indicate receipt availability, you must include that logic in the expenses source adapter.

To configure the receipts indicator:

1. In Informatica PowerCenter Designer, open the SDE_Universal_Adaptor folder.
2. Open the SDE_Universal_ExpenseFact mapping in the Mapping Designer.
3. Select the Expression transformation to open the Edit Transformations dialog, and display the Ports tab.
4. If a port named RECEIPT_FLG is not present, click the Add Port icon and add a port named RECEIPT_FLG.

   Add a port called RECEIPT_FLG_OUT and select only the flag "O". Leave the "I" and "V" flags deselected. Insert your expression for the receipt flag. Your expression can use the input port RECEIPT_FLG or other input columns based on your business logic.

   Re-map the RECEIPT_FLG port to the target and validate the mapping.
5. Save your changes to the repository.

4.3.4.5 How to Configure the Default Expense Distribution Percentage

At times, employee expenses may be distributed across multiple cost centers. For example, technical support associates frequently travel to work in an office with many cost centers; their expenses could be split between those who used their services. This cost center distribution is expected as a percentage from the source system or file; if it is not present a null value is returned. However, this prevents further calculations, so it is preferable to configure the default to be 100% if only one cost center is charged, rather than allow the system to return a null value.

To configure the default expense distribution percentage:

1. In Informatica PowerCenter Designer, open the SDE_Universal_Adaptor folder.
2. Open the SDE_Universal_ExpenseFact mapping in the Mapping Designer.
3. Select the Expression transformation to open the Edit Transformations dialog, and display the Ports tab.
4. If a port named DIST_PERCENTAGE is not present, click the Add Port icon. Then, add a port named DIST_PERCENTAGE = *expression_that_sets_this_to_100%*.
5. Validate your mapplet.
6. Save your changes.

4.3.4.6 How to Configure Lookup Dates for Currency Conversion

The Oracle Business Analytics Warehouse supports conversion of currency to document (transactional, or source, currency) and group (corporate umbrella currency) for exchange rates. The Oracle Business Analytics Warehouse uses a specific lookup date to determine the rate of exchange on the date an expense was incurred (ACTUAL_EXP_DT). If you decide to use a different date as your currency conversion lookup, you must use the following procedure.

To configure the exchange rate lookup date:

1. In Informatica PowerCenter Designer, open the SDE_Universal_Adaptor folder.
2. Open the SDE_Universal_ExpenseFact mapping in the Mapping Designer.
3. Select the Expression transformation to open the Edit Transformations dialog, and display the Ports tab.
4. If a port named EXT_XRATE_LOOKUP_DATE is not present, click the Add Port icon and add a port named EXT_XRATE_LOOKUP_DATE.
5. Edit the lookup date logic by substituting your lookup date for the prepackaged expression.
6. Validate the mapplet and click OK to exit.
7. Save your changes.

5 Configuring Oracle Financial Analytics

This section describes how to configure Oracle Financial Analytics. It contains the following topics:

- Section 5.1, "Overview of Oracle Financial Analytics"
- Section 5.2, "Configuration Required Before a Full Load for Financial Analytics"
- Section 5.3, "Configuration Steps for Controlling Your Data Set"

5.1 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 enables 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.

5.2 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. It contains the following topics:

- Section 5.2.1, "Configuration Steps for Financial Analytics for All Source Systems"
- Section 5.2.2, "Configuration Steps for Financial Analytics for Oracle EBS"
- Section 5.2.3, "Configuration Steps for Financial Analytics for PeopleSoft"
- Section 5.2.4, "Configuration Steps for Financial Analytics for Oracle's JD Edwards EnterpriseOne and JD Edwards World"

5.2.1 Configuration Steps for Financial Analytics for All Source Systems

This section contains configuration steps before you do a full data load that apply to all source systems.

Note: For configuration steps that apply to all Oracle BI Applications modules see Chapter 3, "Configuring Common Areas and Dimensions."

Note: If you want to deploy multi-calendar with Oracle Financial Analytics, you must enable a number of initialization blocks that are disabled by default. For more information, see Section 3.1.4.1.1, "About Enabling Initialization Blocks Required for Calendars."

5.2.2 Configuration Steps for Financial Analytics for Oracle EBS

This section contains configuration steps before you do a full data load that apply to Oracle EBS. It contains the following topics:

- Section 5.2.2.1, "About Configuring Domain Values and CSV Worksheet Files for Oracle Financial Analytics"

- Section 5.2.2.2, "How to Configure Transaction Types for Oracle General Ledger and Profitability Analytics (for Oracle EBS R12)"
- Section 5.2.2.3, "How to Configure Transaction Types for Oracle General Ledger and Profitability Analytics (for Oracle EBS R11i)"
- Section 5.2.2.4, "How to Specify the Ledger or Set of Books for which GL Data is Extracted"
- Section 5.2.2.5, "Oracle EBS Currency Balance Types"

5.2.2.1 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 from 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. You also need to determine which row should have 'Return' and which row should have 'Delivery' in the W_XACT_TYPE_CODE column.

Table 5–1 lists the CSV worksheet files and the domain values for Financial Analytics and Oracle EBS in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

Table 5–1 Domain Values and CSV Worksheet Files for Oracle Financial Analytics and Oracle EBS

Worksheet File Name	Description	Session
domainValues_Xact_Types_DocTypes_ora12.csv	List the Transaction types and the corresponding domain values for the Oracle EBS R12 application. For more information about updating the values in this file, see Section 5.2.2.2, "How to Configure Transaction Types for Oracle General Ledger and Profitability Analytics (for Oracle EBS R12)."	SDE_ORA_TransactionTypeDimension_GLCOGSDerive
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 5.2.2.3, "How to Configure Transaction Types for Oracle General Ledger and Profitability Analytics (for Oracle EBS R11i)."	SDE_ORA_TransactionTypeDimension_GLCOGSDerive

5.2.2.2 How to Configure Transaction Types for Oracle General Ledger and Profitability Analytics (for Oracle EBS R12)

This section explains how to configure Transaction Types for Oracle General Ledger and Profitability Analytics, Release 12, using the configuration file domainValues_Xact_Types_DocTypes_ora12.csv.

To configure Transaction types for Oracle General Ledger and Profitability Analytics:

1. Identify the entry types in the 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 the Oracle Inventory application and their corresponding descriptions.

2. Open the domainValues_Xact_Types_DocTypes_ora12.csv file using a text editor in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   Look for only the rows in the file which has XACT_CODE = 'COGS' or XACT_CODE = 'DEFERRED_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 the 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 the 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.

5. In Informatica PowerCenter Designer, open the mapping SDE_ORA_GLCOGSFact in the folder SDE_ORA*Version*_Adaptor.

6. Open the mapplet mplt_BC_ORA_GLCOGSFact contained inside this mapping

7. Open the Source Qualifier `SQ_MTL_TRANSACTION_ACCOUNTS`.

8. Modify the SQL so that it extracts the additional transaction type IDs, from step 1, that need to be extracted.

9. In Informatica PowerCenter Workflow Manager, make the same change to the SQL Override in the full load session SDE_ORA_GLCOGSFact_Full.

5.2.2.3 How to Configure Transaction Types for Oracle General Ledger and Profitability Analytics (for Oracle EBS R11i)

This section explains how to configure Transaction Types for Oracle General Ledger and Profitability Analytics, Release 11i, using the configuration file domainValues_Xact_Types_DocTypes_ora11i.csv.

To configure Transaction types for Oracle General Ledger and Profitability Analytics:

1. Identify the entry types in the 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 the Oracle Inventory application and their corresponding descriptions. The Oracle Inventory application is a data source from which Oracle General Ledger and Profitability Analytics extract data.

2. Using a text editor, open the domainValues_Xact_Types_DocTypes_ora11i.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   Look for only the rows in the file which has XACT_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 the 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.

5.2.2.4 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 BI 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.

You can use either of these configuration points separately and combine them. When installed out-of-the-box, Oracle BI Applications extract all GL data for all ledgers or set of books.

For Oracle 11i customers, to configure the list of sets of books for which to extract the GL data, do the following:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab.
3. Query for the parameter $$FILTER_BY_SET_OF_BOOKS_ID.
4. Double-click the Value column to open the text box, and enter 'Y'.

   Make sure to put single quotes around the Y.
5. Save the changes.
6. Query for the parameter $$SET_OF_BOOKS_ID_LIST.
7. Double-click the Value column to open the text box, and enter the IDs of the sets of books for which you want to extract GL data.

   Specify the list of sets of book IDs separated by commas (do not enter single or double quotes).

   For example, if you want to extract GL data for sets of books with IDs: 101, 207, and 303, then set the value of this parameter to the following:

   ```
   101, 207, 303
   ```
8. Save the changes.

You can also specify the Set of Books type for which you want to extract GL data. In Oracle 11i, there are three set of books types (this is based on the column GL_SETS_OF_BOOKS.mrc_sob_type_code):

- 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. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab.
3. Query for the parameter $$FILTER_BY_SET_OF_BOOKS_TYPE.
4. Double-click the Value column to open the text box, and enter the value 'Y'.

   Make sure to put single quotes around the Y.
5. Save the changes.
6. Query for the parameter $$SET_OF_BOOKS_TYPE_LIST.
7. Double-click the Value column to open the text box, and enter the types of set of books for which you want to extract GL data.

   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'
   ```
8. Save the changes.

For Oracle EBS R12 customers, to configure the list of ledgers to extract the GL data for, do the following:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab.
3. Query for the parameter $$FILTER_BY_LEDGER_ID.
4. Double-click the Value column to open the text box, and enter the value 'Y'.

   Make sure to put single quotes around the Y.
5. Save the changes.
6. Query for the parameter $$LEDGER_ID_LIST.
7. Double-click the Value column to open the text box, and enter the IDs of ledgers for which you want to extract GL data for.

   Specify the list of ledger IDs separated by commas (do not use single quotes).

   For example, if you want to extract GL data for ledgers with IDs: 101, 207, and 303, then set the value of this parameter as follows:

   ```
   101, 207, 303
   ```
8. Save the changes.

You can also specify types of ledgers you want to extract GL data for. In Oracle EBS R12, there are four types of ledgers (this is based on the column GL_LEDGERS.ledger_category_code):

- PRIMARY
- SECONDARY
- ALC
- NONE

For R12 customers, to configure the types of ledgers to extract the GL data for, do the following:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab, and query for the parameter $$FILTER_BY_LEDGER_TYPE.
3. Double-click the Value column to open the text box, and enter the value 'Y'.

   Make sure to put single quotes around the Y.
4. Save the changes.
5. Query for the parameter $$LEDGER_TYPE_LIST.
6. Double-click the Value column to open the text box, and enter the types of ledgers 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 Primary ledgers and all Secondary ledgers, then simply set the value of this parameter to as follows:

   ```
   'PRIMARY', 'SECONDARY'
   ```
7. Save the changes.

5.2.2.5 Oracle EBS Currency Balance Types

The ETL process extracts four different currency balance types from Oracle EBS, namely base currency, translated currency, entered currency, and statistical balance. These records are loaded and stored in W_GL_BALANCE_F table in the Oracle Data Warehouse.

The following table shows sample records of the different balance types. All four records are from ledger 'US Ledger' with U.S. dollars (USD) as the base ledger currency. These five columns in the following table are columns in the physical data warehouse table:

- LOC_CURR_CODE
- ACCT_CURR_CODE
- TRANSLATED_FLAG
- BALANCE_LOC_AMT
- BALANCE_ACCT_AMT

Line	Ledger Name	LOC_ CURR_ CODE	ACCT_ CURR_ CODE	TRANSLAT ED_FLAG	BALANCE_ LOC_AMT	BALANCE_ ACCT_ AMT	Balance Type Description
1	US Ledger	USD	NULL	N	200	0	A Base Currency Balance record with base ledger currency of USD
2	US Ledger	USD	GBP	Y	0	100	A Translated Currency Balance record with translation currency of GBP (British pounds). USD is the base ledger currency.
3	US Ledger	USD	JPY	N	0	10000	An Entered Currency Balance record with entered currency in JPY (USD is the base ledger currency)
4	US Ledger	USD	STAT	N	0	150	A Statistical balance amount

Users can query these balances by using different value columns and filters. The following tables contain sample queries that you can use to view different currency balance types and their corresponding results from the sample warehouse data in the preceding table.

Note: Ledger 'US Ledger' corresponds to LEDGER_WID = 85228 in the following sample scenarios.

Sample	Desired Queries	Physical Table Value Column	Physical Query Filters	Result for Value Column
A	To query: balance amount for base currency of USD, in ledger 'US Ledger'	BALANCE_LOC_ AMT	LEDGER_WID = 85228	200
B	To query: translated balance amount for translated currency of GBP, in ledger 'US Ledger'	BALANCE_ACCT_ AMT	LEDGER_WID 85228 AND TRANSLATED_ FLAG = 'Y' AND ACCT_CURR_ CODE = 'GBP'	100
C	To query: entered balance amount for entered currency of JPY, in ledger 'US Ledger'	BALANCE_ACCT_ AMT	LEDGER_WID = 85228 AND ACCT_CURR_ CODE = 'JPY' AND TRANSLATED_ FLAG = 'N'	10000
D	To query: statistical balance amounts, in ledger 'US Ledger'	BALANCE_ACCT_ AMT	LEDGER_WID = 85228 AND ACCT_CURR_ CODE = 'STAT'	150

Oracle BI Answers users can create reports on currency balance types by selecting the desired facts from the Financials - GL Balance Sheet presentation table, as shown in the following table.

Sample Scenario	Desired Report	Presentation Table	Presentation Column	Result in Report

A	View the local currency balance amounts (Debit Local Amount), such as in sample query A in the preceding table	Ledger
		Facts – GL Balance
			Ledger name
			Debit Local Amount
				Apply filter on:
				`"Ledger Name" = ’US Ledger’`
				`Ledger Name = US Ledger`
				`Debit Local Amount = 200`

B	View the translated currency balance amounts (Debit Translated Amount), such as in sample query B in the preceding table	Ledger
		Facts – GL Balance
		GL Balance Details
			Ledger Name
			Debit Translated Amount
			Translated Currency Code
				Apply filters on:
				`"Ledger Name" = ’US Ledger’`
				`"Translated Currency Code" = ’GBP’`
				`Ledger Name = US Leger`
				`Debit Translated Amount = 100`
				`Translated Currency Code = GBP`

C	View the entered currency balance amounts (Debit Entered Amount), such as in sample query C in the preceding table	Ledger
		Facts – GL Balance
		GL Balance Details
			Ledger Name
			Debit Entered Amount
			Entered Currency Code
				Apply filters on:
				`"Ledger Name" = ’US Ledger’`
				`"Entered Currency Code" = ’JPY’`
				`Ledger Name = US Leger`
				`Debit Entered Amount = 10000`
				`Entered Currency Code = JPY`

D	View the statistical amount (Statistical Amount), such as in sample query D in the preceding table	Ledger
		Facts - Balance Sheet Statement
			Ledger Name
			Statistical Amount
				`Ledger Name = US Leger`
				`Statistical Amount = 150`

5.2.3 Configuration Steps for Financial Analytics for PeopleSoft

This section contains configuration steps before you do a full data load that apply to PeopleSoft. It contains the following topics:

- Section 5.2.3.1, "About Configuring Domain Values and CSV Worksheet Files for Oracle Financial Analytics"
- Section 5.2.3.2, "How to Configure Domain Values for Oracle Receivables Analytics"
- Section 5.2.3.3, "How to insert beginning GL Balances into Oracle BI Applications"
- Section 5.2.3.4, "Oracle GL Budget Analytics for PeopleSoft Configuration"
- Section 5.2.3.5, "PeopleSoft Currency Balance Types"

5.2.3.1 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 section Section 5.2.3.2, "How to Configure Domain Values for Oracle Receivables Analytics" 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.

Table 5–2 lists the CSV worksheet files and the domain values for Financial Analytics and PeopleSoft in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

Table 5–2 Domain Values and CSV Worksheet Files for Oracle Financial Analytics

Worksheet File Name	Description	Session
domainValues_Xact_Types_DocTypes_psft.csv	List the Transaction types and the corresponding domain values for the PeopleSoft application. For more information about updating the values in this file, see Section 5.2.3.2, "How to Configure Domain Values for Oracle Receivables Analytics."	SDE_PSFT_TransactionTypeDimension_ARDerive

5.2.3.2 How to Configure Domain Values for Oracle Receivables Analytics

This section explains how to configure domain values for Oracle Receivables Profitability Analytics.

To configure domain values for Oracle Receivables Analytics:

1. Identify the entry types in your Receivables application by using the following SQL:

   ```
   SELECT DISTINCT ENTRY_TYPE, DESCR FROM PS_ENTRY_TYPE_TBL
   ```

   This query gives the transaction type codes in your Oracle Inventory application and their corresponding descriptions. The Oracle Inventory application is a data source from which Oracle General Ledger and Profitability Analytics extract data.

2. Using a text editor, open the domainValues_ Xact_Types_DocTypes_psft.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   Look for only the rows in the file that have XACT_CODE = 'RECEIVABLE' and W_XACT_TYPE_CODE = 'ORIGINAL' or 'APPLICATION'. The column XACT_SUBTYPE_CODE will contain the entry type codes from PeopleSoft Receivables and the columns W_XACT_TYPE_CODE and W_XACT_SUBTYPE_CODE are the corresponding domain values these entry type codes will be mapped to.

3. Copy the TYPE column to the XACT_TYPE_CODE column in the file.

   The data must be copied starting from the second line. The first line is the column header.

4. Map the entry type codes returned by the query above from your PeopleSoft Receivable 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 ORIGINAL/INVOICE domain value. Be careful not to map the same entry type to two different domain values.

5. Save and close the file.
6. In Informatica PowerCenter Designer, open mapplets mplt_BC_PSFT_ARTransactionFact_Item and mplt_BC_PSFT_ARTransactionFact_ItemActivity in the SDE_PSFT*Version*_Adaptor folder.
7. Open the Source Qualifier transformation.

   At the end of the SQL, there is a `IN` clause that should include all the entry type codes that map into W_XACT_TYPE_CODE = 'ORIGINAL' in the CSV file prepared above.
8. In Informatica PowerCenter Workflow Manager, make the same change to the full load sessions that correspond to the SDE_PSFT_ARTransactionFact_Item_Full and SDE_PSFT_ARTransactionFact_ItemActivity_Full mappings.

5.2.3.3 How to insert beginning GL Balances into Oracle BI Applications

Before you run the initial ETL, ensure that you set the source system parameters in DAC in order to compute the GL balances correctly.

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab.
3. Set the following parameters:
   - Set the parameter $$INITIAL_EXTRACT_DATE to the date from which you want to extract data from your PeopleSoft application to the data warehouse. Ensure that this Parameter is the beginning of an accounting period and not just any date in the middle of a period. For example, if you decide to extract data from June 2005, and the June 2005 accounting period starts from 5th June, set the date to 5th June, 2005.
   - Setup the parameter $$YEAR to the same as the fiscal year of $$INITIAL_EXTRACT_DATE parameter above. For example, if June 2005 is fiscal year 2005, set this parameter to 2005.
   - Set the parameter $$ACCOUNTING_PERIOD to the accounting period number of the $$INITIAL_EXTART_DATE parameter above. For example, if June 2005 is the 6th period of fiscal year 2005, set this parameter to 6.

5.2.3.4 Oracle GL Budget Analytics for PeopleSoft Configuration

PeopleSoft GL Budget Analytics provides data from the following two areas:

- Commitment Control Budget
- Standard General Ledger Budget

By default, the Data Warehouse is updated with Commitment Control Budget data. To add data from Standard General Ledger Budget to the data warehouse, you have to set up a configuration tag called Budget Standard in DAC. The tag should provide two options-an option for running only Commit Control Budget and an option for running both Commit Control Budget and Standard General Ledger Budget. DAC uses the Commitment Control Budget in the tag by default.

5.2.3.4.1 PeopleSoft Tree for Commitment Control Budget You can set up Commitment Control Budget based on Chartfield and PeopleSoft trees. For instructions about how to perform this configuration, which involves adding the Chartfield hierarchy to the data warehouse, see Section 5.3.3.3, "General Ledger Chartfield and Chartfield Hierarchy Extraction Customization."

5.2.3.5 PeopleSoft Currency Balance Types

The ETL process extracts four different currency balance types from PeopleSoft, namely base currency, translated currency, entered currency and statistical balance. These records are loaded and stored in the warehouse table W_GL_BALANCE_F. The following table shows some sample records of different balance types. The first four records are from Primary ledger 'Euro Ledger' with EUR as the base ledger currency. The next three records are from Translation ledger 'US Ledger' with USD as the base ledger currency. These five columns in the following table are columns in the physical data warehouse table:

- LOC_CURR_CODE
- ACCT_CURR_CODE
- TRANSLATED_FLAG
- BALANCE_LOC_AMT
- BALANCE_ACCT_AMT

Line	Ledger Name	LOC_ CURR_ CODE	ACCT_ CURR_ CODE	TRANSLATED_FLAG	BALANCE_ LOC_AMT	BALANCE_ ACCT_ AMT	Balance Type Description
1	Euro Ledger (Primary ledger)	EUR	EUR	N	390	50	A Base Currency Balance record with base ledger currency of EUR
2	Euro Ledger (Primary ledger)	EUR	USD	N	0	200	An Entered Currency Balance record with entered currency in USD (EUR is the base ledger currency)
3	Euro Ledger (Primary ledger)	EUR	STAT	N	0	80	A Statistical balance amount
4	US Ledger (Translation ledger)	USD	USD	Y	340	340	A Translated Currency Balance record with translation currency of USD **Note**: US Ledger is a Translation ledger
5	US Ledger (Translation ledger)	USD	EUR	N	0	50	An Entered Currency Balance record with entered currency in EUR (USD is the base ledger currency) **Note**: US Ledger is a Translation ledger

Users can query these balances by using different value columns and filters. The following table contains sample queries to show how to view different currency balance types and their corresponding results from the sample warehouse data in the preceding table.

> **Note:** Primary ledger 'Euro Ledger' corresponds to LEDGER_WID = 85228 and Translation ledger 'US Ledger' corresponds to LEDGER_WID = 87320 in the following sample scenarios.

Sample	Desired Queries	Physical Table Value Column	Physical Query Filters	Result for Value Column
A	To query: balance amount for base currency of EUR, in Primary ledger ’Euro Ledger’	BALANCE_LOC_AMT	LEDGER_WID = 85228	390
B	To query: entered balance amount for entered currency of EUR in Primary ledger ’Euro Ledger'	BALANCE_ACCT_AMT	LEDGER_WID = 85228 AND ACCT_CURR_CODE = 'EUR' AND TRANSLATED_FLAG = 'N'	50
C	To query: entered balance amount for entered currency of USD, in Primary ledger ’Euro Ledger'	BALANCE_ACCT_AMT	LEDGER_WID = 85228 AND ACCT_CURR_CODE = 'USD' AND TRANSLATED_FLAG = 'N'	200
D	To query: statistical balance amount, in Primary ledger ’Euro Ledger’	BALANCE_ACCT_AMT	LEDGER_WID = 85228 AND ACCT_CURR_CODE = 'STAT'	80
E	To query: translated balance account amount for translated currency of USD, in Translation ledger ’US Ledger’	BALANCE_ACCT_AMT	LEDGER_WID = 87320 AND TRANSLATED_FLAG = 'Y'	340
F	To query: entered balance amount for entered currency of EUR, in Translation ledger ’US Ledger’	BALANCE_ACCT_AMT	LEDGER_WID = 87320 AND ACCT_CURR_CODE = 'EUR' AND TRANSLATED_FLAG = 'N'	50

Oracle BI Answers users can create reports on currency balance types by selecting the desired facts from the Financials – GL Balance Sheet presentation table, as shown in the following table.

Sample Scenario	Desired Report	Presentation Table	Presentation Column	Result in Report

A	View the local currency balance amounts (Debit Local Amount), such as in sample query A in the preceding table	Ledger
		Facts – GL Balance
			Ledger Name
			Debit Local Amount
				Apply filter on:
				`"Ledger Name" = 'Euro Ledger'`
				`Ledger Name = Euro Ledger`
				`Debit Local Amount = 390`

B	View the local currency balance amounts (Debit Amount), such as in sample query B in the preceding table	Ledger
		Facts – GL Balance
			Ledger Name
			Debit Amount
				Apply filter on:
				`"Ledger Name" = ’Euro Ledger’`
				`Ledger Name = Euro Ledger`
				`Debit Amount = 50`

C	View the entered currency balance amounts (Debit Entered Amount), such as in sample query C in the preceding table	Ledger
		Facts – GL Balance
		GL Balance Details
			Ledger Name
			Debit Entered Amount
			Entered Currency Code
				Apply filters on:
				`"Ledger Name" = ’Euro Ledger’`
				`Entered Currency Code" = ’USD’`
				`Ledger Name = Euro Ledger`
				`Debit Entered Amount = 200`
				`Entered Currency Code = USD`

D	View the statistical amount (Statistical Amount), such as in sample query D in the preceding table	Ledger
		Facts – Balance Sheet Statement
			Ledger Name
			Statistical Amount
				`Ledger Name = Euro Ledger`
				`Statistical Amount = 80`

E	View the translated currency balance amounts (Debit Translated Amount), such as in sample query E in the preceding table	Ledger
		Facts – GL Balance
		GL Balance Details
			Ledger Name
			Debit Translated Amount
			Translated Currency Code
				Apply filters on:
				`"Ledger Name" = 'US Ledger'`
				`"Translated Currency Code" = 'USD'`
				`Ledger Name = US Ledger`
				`Debit Translated Amount = 340`
				`Translated Currency Code = USD`

F	If we want to view the Entered currency balance amounts (Debit Entered Amount), such as in sample query F in the preceding table	Ledger
		Facts – GL Balance
		GL Balance Details
			Ledger Name
			Debit Entered Amount
			Entered Currency Code
				Apply filters on:
				`"Ledger Name" = 'US Ledger'`
				`"Entered Currency Code" = 'EUR'`
				`Ledger Name = US Ledger`
				`Debit Entered Amount = 50`
				`Entered Currency Code = EUR`

5.2.4 Configuration Steps for Financial Analytics for Oracle's JD Edwards EnterpriseOne and JD Edwards World

This section contains configuration steps before you do a full data load that apply to Oracle's JD Edwards EnterpriseOne and JD Edwards World. It contains the following topics:

- Section 5.2.4.1, "About Configuring Domain Values and CSV Worksheet Files for Oracle Financial Analytics for Oracle's JD Edwards EnterpriseOne and JD Edwards World"
- Section 5.2.4.2, "How to Configure domainvalues_payment_method_jde.csv"
- Section 5.2.4.3, "How to Configure domainvalues_src_transaction_type_codes_jde.csv and domainvalues_lkp_transaction_type_codes_jde.csv"
- Section 5.2.4.4, "How to Configure domainvalues_make_buy_indicator_product_jde.csv"
- Section 5.2.4.5, "About the file_fin_stmt_item_names_jde.csv and file_fin_sub_names_jde.csv Files"
- Section 5.2.4.6, "How to Configure the file_group_acct_codes_jde.csv"
- Section 5.2.4.7, "How to Configure the file_glacct_catcodes_segment_config_jde.csv"
- Section 5.2.4.8, "How to Configure the file_glacct_segment_config_jde.csv"
- Section 5.2.4.9, "How to Configure the file_lkp_fiscal_period_Qtr_Config_jde.csv"
- Section 5.2.4.10, "Configuring the Exchange Date for JD Edwards EnterpriseOne Accounts Payable"
- Section 5.2.4.11, "About the Exchange Date for JD Edwards EnterpriseOne Account Receivables"

5.2.4.1 About Configuring Domain Values and CSV Worksheet Files for Oracle Financial Analytics for Oracle's JD Edwards EnterpriseOne and JD Edwards World

Table 5–3 lists the CSV worksheet files and the domain values for Financial Analytics for JD Edwards EnterpriseOne and JD Edwards World.

Table 5–3 Domain Values and CSV Worksheet Files for Oracle Financial Analytics for JD Edwards EnterpriseOne and JD Edwards World

Worksheet File Name	Description	Session
domain_values_payment_method_jde.csv	Lists the Payment Method and the corresponding domain values for the JD Edwards EnterpriseOne or JD Edwards World application.	SDE_JDE_Payment_Method_Dimension
domainvalues_src_Transaction_Type_Codes_jde.csv domainvalues_lkp_Transaction_Type_Codes_jde.csv	Lists the Transaction Types and the corresponding domain values for the JD Edwards EnterpriseOne or JD Edwards World application.	SDE_JDE_Transaction_Type_Dimension
domainvalues_make_buy_indicator_product_jde.csv	Lists the Make Buy Indicator Product and the corresponding domain values for the JD Edwards EnterpriseOne or JD Edwards World application.	SDE_JDE_ProductDimensions

Table 5–3 (Cont.) Domain Values and CSV Worksheet Files for Oracle Financial Analytics for JD Edwards EnterpriseOne and JD Edwards World

Worksheet File Name	Description	Session
file_fin_stmt_item_names_jde.csv file_fin_sub_names_jde.csv	Lists the Financial Statement Item Name and the corresponding domain values for the JD Edwards EnterpriseOne or JD Edwards World application.	SDE_JDE_CodeDimension_Fin_Stmt_Item_Names and SDE_JDE_CodeDimension_Fin_Sub_Names
file_group_acct_codes_jde.csv	Lists the Group Account Codes and the corresponding domain values for the JD Edwards EnterpriseOne or JD Edwards World application.	SDE_JDE_Stage_GroupAccountNumberDimension
file_grpact_fstmt.csv	Lists the Group Account financial Statement and the corresponding domain values for the JD Edwards EnterpriseOne or the JD Edwards World application. This file works in conjunction with the group_account_codes flatfile.	SIL_Stage_GroupAcctNumber_FinStatementItem
file_glacct_catcodes_segment_config_jde.csv	Lists the GL Account Category Codes Segment and the corresponding domain values for the JD Edwards EnterpriseOne or JD Edwards World application.	SDE_JDE_Stage_GLAccountDimension_CatCodesFlatFile
file_glacct_segment_config_jde.csv	Lists the GL Account Segment and the corresponding domain values for the JD Edwards EnterpriseOne or JD Edwards World application.	SDE_JDE_Stage_GLAccount_SegmentConfig_Extract
file_lkp_fiscal_period_Qtr_Config_jde.csv	Lists the Time Dimension Fiscal Period and the corresponding domain values for the JD Edwards EnterpriseOne or JD Edwards World application.	SDE_JDE_Timedimension_FiscalPeriod_Extract

5.2.4.2 How to Configure domainvalues_payment_method_jde.csv

This section explains how to configure the domainvalues_payment_method_jde.csv. You populate this lookup flat file with data before running the mapping to load W_PAYMENT_METHOD_D.

To configure domainvalues_payment_method_jde.csv:

1. Identify the Payment Methods in your JD Edwards EnterpriseOne or JD Edwards World source system by using the following SQL:

   ```
   Select
   DRKY
   from F0005
   WHERE
   DRSY ='00' AND DRRT ='PY'
   ```

2. Using a text editor, open the flat file domainvalues_payment_method_jde.csv located in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).
3. Copy the output of the query to the UDC column of the flat file.
4. Map each of the values under the UDC column to one of the Payment Methods defined for the Oracle Business Analytics Warehouse and enter it under column W_PAYMENT_METHOD_D.

   For more information about the W_PAYMENT_METHOD_D domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

5. Use commas to separate the entries under the UDC column and W_PAYMENT_METHOD_D. Ensure that you do not include any spaces.
6. Save and close the file.

5.2.4.3 How to Configure domainvalues_src_transaction_type_codes_jde.csv and domainvalues_lkp_transaction_type_codes_jde.csv

This section explains how to configure the domainvalues_src_transaction_type_codes_jde.csv and domainvalues_lkp_transaction_type_codes_jde.csv files. These flat files are used as source flat files in SDE_JDE_Transaction_Type_Dimension to retrieve the value of W_XACT_TYPE_CODE, W_XACT_SUBTYPE_CODE, and RECEIPT_TYPE by matching the UDC value of the lookup table with the DRKY value from the source table.

If a UDC value from 00/DT or 98/IT does not map to one of the XACT_CODEs listed in the Transaction Types dimension class table for Financial Analytics, then do not include it in the domainvalues_src_transaction_type_codes_jde.csv file. For example, Document Type IA (Inventory Adjustments) does not map to a XACT_CODE domain value and therefore it should not be included in the CSV file.

Furthermore, review the UDC values that come pre-populated in the domainvalues_src_transaction_type_codes_jde.csv file and make any necessary changes to the pre-populated domain values in order to align with how these UDCs are used in your JD Edwards EnterpriseOne or JD Edwards World implementation. For example, if UDC = '%' does not correspond to any of the XACT_CODE domain values then remove the row from the domainvalues_src_transaction_type_codes_jde.csv file.

All UDCs that do correspond to a XACT_CODE domain values must be included in this CSV file and properly configured. If not, then metrics pertaining to the domain values (for example Payables and Receivables) are at risk for not calculating correctly.

For more information on XACT_CODE domain values, see Transaction Types Dimension Class Table for Financial Analytics which appears under the W_XACT_TYPE_D Domain Values in the *Oracle Business Analytics Warehouse Data Model Reference*.

To configure the CSV files:

1. Identify XACT_TYPE_CODE in your JD Edwards EnterpriseOne or JD Edwards World source system by using the following SQL:

   ```
   Select
   DRKY
   From F0005
   WHERE (DRSY ='00' AND DRRT ='DT') OR
   (DRSY ='98' AND DRRT ='IT')
   ```

2. Using a text editor, open the domainvalues_src_transaction_type_codes_jde.csv file located in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).
3. Using the output from the query, append any UDC values that are not in the CSV file to the end of the CSV file. This is necessary only if the UDC pertains to the XACT_CODE domain value.
4. Map each UDC value to one domain value in each of these domain columns: XACT_CODE, W_XACT_TYPE_CODE, and W_XACT_SUBTYPE_CODE.
5. Save and close the file.

6. Copy and paste the domainvalues_src_transaction_type_codes_jde.csv into the $PMServer\lkpfiles location and change the file name to domainvalues_lkp_transaction_type_codes_jde.csv.
7. Update the above files when necessary.

5.2.4.4 How to Configure domainvalues_make_buy_indicator_product_jde.csv

This section explains how to configure the domainvalues_make_buy_indicator_product_jde.csv file. Use this file to identify the Product Stocking Types in your JD Edwards EnterpriseOne or JD Edwards World source system. In W_PRODUCT_D, the MAKE_BUY_IND column represents the source system domain value.

To configure the domainvalues_make_buy_indicator_product_jde.csv:

1. Identify XACT_TYPE_CODE in your JD Edwards EnterpriseOne or JD Edwards World source system by using the following SQL:

   ```
   Select
   DRKY
   From F0005
   WHERE DRSY ='41' AND DRRT =' I'
   ```

2. Using a text editor, open the domainvalues_make_buy_indicator_product_jde.csv file located in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).
3. Copy the output of the query to the UDC column of the flat file.
4. Map each of the values under the Product Stocking Type column to the Make Buy Indicator column in the Oracle Business Analytics Warehouse.

   For more information about the W_PRODUCT_D domain values, see *Oracle Business Analytics Warehouse Data Model Reference.*

5. Use commas to separate the entries under the Product Stocking Type and Make Buy Indicator columns. Ensure that you do not include any spaces.
6. Save and close the file.

5.2.4.5 About the file_fin_stmt_item_names_jde.csv and file_fin_sub_names_jde.csv Files

The file_fin_stmt_item_names_jde.csv is used to identify the CODE-NAME combinations of Financial Statement Item Names. For more information about the published values, see *Oracle Business Analytics Warehouse Data Model Reference*. The file_fin_sub_names_jde.csv can be ignored.

The file_fin_stmt_item_names_jde.csv flat file is pre-populated; you do not have to configure it. However, you should review it to make sure the file is populated with values. The file_fin_sub_names_jde.csv flat file is also pre-populated. You can ignore this file, but do not delete it.

To review the files:

1. Using a text editor, open the file_fin_stmt_item_names_jde.csv located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   The target for this data is W_CODE_D and therefore follows the typical CODE dimension load format.

2. Save and close the file.

5.2.4.6 How to Configure the file_group_acct_codes_jde.csv

This section explains how to configure the file_group_acct_codes_jde.csv. This flat file is used to identify the Group Account Codes for each object account range for each company. For example, for company 00001 you might specify group account codes for accounts 1000 to 2000 as REVENUE.

For a detailed list of the domain values for all possible Group Account Codes, see *Oracle Business Analytics Warehouse Data Model Reference.*

To configure file_group_acct_codes_jde.csv:

1. Using a text editor, open the file_group_acct_codes_jde.csv located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
2. In the Company field, populate the company that you are setting up and specify the ranges in the From and To column of that row with the corresponding Group Account Code.
3. Save and close the file.

5.2.4.7 How to Configure the file_glacct_catcodes_segment_config_jde.csv

This section explains how to configure the file_glacct_catcodes_segment_config_jde.csv. This file is used to determine which Account Segment Code (1 to 30) takes which F0901 column value for each record. You associate a segment code (1 to 30) to these columns in the F0901 account master: CO, MCU, OBJ, SUB, SBL, SBLT, and any of 43 category codes based on your reporting needs.

Determine the columns that you need for reporting and populate the flat file accordingly. Do not change the values of the SEGMENT_NAME column. The list of values that the JDE_COLUMN can take include GMCO, GMMCU, SBL, SBLT, GMSUB, GMOBJ, and any value from GMR001 through GMR0043.

To configure file_glacct_catcodes_segment_config_jde.csv:

1. Using a text editor, open the file_glacct_catcodes_segment_config_jde.csv located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
2. Add values beside the comma using the above domain values for the configuration. Ensure that you do not include any spaces.
3. Save and close the file.

5.2.4.8 How to Configure the file_glacct_segment_config_jde.csv

This section explains how to configure the file_glacct_segment_config_jde.csv. This file is generic and determines which among the first six segment codes of account dimension is used for aggregation in the GL_BALANCE_F. The row with row_id = 1 is redundant; whether or not it is populated has no impact. However, the dummy row, available in the sample file, might be necessary. The role of this row with row_id = 1 is taken care of by the file_glacct_catcodes_segment_config_jde.csv file.

To configure file_glacct_segment_config_jde.csv:

1. Using a text editor, open the file_glacct_segment_config_jde.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

2. In the row with ROW_ID = AGGREGATION, assign a value of Y to any of the first six segment codes to include these columns in the aggregation. Leave the trailing commas as such. Ensure that you do not include any spaces.
3. Save and close the file.

5.2.4.9 How to Configure the file_lkp_fiscal_period_Qtr_Config_jde.csv

This section explains how to configure the file_lkp_fiscal_period_Qtr_Config_jde.csv. You must configure this file to support the metrics that are based on the Fiscal Quarter.

To configure file_lkp_fiscal_period_Qtr_Config_jde.csv:

1. Identify the Fiscal Quarter data in your JD Edwards EnterpriseOne or JD Edwards World source system by using the following SQL:

   ```
   Select
   CDDTPN,CDFY from F0008
   ```

2. Using a text editor, open the file_lkp_fiscal_period_Qtr_Config_jde.csv located in $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. For each Date pattern, set the following fields:

Field	Value
FiscalPattern	CDDTPN
YEAR	CDFY
Period	Period Number Numeric
Quarter No	As there is no concept of Quarter in JD Edwards EnterpriseOne and JD Edwards World, define a numeric quarter number to which the period belongs.
Quarter Start Date	Customized Quarter Start Date for each period. Each quarter can span as many periods as users configure. The format is DD/MM/YYYY.
Quarter End Date	Customized Quarter End Date for each period. Each quarter can span as many periods as users configure. The format is DD/MM/YYYY.

Note: Ensure that there are no unnecessary spaces in the flat file.

4. Save and close the file.

5.2.4.10 Configuring the Exchange Date for JD Edwards EnterpriseOne Accounts Payable

You can populate the field EXCHANGE_DT in Accounts Payable (AP) with either the invoice date or the general ledger date using the DAC parameter $$EXCHANGE_DATE. Based on your selection, the EXCHANGE_DT field in the W_AP_XACT_FS table is populated with the appropriate date in the Oracle Business Analytics Warehouse.

To configure the $$EXCHANGE_DT parameter:

1. Make sure to configure the invoice date or general ledger date the same as it is in JD Edwards EnterpriseOne so that your results are the same as the JD Edwards EnterpriseOne source system values.
2. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
3. Click the Tasks tab, and perform a query for the AP transaction fact-related tasks.
4. In the Detail area of the selected tasks, click the Parameters tab.
5. Modify the parameter value as follows:
   - Specify INV to use the invoice date.
   - Specify GL to use the general ledger date.
6. Save your changes.

5.2.4.11 About the Exchange Date for JD Edwards EnterpriseOne Account Receivables

The EXCHANGE_DT field in W_AR_XACT_FS is populated from the F03B11-RPERDJ table. The JD Edwards EnterpriseOne application stores the exchange date in the F03B11 table, which contains the value selected by the user through the processing option.

5.3 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle Financial Analytics. It contains the following topics:

- Section 5.3.1, "Configuration Steps for Financial Analytics for All Source Systems"
- Section 5.3.2, "Configuration Steps for Financial Analytics for Oracle EBS"
- Section 5.3.3, "Configuration Steps for Financial Analytics for PeopleSoft"
- Section 5.3.4, "Configuration Steps for Financial Analytics for Oracle’s JD Edwards EnterpriseOne and JD Edwards World"

5.3.1 Configuration Steps for Financial Analytics for All Source Systems

This section contains additional configuration steps that apply to all source systems. It contains the following topics:

- Section 5.3.1.1, "How to Include Unapplied Payments in AR Aging Calculations"
- Section 5.3.1.2, "How to Configure the AP/AR Aging Tables"
- Section 5.3.1.3, "How to Set Up Drill Down in Oracle BI Answers from General Ledger to Subledger"
- Section 5.3.1.4, "About the Customer Costs Lines and Product Costs Lines Tables for Oracle Profitability Analytics"
- Section 5.3.1.5, "How to Configure the Customer Costs Lines and Product Costs Lines Tables for Oracle's Profitability Analytics"
- Section 5.3.1.6, "How to Configure Budgets for Oracle Profitability Analytics"

5.3.1.1 How to Include Unapplied Payments in AR Aging Calculations

Follow these steps to include Unapplied Payments in Aging Calculations for AR Aging tables:

1. In Informatica Designer, open the mapping PLP_ARSnapshotInvoiceAging in the PLP Folder.
2. Edit the Source Qualifier transformation `SQ_IA_AR_XACTS`, and open the SQL Query in the Properties tab.
3. In the `WHERE` clause of the SQL statement:
   a. Change the SQL text from:

   ```
   W_XACT_TYPE_D.W_XACT_SUBTYPE_CODE IN ('INVOICE','CR MEMO','DR MEMO') AND
   ```

   To:

   ```
   W_XACT_TYPE_D.W_XACT_SUBTYPE_CODE IN ('INVOICE','CR MEMO','DR MEMO',
   'PAYMENT') AND
   ```

   b. Change the SQL text from:

   ```
   W_XACT_TYPE_D.W_XACT_TYPE_CODE = 'ORIGINAL' AND
   ```

   To:

   ```
   W_XACT_TYPE_D.W_XACT_TYPE_CODE IN ('ORIGINAL','PAYMENT') AND
   ```

4. Save the changes.

   The next ETL will populate the Aging tables using UnApplied payments in the calculations.

5.3.1.2 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

Use DAC to configure the lengths of these aging buckets, as described in the following procedure.

To configure the length of aging buckets:

1. In the Design view, display the Tasks tab.
2. Query for the following tasks:

- PLP_APSnapshotInvoiceAging
- PLP_ARSnapshotInvoiceAging

3. For each of these tasks, display the Parameters tab in the lower pane and update the parameters with names starting with '$$BUCKET'.

   You can also use the $$HISTORY_MONTHS parameter to specify how many historic month end snapshots you want to retain (the out-of-the-box value is 24 months).

   Note: You cannot create aging snapshots for months prior to your initial load ETL. Aging snapshots are created one month at a time, as you go forward with your regular ETL runs. For example, if you start your system in January 2009 (that is, run the initial ETL some time in January 2009), and you set $$HISTORY_MONTHS to 12, you will have one snapshot at the end of January 2009, two at the end of February 2009, three at the end of March 2009 and so on until the end of December 2009 when you will have 12 snapshots.

4. Save your changes.

5.3.1.3 How to Set Up Drill Down in Oracle BI Answers from General Ledger to Subledger

> **Note:** The Drill Down feature is not supported on Oracle JD Edwards source systems.

Oracle BI Applications enables you to trace a GL Journal to the subledger transaction that created that journal. This ability (or drill down) is achieved through the 'Navigation' feature in Oracle BI Answers.

To set up drill down in Oracle BI Answers from General Ledger to subledger:

1. Create your subledger 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.

   You may add multiple navigation targets if your GL report shows transactions from multiple subledgers and you want to drill from GL to the appropriate Subledger report. For example, if your GL report shows transactions from AP, AR and Revenue, and you have three subledger reports for each of these, you can add three navigation targets (by selecting the option 'Add Navigation Targets') and set the locations to each of these reports. Subsequently, when you run the GL report and when you click the "GL Journal ID" column Value, a popup appears, where you need to click the appropriate target based on the journal you selected. This will not happen automatically. For example, if you click a journal transaction originating from AP, you need to pick the appropriate subledger report (that is, the AP report in this case) to drill into the AP report and see the details. You can add the Group Account Number attribute from GL Account Dimension to your GL report to easily identify the subledger that the GL transaction belongs to.

Note: For COGS, the 'GL Journal ID' column is not exposed in any subject areas. It is available in the Business Model and Mapping layer of the Oracle BI repository under the logical tables 'Dim - GL COGS Details'. As a workaround, you can create subject areas to report of detail level transactions for COGS and expose this column under the 'Document Details' folder in the subject area and use similar steps as above to set up a drill-down from GL to COGS.

To create a subject area for COGS:

1. Using the Administration Tool, open OracleBIAnalyticsApps.rpd.

   The OracleBIAnalyticsApps.rpd file is located at:

   ```
   ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\
   coreapplication_obisn\repository
   ```

2. Create an empty subject area (for example, Financials – GL Cost of Goods Sold). Set properties by following other subject areas.
3. Drag 'Dim – GL COGS Details' and 'Fact - Fins - GL Cost of Goods Sold Posted' to the subject area.
4. Drag other dimensions.
5. Rename the presentation table 'Dim – GL COGS Details' to 'Document Details'.
6. Rename the presentation table 'Fact - Fins - GL Cost of Goods Sold Posted' to 'Facts - GL Cost of Goods Sold'. Rename other dimensions if necessary.

You might also follow this same process to create a presentation table for Revenue to be able to drill from GL to Revenue level detail transactions.

5.3.1.4 About the Customer Costs Lines and Product Costs Lines Tables for Oracle Profitability Analytics

This configuration is required only if you are implementing Oracle Profitability Analytics and you want to allocate your expenses by product or by customer dimensions. The out-of-the-box adapter does not capture the miscellaneous cost and expenses associated with generating revenue from a customer or from a product (for example, marketing campaign expenses). You must provide this miscellaneous data through the Universal Adapter, as described in this section.

The configuration files file_customer_cost_line.csv and file_product_cost_line.csv are used to configure the Customer Cost Lines table and the Product Cost Lines table before an ETL full load. These configuration files are located by default in *MW_HOME*\biapps\dwrep\Informatica\Sample Universal Source Files.

Note: If these configuration files have file has been previously configured, updated copies of the files are located in the *INFA_HOME*\server\infa_shared\SrcFiles folder on the Informatica Server computer.

- The file_customer_cost_line.csv file loads the Customer Costs Lines data in the W_CUSTOMER_COST_LINE_F table, and contains the following data:

  - Cost details by customers. The total cost by customers includes both direct cost that is captured in the financial system by products, and the allocated costs that are captured in the costing system.
  - The source of allocations. For example, the Sales and Marketing costs are not captured as direct costs by customers. However, at a later point in time, the costs could be allocated from a combination of General Ledger accounts or Cost Centers into various customers.
  - The source cost centers and General Ledger accounts. The Customer Costs can be categorized by Sales, Marketing, Service, and Operating costs. These could be further broken down to Salaries, Events, and Promotions. The Oracle Business Analytics Warehouse provides a set of categories, and these can be changed depending on the user needs and cost components by which customers are tracked.
  - The actual cost lines are also tracked in this table. In addition to Customers, there are number of other dimensions by which the costs can be tracked such as Sales Region, Sales Geography, Company, Business Area and the associated hierarchies. Not all dimension keys are relevant to all costs.

- The file_product_cost_line.csv file loads the Product Costs Lines data in the W_PRODUCT_COST_LINE_F table, and contains the following data:
  - Cost details by products. The total cost by products include both direct cost that is captured in the financial system by products, and the allocated costs that are captured in the costing system.
  - The source of allocations. For example, the Sales and Marketing costs are not captured as direct costs by products. However, at a later point in time, the costs are allocated from a combination of General Ledger accounts and Cost Centers into various products.
  - The source cost centers and General Ledger accounts. The product costs can be categorized by Sales, Marketing, Service, and Operating costs. These could be further broken down into Salaries, Events, and Promotions. Oracle Business Analytics Warehouse provides a set of common categories, and these can be changed depending on the user needs and cost components by which products are tracked.
  - The actual cost lines. In addition to Products, there are number of other dimensions by which the costs are tracked such as Sales Region, Sales Geography, Company, Business Area, and the associated hierarchies. Not all dimension keys are relevant to all costs.

5.3.1.5 How to Configure the Customer Costs Lines and Product Costs Lines Tables for Oracle's Profitability Analytics

Follow this procedure to configure the Customer Cost Lines and Product Costs Lines before you perform a full load ETL.

Note: The General Ledger Revenue and General Ledger COGS fact tables are populated by either the Universal source system, the Oracle 11i source system, the JD Edwards EnterpriseOne source system, or the JD Edwards World source system. The Customer Costs Lines and Product Costs Lines fact tables are populated by the universal source system.

To configure the Customer Costs Lines and Product Costs Lines tables:

1. Copy the configuration files file_customer_cost_line.csv and file_product_cost_line.csv from the *MW_HOME*\biapps\dwrep\Informatica\Sample Universal

Source Files folder to the *INFA_HOME*\server\infa_shared\SrcFiles folder on the Informatica Server computer.

Note: You can omit this step if the configuration files have already been copied to the *INFA_HOME*\server\infa_shared\SrcFiles folder on the Informatica Server computer.

2. Using a text editor, open the file_customer_cost_line.csv file in the *INFA_HOME*\server\infa_shared\SrcFiles folder.
3. Insert a record into the file for each customer costing transaction you want to load into the Customer Cost fact table.
4. Save the file.
5. Using a text editor, open the file_product_cost_line.csv file in the *INFA_HOME*\server\infa_shared\SrcFiles folder.
6. Insert a record into the file for each product costing transaction you want to load into the Product Cost fact table.
7. Save the file.

   You are now ready to perform a full load ETL to load the Customer Cost Lines and Product Costs Lines. To do this, use DAC to create and execute an Execution Plan that includes the Universal adapter Subject Areas 'Financials - Customer Expenses' and 'Financials - Product Expenses'.

5.3.1.6 How to Configure Budgets for Oracle Profitability Analytics

If you are using Oracle EBS, PeopleSoft, JD Edwards EnterpriseOne, or JD Edwards World source systems and would like to extract the budget data from these sources and import it into the data warehouse, you can use the preconfigured adapter mappings provided with Oracle Profitability Analytics. However, if you want to use budget data from other external systems, you can import the data into the data warehouse using the Universal adapter as described in this section. This section includes the following topics:

- Section 5.3.1.6.1, "About Configuring Universal Source Files"
- Section 5.3.1.6.2, "How to Import Budget Data into the Data Warehouse Through the Universal Adapter"

5.3.1.6.1 About Configuring Universal Source Files The following tables describe the columns in the universal source CSV files file_budget.csv and file_acct_budget.csv, their data types, and how to populate them where applicable.

Table 5–4 shows the structure of the file_budget.csv file. The records in file_budget.csv are loaded into W_BUDGET_D.

Table 5–4 Universal Source for Budget Fact (file_budget.csv)

Column Name	Datatype	Size	Description
BUDGET_NAME	string	80	Budget name.
BUDGET_VERSION	string	30	Budget version.
BUDGET_STATUS	string	30	Budget status.
BUDGET_TYPE	string	30	Budget type.
CREATED_BY_ID	string	80	ID of created by user. Populate with Integration_ID from w_user_d.

Table 5–4 (Cont.) Universal Source for Budget Fact (file_budget.csv)

Column Name	Datatype	Size	Description
CHANGED_BY_ID	string	80	ID of changed by user. Populate with Integration_ID from w_user_d.
CREATED_ON_DT	string	14	Created date.
CHANGED_ON_DT	string	14	Changed date. Used for updating an existing record in the warehouse. Increase the date if you want to update the record. If a record with the same integration_ID already exists in the target table W_BUDGET_D, then the load process will compare the CHANGED_ON_DT values between this record and the record in W_BUDGET_D. If this record's CHANGED_ON_DT is later than the record in W_BUDGET_D, then the load process will perform an update against the record in W_BUDGET_D; otherwise the load process will ignore this record, and no update or insertion will occur. If there is no matching record in W_BUDGET_D with the same integration_ID, then the load process will insert this record into W_BUDGET_D.
AUX1_CHANGED_ON_DT	string	14	-
AUX2_CHANGED_ON_DT	string	14	-
AUX3_CHANGED_ON_DT	string	14	-
AUX4_CHANGED_ON_DT	string	14	-
DELETE_FLG	string	1	-
DATASOURCE_NUM_ID	number	10	A number for your data source. Populate the same datasource_num_id as your main source application.
INTEGRATION_ID	string	80	A unique identifier for the record.
TENANT_ID	string	80	-
X_CUSTOM	string	10	-

Table 5–5 shows the structure of the file_acct_budget.csv file. The records in the file_acct_budget.csv will be loaded into W__ACCT_BUDGET_F.

Table 5–5 Universal Source for Budget Fact (file_acct_budget.csv)

Column Name	Datatype	Size	Description
ADJUSTMENT_FLG	string	1	TBD
AUX1_CHANGED_ON_DT	string	14	TBD
AUX2_CHANGED_ON_DT	string	14	TBD
AUX3_CHANGED_ON_DT	string	14	TBD

Table 5–5 (Cont.) Universal Source for Budget Fact (file_acct_budget.csv)

Column Name	Datatype	Size	Description
AUX4_CHANGED_ON_DT	string	14	TBD
BUDG_BUSN_AREA_ORG_ID	string	80	Company Org identifier. Populate with integration_id from w_int_org_d where business_area_flg = Y.
BUDG_CTRL_AREA_ORG_ID	string	80	Company Org identifier. Populate with integration_id from w_int_org_d where ctrl_area_flg = Y.
BUDG_FIN_AREA_ORG_ID	string	80	Company Org identifier. Populate with integration_id from w_int_org_d where fin_area_flg = Y.
BUDGET_CALENDAR_ID	string	80	TBD
BUDGET_DOC_AMT	number	22	Budget amount in document currency.
BUDGET_GRP_AMT	number	22	TBD
BUDGET_ID	string	80	Populate with the value from integration_id in file_budget.csv
BUDGET_LEDGER_ID	string	80	TBD
BUDGET_LOC_AMT	number	22	Budget amount in local currency.
CHANGED_BY_ID	string	80	ID of changed by user. Populate with Integration_ID from w_user_d.
CHANGED_ON_DT	string	14	Changed date. Used for updating an existing record in the warehouse. Increase the date if you want to update the record. If a record with the same integration_ID exists in the target table W_ACCT_BUDGET_F already, then the load process will compare the CHANGED_ON_DT values between this record and the record in W_ACCT_BUDGET_F. If this record's CHANGED_ON_DT is later than the record in W_ACCT_BUDGET_F, then the load process will perform an update against the record in W_ACCT_BUDGET_F; otherwise it'll ignore this record, no update or insertion will occur. If there's no matching record in W_ACCT_BUDGET_F with the same integration_ID, then the load process will insert this record into W_ACCT_BUDGET_F.
COMPANY_ORG_ID	string	80	Company Org identifier. Populate with integration_id from w_int_org_d where company_flg = Y.
COST_CENTER_ID	string	80	Cost Center identifier. Populate with integration_id from w_cost_center_d.
CREATED_BY_ID	string	80	ID of created by user. Populate with Integration_ID from w_user_d.
CREATED_ON_DT	string	14	Created date.

Table 5–5 (Cont.) Universal Source for Budget Fact (file_acct_budget.csv)

Column Name	Datatype	Size	Description
DATASOURCE_NUM_ID	number	10	A number for your data source. Populate the same datasource_num_id as your main source application.
DELETE_FLG	string	1	TBD
DOC_CURR_CODE	string	30	Document currency code.
GL_ACCOUNT_ID	string	80	GL Account identifier. Populate with integration_id from w_gl_account_d.
GRP_CURR_CODE	string	30	TBD
INTEGRATION_ID	string	80	A unique identifier for the record.
LOC_CURR_CODE	string	30	Local currency code.
PERIOD_BEGIN_DT	string	14	TBD
PERIOD_END_DT	string	14	Populate with your budget period's end date. If your budget is monthly, populate with the month end date.
POSTED_ON_DT	string	14	A date on which this transaction can be reported.
PRODUCT_ID	string	80	Product identifier. Populate with integration_id from w_product_d.
PROFIT_CENTER_ID	string	80	Profit Center identifier. Populate with integration_id from w_profit_center_d.
PROJECT_ID	string	80	-
TENANT_ID	string	80	TBD
X_CUSTOM	string	10	TBD

Note: Date columns should be populated in the CSV file as a number in the format YYYYMMDDHH24MISS.

Use Table 5–6 to understand how the integration_id (key) of some of the key dimensions are constructed for the Oracle EBS source system. You can use this information to populate the dimension foreign key identifiers in the above universal source CSV file for budget fact, if you have to use budget fact in conjunction with dimensions populated from Oracle EBS.

Table 5–6 Populating the integration_id fields in Oracle EBS Source Systems

Field	How to populate
GL_ACCOUNT_ID (w_gl_account_d)	ccid
COMPANY_ORG_ID (w_int_org_d)	No need to populate; will be calculated based on GL Account ID.
COST_CENTER_ID (w_cost_center_d)	No need to populate; will be calculated based on GL Account ID.
PROFIT_CENTER_ID (w_profit_center_d)	No need to populate; will be calculated based on GL Account ID.
LEDGER_ID (w_ledger_d)	For Oracle 11i, populate as set of book ID. For Oracle R12, populate as ledger ID.

Use Table 5–7 to understand how the integration_id (key) of some of the key dimensions are constructed for Oracle's JD Edwards EnterpriseOne and JD Edwards World source systems. You can use this information to populate the dimension foreign key identifiers in the above universal source CSV file for budget fact, if you have to use budget fact in conjunction with dimensions populated from Oracle's JD Edwards EnterpriseOne or JD Edwards World.

Table 5–7 Populating the integration_id fields in Oracle's JD Edwards EnterpriseOne and JD Edwards World Source Systems

Field	How to populate
GL_ACCOUNT_ID (w_gl_account_d_)	GBAID||'~'||GBSBL||'~'||GBSBLT
COMPANY_ORG_ID (w_int_org_d)	GBCO
COST_CENTER_ID (w_cost_center_d)	GBMCU
PROFIT_CENTER_ID (w_profit_center_d)	GBCO
LEDGER_ID (w_ledger_d)	GBCO
PRODUCT_ID (w_product_d)	If GBSBLT points to item, then update product ID with that GBSBL.
PROJECT_ID (w_product_d)	Not applicable
BUDG_BUSN_AREA_ORG_ID (w_int_org_d)	GBMCU
BUDG_FIN_AREA_ORG_ID (w_int_org_d)	GBMCU
BUDG_CTRL_AREA_ORG_ID (w_int_org_d)	GBMCU
BUDGET_ID (w_budget_d)	Not applicable

5.3.1.6.2 How to Import Budget Data into the Data Warehouse Through the Universal Adapter

Follow these steps to import budget data into the data warehouse through the Universal adapter.

1. Populate the file_budget.csv and file_acct_budget.csv files with your budget data.

   The CSV files are located in the $pmserver\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).

   Refer the tables above for details of how to populate these files.

2. In DAC, create a new Execution Plan using the Subject Area 'Financials – Budget' in the 'Universal' container.

3. Run the new Execution Plan that you created in the previous step.

   Note: This Execution Plan must be run after the regular Execution Plan to populate the data warehouse for the other Subject Areas has completed.

4. Load new the new budget data or changes to existing budget data.

   Repeat Step 1 and Step 3 as needed to load new budget for the next fiscal period or make corrections to already loaded budget data.

5.3.2 Configuration Steps for Financial Analytics for Oracle EBS

This section contains additional configuration steps that apply to Oracle EBS. It contains the following topics:

- Section 5.3.2.1, "How GL Balances Are Populated in Oracle EBS"

- Section 5.3.2.2, "How to Configure Oracle Profitability Analytics Transaction Extracts"
- Section 5.3.2.3, "How to Configure Cost Of Goods Extract (for Oracle EBS 11i)"
- Section 5.3.2.4, "How to Configure AP Balance ID for Oracle Payables Analytics"
- Section 5.3.2.5, "How to Configure AR Balance ID for Oracle Receivables Analytics and Oracle General Ledger and Profitability Analytics"
- Section 5.3.2.6, "How to Configure the AR Adjustments Extract for Oracle Receivables Analytics"
- Section 5.3.2.7, "How to Configure the AR Schedules Extract"
- Section 5.3.2.8, "How to Configure the AR Cash Receipt Application Extract for Oracle Receivables Analytics"
- Section 5.3.2.9, "How to Configure the AR Credit-Memo Application Extract for Oracle Receivables Analytics"
- Section 5.3.2.10, "How to Enable Project Analytics Integration with Financial Subject Areas"

5.3.2.1 How GL Balances Are Populated in Oracle EBS

Oracle BI Applications provides two ways to populate the GL balances (stored in the W_GL_BALANCE_F table), as follows:

- By extracting the GL balances directly from Oracle General Ledger, as follows:

  a. In DAC, go to the Design view, and then display the Subject Areas tab.

  b. Query for the Subject Area 'Financials – General Ledger'. Display the Configuration Tags subtab, and query for the tag 'Oracle – Extract GL Balance'. Activate the tag by deselecting the Inactive check box.

  c. Inactivate the tag 'Financials – Calculate GL Balance by selecting the is Inactive check box.

  d. Click Assemble to reassemble the Subject Area.

  e. Rebuild the Execution Plans that contain this Subject Area.

- By calculating the GL balances based on the records in the W_GL_OTHER_F table, which stores all journal lines, as follows:

  a. In DAC, go to the Design view, and then display the Subject Areas tab.

  b. Query for the Subject Area 'Financials – General Ledger'. Display the Configuration Tags subtab, and query for the tag 'Financials – Calculate GL Balance'. Activate the tag by deselecting the Inactive check box.

  c. Inactivate the tag 'Oracle – Extract GL Balance' by selecting the is Inactive check box.

  d. Click Assemble to reassemble the Subject Area.

  e. Rebuild the Execution Plans that contain this Subject Area.

5.3.2.2 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 Business Component.

To modify the extract filter for Oracle Profitability Analytics Revenue:

1. In Informatica PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA*Version*_Adaptor).
2. In Mapplet Designer, open the mplt_BC_ORA_GLRevenueFact mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations dialog, and display the Properties tab.

   In the User Defined Join field and in the SQL Query field, remove the statement:

   ```
   AND RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG = Y
   ```

4. Validate and save your changes to the repository.

5.3.2.3 How to Configure Cost Of Goods Extract (for Oracle EBS 11i)

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 Business Component mapplet. You need to modify the Business Component mapplet for both the regular extract mapplet as well as the primary extract mapplet. This does not apply to EBS R12; 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 Informatica PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA*Version*_Adaptor).
2. In Mapplet Designer, open mplt_BC_ORA_GLCOGSFact.
3. Double-click the Source Qualifier to open the Edit Transformations dialog, and display the Properties tab.

   In the User Defined Join field and in the SQL Query field, remove the statement:

   ```
   AND MTL_TRANSACTION_ACCOUNTS.GL_BATCH_ID <> -1
   ```

4. In Mapplet Designer, open mplt_BC_ORA_GLCOGSFact_Primary.
5. Double-click the Source Qualifier to open the Edit Transformations dialog, and display the Properties tab.

   In the User Defined Join field and in the SQL Query field, remove the statement:

   ```
   AND MTL_TRANSACTION_ACCOUNTS.GL_BATCH_ID <> -1
   ```

6. Validate and save your changes to the repository.

   Note: Steps 4 to 6 are optional. Although the mplt_BC_ORA_GLCOGSFact_Primary mapplet is not used in any of the out-of-the-box Subject Areas, Oracle recommends that you follow steps 4 to 6 in case the mplt_BC_ORA_GLCOGSFact_Primary mapplet is used in a custom Subject Area.

7. Make appropriate changes to the SQL in the Informatica session, 'SDE_ORA_GLCOGSFact_Full', which is present in the corresponding folder in Informatica PowerCenter Workflow Manager.
8. If you customized the mapping SDE_ORA_GLCOGSFact to extract from Oracle EBS versions 11.5.8, 11.5.9, or 11.5.10, modify the domain values file domainValues_Xact_Types_DocTypes_ora11i.csv.
9. To extract from Oracle EBS version R12, modify the file domainValues_Xact_Types_DocTypes_ora12.csv.

5.3.2.4 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||'~'||
ORG_ID||'~'||VENDOR_ID
```

However, if you want to maintain your AP balance at a different grain, you can redefine the Balance ID value in the applicable mapplets.

To modify the Accounts Payable Balance ID:

> **Note:** To modify the Accounts Payable Balance ID, you must modify the following mapplets:
>
> - mplt_SA_ORA_APTransactionFact_LiabilityDistribution
> - mplt_SA_ORA_APTransactionFact_ExpenseDistribution
> - mplt_SA_ORA_APTransactionFact_Payment

1. In Informatica PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA*Version*_Adaptor).
2. In Mapplet Designer, open the mapplet (for example, mplt_SA_ORA_APTransactionFact_LiabilityDistribution).
3. Double-click the Expression transformation to open the Edit Transformations dialog.
4. Display the Ports tab, and edit the expression for the EXT_BALANCE_ID port.

   For example, you might click the Expression field to display the Expression Editor and modify the text in the **Formula** box.
5. Validate and save your changes to the repository.
6. Repeat steps 1 to 5 for each mapplet that is listed above.

5.3.2.5 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 mapplets.

To modify the AR Balance ID:

Note: To modify the AR Balance ID, you must modify the following mapplets:

- mplt_SA_ORA_ARTransactionFact_Adjust
- mplt_SA_ORA_ARTransactionFact_ARScheduleDerive
- mplt_SA_ORA_ARTransactionFact_CreditMemoApplication
- mplt_SA_ORA_ARTransactionFact_ReceivableApplication

1. In Informatica PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA*Version*_Adaptor).
2. In Mapplet Designer, open the mapplet (for example, mplt_SA_ORA_ARTransactionFact_Adjust).
3. Double-click the Expression transformation to open the Edit Transformations dialog.
4. Display the Ports tab, and edit the expression for the EXT_NU_AR_BALANCE_ID port.

   For example, you might click the Expression field to display the Expression Editor and modify the text in the **Formula** box.
5. Validate and save your changes to the repository.
6. Repeat steps 1 to 5 for each mapplet that is listed above.

5.3.2.6 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 Business Component mapplet. By modifying or removing the filter, you can extract other entries, such as those that require more research, those that are rejected, and those that are not accrued charges.

To modify the extract filter for Accounts Receivable adjustments:

1. In Informatica PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA*Version*_Adaptor).
2. In Mapplet Designer, open the mplt_BC_ORA_ARTransactionFact_Adjust mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations dialog, and display the Properties tab.

   In the SQL Query field and in the User Defined Join field, modify the statement:

   ```
   AND AR_ADJUSTMENTS_ALL.STATUS = A
   ```
4. Validate and save your changes to the repository.

5. Repeat Step 2 to Step 4 for the mplt_BC_ORA_ARTransactionFact_AdjustPrimary mapplet.

5.3.2.7 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 Business Component mapplet. 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 Informatica PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA*Version*_Adaptor).
2. In Mapplet Designer, open the mplt_BC_ORA_ARTransactionFact_ARSchedules mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations dialog.

   In the User Defined Join field and in the SQL Query field, modify the statement:

   ```
   AND RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG(+) = Y
   ```

4. Validate and save your changes to the repository.

5.3.2.8 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 Business Component mapplet. By modifying or removing the filter, you can extract other entries, such as unconfirmed applications.

You must modify both the regular mapplet (mplt_BC_ORA_ARTransactionFact_ReceivableApplication) as well as the primary extract mapplet (mplt_BC_ORA_ARTransactionFact_ReceivableApplicationPrimary).

To modify the extract filter for AR cash receipt application:

1. In Informatica PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA*Version*_Adaptor).
2. In Mapplet Designer, open the mplt_BC_ORA_ARTransactionFact_ReceivableApplication mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations dialog, and display the Properties tab.

   In the User Defined Join field and in the SQL Query field, modify the statement:

   ```
   AND NVL(AR_RECEIVABLE_APPLICATIONS_ALL.CONFIRMED_FLAG,'Y') = Y
   ```

4. Validate and save your changes to the repository.
5. Repeat Step 2 to Step 4 for the mplt_BC_ORA_ARTransactionFact_ReceivableApplicationPrimary mapplet.

5.3.2.9 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 mapplet (mplt_BC_ORA_ARTransactionFact_CreditmemoApplication), as well as the primary extract mapplet (mplt_BC_ORA_ARTransactionFact_CreditmemoApplicationPrimary). Repeat the following procedure for each mapplet.

To modify the extract filter for Accounts Receivable Credit-Memo Application:

1. In Informatica PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA*Version*_Adaptor).
2. In Mapplet Designer, open the mplt_BC_ORA_ARTransactionFact_CreditmemoApplication mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations dialog, and display the Properties tab.

   In the User Defined Join field and in the SQL Query field, modify the statement:

   ```
   AND NVL(AR_RECEIVABLE_APPLICATIONS_ALL.CONFIRMED_FLAG,'Y') = Y
   ```

4. Validate and save your changes to the repository.
5. Repeat Step 2 to Step 4 for the mplt_BC_ORA_ARTransactionFact_CreditmemoApplicationPrimary mapplet.

5.3.2.10 How to Enable Project Analytics Integration with Financial Subject Areas

You can enable Oracle Financial Analytics for EBS to use dimension tables in Oracle Project Analytics. You can only perform this integration if you have licensed Oracle Project Analytics.

You can configure the following Oracle Financial Analytics for EBS Subject Areas to use Project Analytics for EBS tables:

- Financials -Payables
- Financials -Receivables
- Financials - Cost of Goods Sold
- Financials - Revenue

The following Oracle Financial Analytics fact tables integrate with Project Analytics dimensions:

- W_AP_INV_DIST_F
- W_AP_XACT_F
- W_AR_XACT_F
- W_GL_COGS_F
- W_GL_REVN_F
- W_AP_AGING_INVOICE_A

- W_AR_AGING_INVOICE_A

To enable the integration:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Select each Subject Area, and then in the Configuration Tags tab, clear the Inactive check box for the Enable Project Dimensions configuration tag.
3. Assemble the Subject Areas.
4. Click the Execute button and build the Execution Plan for the Subject Areas that you updated.

5.3.3 Configuration Steps for Financial Analytics for PeopleSoft

This section contains additional configuration steps that apply to PeopleSoft. It contains the following topics:

- Section 5.3.3.1, "Overview of PeopleSoft Tree Data Structures"
- Section 5.3.3.2, "About PeopleSoft Trees in Oracle Financial Analytics"
- Section 5.3.3.3, "General Ledger Chartfield and Chartfield Hierarchy Extraction Customization"
- Section 5.3.3.4, "How to Set Up Default Fiscal Calendars For PeopleSoft"
- Section 5.3.3.5, "How to Enable Project Analytics Integration with Financial Subject Areas"
- Section 5.3.3.6, "Configuring Number of Days based Metrics for PeopleSoft"
- Section 5.3.3.7, "How to Update Dashboard Pages with PeopleSoft Prompts"

5.3.3.1 Overview of PeopleSoft Tree Data Structures

For PeopleSoft, the Oracle Financial Analytics application uses data from a data structure called PeopleSoft trees to get information about the organization's GL account hierarchies, profit center hierarchies, cost center hierarchies, and so on.

PeopleSoft trees are a flexible, generic way of constructing hierarchical relationships among PeopleSoft entities, such as chartfields, business units, and so on.

5.3.3.2 About PeopleSoft Trees in Oracle Financial Analytics

Oracle BI Applications provides a set of mechanisms to support various tree structures in PeopleSoft applications. Table 5–8 shows the supported structure types, and Table 5–9 shows the supported level types.

Table 5–8 PeopleSoft Tree Structure Types

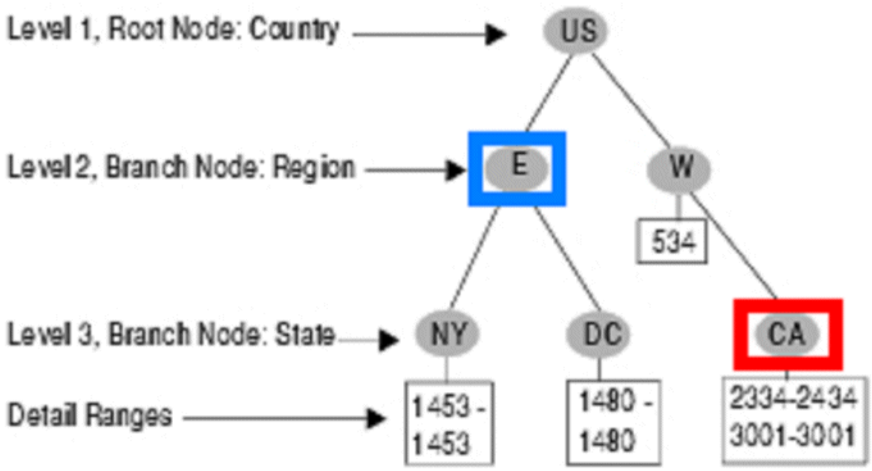
Structure Type	Support
Winter tree	Y
Detail tree	Y
Summer tree	Y

Table 5–9 PeopleSoft Tree Level Types

Level Type	Support
Strict-level tree	Y
Loose-level tree	Y

A flattened structure is used to store the hierarchy-related data. Each record of W_HIERARCHY_D and other hierarchy-related dimension tables (for example, W_INT_ORG_DH) represents a single node in the tree. It stores the path information from the top-most node to that node. For the bottom nodes in the detail tree, the extraction process generates a number of records according to the node's range and the record in the relevant table and puts those records into the hierarchy table. Figure 5–1 provides an illustration of this functionality.

Figure 5–1 Example GL Chartfield Hierarchy


Level 1, Root Node: Country
US
Level 2, Branch Node: Region
E
W
534
Level 3, Branch Node: State
NY
DC
CA
Detail Ranges
1453 - 1453
1480 - 1480
2334-2434
3001-3001

For any node in the tree, except for the bottom nodes, W_HIERARCHY_D will have one corresponding record. For example, for the 'E' node in the tree above (the one highlighted with a blue square), the record shown in Table 5–10 will appear in the database.

Table 5–10 How Node E Is Stored in W_HIERARCHY_D Table

HIER1_CODE	HIER2_CODE	HIER3_CODE	HIER4_CODE	HIER[5-20]_CODE
US	E	E	E	E

If the actual hierarchy level in the tree ends earlier than the predefined level of W_HIERARCHY_D, then the lower hierarchy level in the table will repeat the same value as its next higher level. The predefined level is 20, which means if one path of the tree only has five levels, then levels six through 20 will repeat the value of level five.

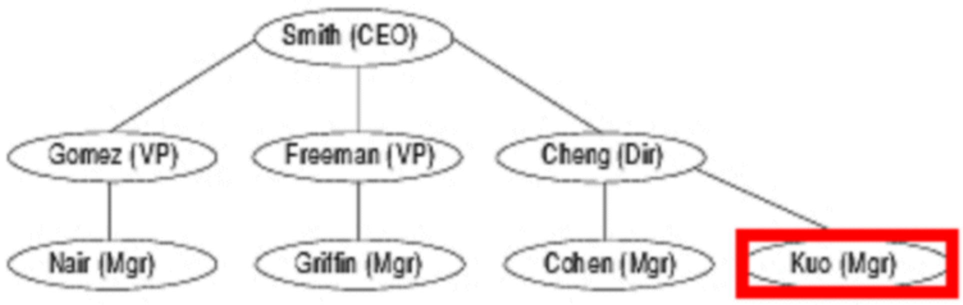
For the leaf node in the detail tree shown in Figure 5–1, W_HIERARCHY_D will have a record for every existing record within this detail range. The extraction process will find the single record in the relevant table according to the tree's structure. For example, in the CA node (the one with the red square in Figure 5–1), the range is 2334-2434 and 3001-3001. The state table (for example, State_TBL) contains two records: 2340 and 3001. When generating the CA node, the extraction process will extract these two records and put them into W_HIERARCH_D, as shown in Table 5–11.

Table 5–11 Example of CSV Chartfield Mapping Values

HIER1_CODE	HIER2_CODE	HIER3_CODE	HIER4_CODE	HIER5_CODE
US	W	CA	2340	2340
US	W	CA	3001	3001

With both the Summer tree and detail tree structure types, the lowest level node is always a node with a specific value. For the winter tree structure type, the lowest level code is the same as the higher level, as illustrated in Figure 5–2.

Figure 5–2 Winter Loose-Level Tree



The record 'Kuo' is stored in the database as shown in Table 5–12.

Table 5–12 Winter Loose-Level Tree

HIER1_CODE	HIER2_CODE	HIER3_CODE	HIER4_CODE	HIER5_CODE
Smith	Cheng	Kuo	Kuo	Kuo

Oracle BI Applications uses temporary tables and staging tables to handle the extraction and load process of the tree. These tables are standard and not tree-specific or tree-structure specific. Table 5–13 lists the tables used during the GL Hierarchy extraction process. The sequence indicates the order in which these tables are populated.

Table 5–13 GL Chartfield Hierarchy

Sequence	Table Name
1	W_PSFT_TREE_TMP
2	W_PSFT_GLHIER_CF_DH_TMP
3	W_PSFT_GLHIER_DTLRGE_DH_TMP
4	W_PSFT_GLHIER_VERT_DH_TMP
5	W_PSFT_GLHIER_FLAT_DH_TMP
6	W_HIERARCHY_DS

Table 5–14 lists the tables used during the Internal Organization Hierarchy extraction process.

Table 5–14 Internal Organization Hierarchy

Sequence	Table Name
1	W_PSFT_INT_ORG_TREE_TMP
2	W_PSFT_INT_ORG_GLBU_DH_TMP
3	W_PSFT_INT_ORG_DTLRGE_DH_TMP
4	W_PSFT_INT_ORG_VERT_DH_TMP
5	W_PSFT_INT_ORG_FLAT_DH_TMP
6	W_INT_ORG_DHS
7	W_INT_ORG_DH

Note: The hierarchy related extraction and load tasks are inactivated in the standard application. To activate the hierarchy extractions and load tasks, you need to activate certain configuration tags in DAC, as described in the following section.

Using DAC Configuration Tags to Activate and Inactivate Tasks

The Configuration Tag feature in DAC enables you to activate or inactivate a certain set of tasks in the ETL process. When inactivated, the set of tasks will not be executed when the Execution Plan runs. When activated, the set of tasks will be executed when the Execution Plan runs.

Oracle BI Applications includes two configuration tags for hierarchy extractions and loads, as follows:

- 'PeopleSoft - Populate Non Business Unit Related Hierarchies,' which contains all but business unit-related hierarchies' extractions and load tasks.
- 'PeopleSoft - Populate Business Unit Related Hierarchies', which contains business unit-related hierarchies' extractions and load tasks.

By activating and inactivating configuration tags in a Subject Area, you can control whether the hierarchy extraction and load tasks get executed in the Execution Plan.

To activate or inactivate a set of tasks in your ETL process:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Subject Areas tab.
3. For each Financials Subject Area, do the following:
   a. Select the Financials Subject Area in the top pane, for example Cost of Goods Sold, General Ledger, and so on.
   b. In the bottom pane, display the Configuration Tags subtab.
   c. Deselect the Inactive check box of the hierarchy you want to enable, as follows:

      - For GL Chartfield, deselect the Inactive check box for PeopleSoft - Populate Non Business Unit Related Hierarchies.

- For Business Unit, deselect the Inactive check box for PeopleSoft - Populate Business Unit Related Hierarchies.

d. Save the changes.

4. Click Assemble to reassemble the Subject Areas that you have modified.
5. Go to the Execute view, and display the Execute view.
6. For every Execution Plan that contains the Subject Areas that you modified, rebuild the Execution Plan.
7. Save the Execution Plans that you modified.

Sample Mappings for Internal Organization Hierarchy and GL Chartfield Hierarchy

Oracle BI Applications provides two sets of mappings to extract Internal Organization Hierarchy and General Ledger Chartfield Hierarchy related trees. You can customize the extractions by copying the mappings and modifying the parameters. For instructions, see Section 5.3.3.3, "General Ledger Chartfield and Chartfield Hierarchy Extraction Customization."

5.3.3.3 General Ledger Chartfield and Chartfield Hierarchy Extraction Customization

This section provides an example of how to customize mappings for a GL Chartfield and Hierarchy extraction. In this example, the extraction involves a tree named DEPARTMENT_HIERARCHY with SetID XYZ on chartfield Department, which uses DEPT_TBL as its structure's Record Name and DEPTID as the Field Name.

The mappings that require customization for this example are shown in Table 5–15.

Table 5–15 General Ledger Chartfield Hierarchy Mappings

Execution Sequence	Mapping Name	Needs Customization?
1	SDE_PSFT_Stage_GLHierarchy_AccountHierarchy_Extract	Y
2	SDE_PSFT_Stage_GLHierarchy_GetAccountChartField	Y
3	SDE_PSFT_GLSegmentDimension_GLAccount	Y
4	SDE_PSFT_Stage_GLHierarchy_GetHierarchyLevel	N
5	SDE_PSFT_Stage_GLHierarchy_DeriveRange	N
6	SDE_PSFT_Stage_GLHierarchy_Flatten	N
7	SDE_PSFT_GLHierarchy_Derive	N

The following procedure shows the steps required to perform this extraction. Note that if your chartfields do not have hierarchies/trees defined in your PeopleSoft system, you might still want to extract just the chartfields and ignore the hierarchy extraction. To do this, in the following steps, perform only the tasks that correspond to the mapping SDE_PSFT_GLSegmentDimension_xyz and ignore the tasks that correspond to mappings SDE_PSFT_Stage_GLHierarchy_xyz.

To customize mappings for a GL Chartfield and Hierarchy extraction:

1. In Informatica PowerCenter Designer, create new mappings by copying the following existing mappings:
   - SDE_PSFT_Stage_GLHierarchy_AccountHierarchy_Extract

- SDE_PSFT_Stage_GLHierarchy_GetAccountChartField
- SDE_PSFT_GLSegmentDimension_GLAccount

Name the new mappings as follows:

- SDE_PSFT_Stage_GLHierarchy_DepartmentHierarchy_Extract
- SDE_PSFT_Stage_GLHierarchy_GetDepartmentChartField
- SDE_PSFT_GLSegmentDimension_Department

2. Open the mapping SDE_PSFT_Stage_GLHierarchy_DepartmentHierarchy_Extract and edit the settings as follows:

   a. In SDE_PSFT_Stage_GLHierarchy_DepartmentHierarchy_Extract, open the source definition PSFT_TREE_VERTICAL_FLATTENING. Under the 'Properties' tab, change the Tree Name to DEPARTMENT_HIERARCHY, set the SetID to XYZ, and set the Effective date to 01/01/1900 00:00:00. Note that Tree Name, SetID, and Effective Date must be the same as the tree definition in the OLTP system.

   b. Open the Exp_TreeName_SetID Expression transformation, and change the values as follows:

      - Change the value of TREE_NAME to DEPARTMENT_HIERARCHY.

      - Change the value of SETID to XYZ.

      - Change the value of CHARTFIELD to Department. (This name is the same as that defined in file_glacct_segment_config_psft.csv.)

3. Open the mapping SDE_PSFT_Stage_GLHierarchy_GetDepartmentChartField and edit the settings as follows:

   - In the SDE_PSFT_Stage_GLHierarchy_GetDepartmentChartField mapping, change the source table and Source Qualifier to DEPT_TBL.

   - Connect SETID, DEPTID, EFFDT, EFF_STATUS, and DESCR from the Source Qualifier to the corresponding fields on the filter `Filter_GLACCT` (connect DEPTID to CHARTFIELD).

   - In the Expression transformation Exp_TMPGLACCT (the last Expression transformation of this mapping), change the Expression field SOURCE_CHARTFIELD_NAME to Department. (This is the same name as the source table.)

4. Open the mapping SDE_PSFT_GLSegmentDimension_Department and edit the settings as follows:

   a. Delete the source, then import DEPT_TBL from source system through the Source Analyzer view, drag the imported source to this mapping, and delete the original Source and Source Qualifier after the import.

   b. Connect SETID, DEPTID, EFFDT, EFF_STATUS, and DESCR from the Source Qualifier to the corresponding fields in the filter `Filter_GLACCT` (connect DEPTID to CHARTFIELD).

   c. In the Expression transformation Exp_CHARTFIELD_NAME (the last Expression transformation of this mapping), change the Expression field SOURCE_CHARTFIELD_NAME to Department.

5. Save all your changes in the Informatica folder.

6. Add corresponding sessions and workflows in Informatica PowerCenter Workflow Manager, as follows:

a. In Informatica PowerCenter Workflow Manager, locate the appropriate PeopleSoft folder (for example, SDE_PSFT_*Version*_Adaptor).

b. Open the Task Developer and create a new Workflow for the mappings you just created.

   Tip: To quickly create a new Workflow, copy the SDE_PSFT_Stage_GLHierarchy_AccountHierarchy_Extract workflow and rename it to SDE_PSFT_Stage_GLHierarchy_DepartmentHierarchy_Extract.

c. Edit the task in the new SDE_PSFT_Stage_GLHierarchy_DepartmentHierarchy_Extract Workflow, and use the various tabs on the Edit Tasks dialog to modify the parameters as specified in Table 5–16.

   Note: Do not change the values of the other mappings.

Table 5–16 General Ledger Chartfield Hierarchy Mappings

Tab Name	Parameter Name	Value
Properties	Parameter Filename	SDE_PSFT_88_Adaptor.*mapping_name*.log
Properties	$Source connection value	$DBConnection_OLTP
Properties	$Target connection value	$DBConnection_OLAP
Properties	DTM buffer size	32000000
Config Object	Default buffer block size	128000
Config Object	Stop on errors	1
Config Object	Override tracing	Normal
Mapping	Connection W_PSFT_TREE_TMP - Rational	$DBConnection_OLAP
Mapping	Connection SQ_PSFT_TREE_VERTICAL_FLATTENING - Application	$DBConnection_OLTP
Mapping	Connection W_PSFT_TREE_TMP1 - Rational	$DBConnection_OLAP
Mapping	Connection Lkp_Tree_ChartField - Rational	$DBConnection_OLTP
Mapping	SDE_PSFT_Stage_GLHierarchy_DepartmentHierarchy_Extract – Session $Source Connection Value	$DBConnection_OLTP
Mapping	SDE_PSFT_Stage_GLHierarchy_DepartmentHierarchy_Extract – Session $Target Connection Value	$DBConnection_OLAP

d. Save your changes.

e. In Informatica PowerCenter Workflow Manager, create a workflow with the same name as the session.

f. Drag the newly created session to the panel.

g. Connect the start with the session.

h. Create a similar session and workflow for SDE_PSFT_GLSegmentDimension_Department.

7. Save and check-in all your changes in Informatica PowerCenter Designer and Informatica PowerCenter Workflow Manager.
8. Reconfigure the DAC settings, as follows:
   a. Log in to DAC, open your custom container, and display the Tasks tab.
   b. Add the three workflows that you just created as new tasks.

      - SDE_PSFT_Stage_GLHierarchy_DepartmentHierarchy_Extract

      - SDE_PSFT_Stage_GLHierarchy_GetDepartmentChartField

      - SDE_PSFT_GLSegmentDimension_Department

      For information about what values to set task property values to, refer to the following tasks:

      - SDE_PSFT_Stage_GLHierarchy_AccountHierarchy_Extract

      - SDE_PSFT_Stage_GLHierarchy_GetAccountChartField

      - SDE_PSFT_GLSegmentDimension_GLAccount
   c. Right-click each task and select Synchronize tasks.

      The Synchronize tasks process reads the source table and target table information for these tasks from the Informatica Repository. To see this information, go to the Tasks tab and display the Source Tables subtab and the Target Tables subtab.
   d. Add the newly created tasks into the DAC task groups.

      Note: Because intermediate temporary tables are reused, it is necessary to include the newly added tasks to task groups to prevent over-loading or pretruncating of data. GL Hierarchy Extract has the following task groups:

      - TASK_GROUP_Extract_GLHierarchy

      - TASK_GROUP_Extract_GLHierarchy_GetChartFieldValues

      - TASK_GROUP_Extract_GLSegmentDimension

      For this example, you need to include the newly added tasks into task groups as follows:

      - Add SDE_PSFT_Stage_GLHierarchy_DepartmentHierarchy_Extract to the task group TASK_GROUP_Extract_GLHierarchy

      - Add SDE_PSFT_Stage_GLHierarchy_GetDepartmentChartField to task group TASK_GROUP_Extract_GLHierarchy_GetChartFieldValues

      - Add SDE_PSFT_GLSegmentDimension_Department to task group TASK_GROUP_Extract_GLSegmentDimension

      The steps for adding a task to a task group are as follows:

      - In DAC, go to the Design view, and then select the Task Group tab.

      - In the top pane, select the appropriate task group.

      - Display the Child Tasks subtab, and then click Add/Remove in the bottom pane toolbar.

      - Query for the appropriate task, and then click Add.

      - Click OK, and then click Save.

e. Add SDE_PSFT_Stage_GLHierarchy_AccountHierarchy_Extract and SDE_PSFT_Stage_GLHierarchy_GetAccountChartField tasks as configuration tags in DAC:

   - In the Design view, click Configuration Tags.

   - Query for the tag PeopleSoft - Populate Non Business Unit Related Hierarchies.

   - Display the Tasks subtab, and then click Add/Remove in the bottom pane toolbar.

   - Query for the new mappings you just entered into DAC.

   - Select the mappings, and add them into the right panel.

   - Click OK, and then click Save.

f. Reassemble all the financial-related Subject Areas:

   - In the Subject Areas tab, select the Subject Area, and click Assemble in the tool bar.

   - Repeat this step for all of the financial-related Subject Areas (for example, Cost of Goods Sold, General Ledger, Payables, Receivables, Revenue)

g. Rebuild the Execution Plans that contain the Subject Areas you reassembled:

   - Go to the Execute view, and display the Execution Plans tab.

   - Select the appropriate Execution Plan (for example, Financials_PeopleSoft).

   - Click Build in the toolbar.

9. Make the following changes in the RPD metadata using the Oracle BI Administration tool. The metadata contains multiple logical tables that represent each GL Segment, such as Dim - GL Segment1, Dim - GL Segment2, and so on. Because all these logical tables are mapped to the same physical table, W_GL_SEGMENT_D, a filter should be specified in the logical table source of these logical tables to restrain the output of the logical table to represent only that particular segment.

   You must set the filter on the physical column SEGMENT_LOV_ID to "CHARTFIELD NAME~SET_ID" (in other words, the chartfield name and setid concatenated with a '~' character) that are applicable for that particular segment. Note the following:

   ■ The "CHARTFIELD NAME" is the same as the CHARTFIELD NAME you specified in the above mappings that extract the trees and/or chartfields. It is also the same as the one defined in the file_glacct_segment_config_psft.csv file.

   ■ The "SET_ID" is the same as the SET_ID you specified in the above mappings that extract the trees and/or the setids that were extracted in the above mappings that extract chartfields. You might have multiple setids for a particular segment, but only one chartfield name for each segment.

   To specify a filter in the Business Model and Mapping layer of the Oracle BI Repository, do the following:

   a. Using the Administration Tool, open OracleBIAnalyticsApps.rpd.

      The OracleBIAnalyticsApps.rpd file is located at:

      ```
      ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_
      obisn\repository
      ```

b. Expand each logical table, for example, Dim - GL Segment1, and open the logical table source under it.

c. Display the Content tab.

d. In the 'Use this WHERE clause…' box, apply a filter on the corresponding physical table alias of W_GL_SEGMENT_D. For example:

```
"Oracle Data Warehouse"."Catalog"."dbo"."Dim_W_GL_SEGMENT_D_
Segment1"."SEGMENT_LOV_ID" IN (comma seperated CHARTFIELD NAME~SET_ID)
```

If there are multiple setids for a segment, enter all possible "CHARTFIElD_ NAME~SETID" values, separated by commas, that correspond to this segment. For example, if this is the "Account" segment, the value for the filter might look like: ('Account~xyz1', 'Account~xyz2', 'Account~xyz3'), where xyz1, xyz2, and xyz3 are the setids that correspond to your Account chartfield values.

10. Oracle Financial Analytics supports up to 30 segments in the GL Account dimension, and by default delivers 10 GL Segment dimensions in the RPD. If you need more than 10 GL Segments, perform the following steps to add new segments:

a. In the Physical layer:

Create a new physical alias of W_GL_SEGMENT_D as "Dim_W_GL_ SEGMENT_D_SegmentXX". To do this, right-click the physical table W_GL_ SEGMENT_D and select **New Object** and then **Alias**. Name the new alias Dim_W_GL_SEGMENT_D_SegmentXX. Similarly, create a new alias of W_ HIERARCHY_D as "Dim_W_HIERARCHY_D_SegmentXX".

In the Physical Diagram, create a Physical Foreign Key between Dim_W_ HIERARCHY_D_SegmentXX and Dim_W_GL_SEGMENT_D_SegmentXX similar to the one between Dim_W_HIERARCHY_D_Segment1 and Dim_W_ GL_SEGMENT_D_Segment1. The direction of the foreign key should be from W_HIERACHY_D to W_GL_SEGMENT_D; for example, on a '0/1':N cardinality join, W_HIERARCHY_D will be on the '0/1' side and W_GL_ SEGMENT_D will be on the 'N' side. See *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* for more information about how to create physical foreign key joins.

Similarly, create a physical foreign key join between Dim_W_GL_SEGMENT_ D_SegmentXX and Dim_W_GL_ACCOUNT_D, with W_GL_SEGMENT_D on the '1' side and W_GL_ACCOUNT_D on the 'N' side.

Save your changes.

b. In the Business Model and Mapping layer:

Create a new logical table "Dim - GL SegmentXX" similar to "Dim – GL Segment1". This logical table should have a logical table source that is mapped to the physical tables created above (for example, it will have both Dim_W_ GL_SEGMENT_D_SegmentXX and Dim_W_HIERARCHY_D_SegmentXX). This logical table should also have all attributes similar to "Dim – GL Segment1" properly mapped to the respective physical tables, Dim_W_GL_ SEGMENT_D_SegmentXX and Dim_W_HIERARCHY_D_SegmentXX.

In the Business Model Diagram, create a logical join from "Dim – GL SegmentXX" to all the relevant logical fact tables similar to "Dim – GL Segment1", with the GL Segment Dimension Logical table on the '0/1' side and the logical fact table on the 'N' side. To see all the relevant logical fact tables, first include Dim – GL Segment1 on the Business Model Diagram, and then right-click that table and select **Add Direct Joins**.

Add the content filter in the logical table source of "Dim – GL SegmentXX" as described in the previous step.

Create a dimension by right-clicking "Dim – GL SegmentXX", and select **Create Dimension**. Rename this to "GL SegmentXX". Make sure the drill-down structure is similar to "GL Segment1". If you are not sure how to do this, follow these steps: By default, the dimension will have two levels: the Grand Total Level and the Detail Level. Rename these levels to "All" and "Detail" respectively. Right-click the "All" level and select "New Object" and then "Child Level". Name this level as Level1. Similarly, create a level under Level1 and name it as Level2. Repeat this process until you have Level19 under Level18. Now drag the "Detail" level under "Level19" so that "Detail" is the last level of the hierarchy. Now, from the new logical table Dim - GL SegmentXX, drag Level1 Code and Level1 Name attributes to the "Level1" level of the hierarchy. Now go the Properties of the Level and from the Keys tab, create two new keys, one for Level1 Code and one for Level1 Name. When you create keys, make sure the 'Use for Drilldown' option is OFF for Level1 Code and ON for Level1 Name. Also, make sure the Primary Key drop-down is set to Level1 Code. Now, proceed similarly to all 19 levels by dragging the appropriate two attributes to the appropriate levels and create the keys similar to above. For the "Detail" level, drag the Level20 Code and Level20 Name attributes into it and create the keys similar to above.

Open the Logical Table Source of the new logical table created, Dim - GL SegmentXX. Set the Aggregation Content in the Content tab by setting the Logical Level to "Detail" level of the "GL SegmentXX" dimension/hierarchy created above.

Similarly, we need to set the aggregation content to all relevant fact logical table sources. Open all Logical Table Sources of all the logical fact tables that are relevant to the new logical table one at a time. Go to the Content tab. If you see that it is set to the "Detail" level of other GL Segment Dimensions such as GL Segment1, GL Segment2, and so on, then set it to the "Detail" level of the "GL Segment XX" dimension/hierarchy created above. If not, skip that logical table source and go to the next one.

c. Drag your new "Dim - GL Segment XX" dimensions into the appropriate subject areas in the Presentation layer. Typically, you can expose these GL Segment dimensions in all subject areas where the GL Account dimension is exposed. You can also find all appropriate subject areas by right-clicking Dim – GL Segment1 and select Query Related Objects, then selecting Presentation, and then selecting Subject Area.

d. Save your changes and check global consistency.

11. Each GL Segment denotes a certain meaningful CHARTFIELD NAME~SET_ID in your OLTP. To clearly identify each segment in the report, you can rename the presentation table "GL SegmentX", logical dimension "GL SegmentX", and logical table "Dim - GL SegmentX" according to its own meaning.

    For example, if you populate Program chartfield into Segment1, you can rename logical table "Dim - GL Segment1" as "Dim - GL Segment Program" or any other appropriate name and then rename the tables in the Presentation layer accordingly.

5.3.3.4 How to Set Up Default Fiscal Calendars For PeopleSoft

In Oracle Business Intelligence Applications Release 7.9.6.3, the semantic layer (RPD) metadata contains session variables to store the current fiscal year, fiscal quarter, fiscal period and so on for a user. Since multiple fiscal calendars are supported in Release 7.9.6.3, you need to get the default fiscal calendar for a user based on the Ledger or Business Unit assigned to the user, and then get the current fiscal year, quarter and so on based on this default fiscal calendar.

The following initialization blocks need to be created in the RPD metadata to get the default fiscal calendar for a user. These initialization blocks read information from the PeopleSoft security tables such as PS_SEC_BU_OPR, PS_SEC_BU_CLS, PS_SEC_LEDGER_OPR, PS_SEC_LEDGER_CLS.

To set up default Fiscal Calendar for PeopleSoft:

1. Create an initialization block to get one Business Unit for a user. This could be a GL BU, AP BU, AR BU, and so on. Follow these steps:

   a. Create a new initialization block called Operating Unit for Fiscal Calendar, and populate a variable called OU_ORG_FSCL_CALENDAR.

   b. Use the following SQL for the initialization block:

   ```
   SELECT MAX(BUSINESS_UNIT) FROM (
   SELECT BUSEC.BUSINESS_UNIT
   FROM PS_SEC_BU_OPR BUSEC, PS_INSTALLATION_FS INST
   WHERE INST.SECURITY_TYPE = 'O' AND BU_SECURITY = 'Y'AND BUSEC.OPRID =
   ':USER'
   UNION ALL
   SELECT BUSEC.BUSINESS_UNIT
   FROM PS_SEC_BU_CLS BUSEC, PS_INSTALLATION_FS INST, PSOPRDEFN OPR
   WHERE INST.SECURITY_TYPE = 'C' AND BU_SECURITY = 'Y' AND BUSEC.OPRCLASS =
   OPR.OPRCLASS AND OPR.OPRID = ':USER');
   ```

   Note: You need to use the PeopleSoft OLTP Connection Pool for this initialization block.

2. Create an initialization block to get one GLBU+Ledger combination for a user, because calendar is defined at a GLBU+Ledger combination in PeopleSoft, not on Ledger alone. Follow these steps

   a. Create a new initialization block called Ledger for Fiscal Calendar, and populate a variable called LEDGER_FSCL_CALENDAR.

   b. Use the following SQL for the initialization block:

   Note: Remove the comments in the SQL below (lines beginning with "--") from the actual SQL in the initialization block.

   ```
   SELECT MAX(LEDGER_ID_FOR_MCAL) FROM (<insert the appropriate SQL from the
   SQL code below, based on the user level and business filters>.
   ```

   -- If security is set at user level with filters on Business Unit and Ledger.

   ```
   SELECT A.BUSINESS_UNIT||'~'||C.SETID||'~'||C.LEDGER LEDGER_ID_FOR_MCAL
   FROM PS_SEC_BU_OPR BUSEC, PS_SEC_LEDGER_OPR LEDSEC, PS_BU_LED_GRP_TBL A,
   PS_SET_CNTRL_REC B, PS_LED_GRP_LED_TBL C, PS_INSTALLATION_FS INST
   WHERE BUSEC.BUSINESS_UNIT = A.BUSINESS_UNIT AND LEDSEC.LEDGER_GROUP =
   A.LEDGER_GROUP AND LEDSEC.LEDGER = C.LEDGER AND
   A.BUSINESS_UNIT = B.SETCNTRLVALUE AND B.RECNAME = 'LED_GRP_LED_TBL' AND
   B.SETID = C.SETID AND
   A.LEDGER_GROUP = C.LEDGER_GROUP
   AND INST.SECURITY_TYPE = 'O' AND BU_SECURITY = 'Y' AND LEDGER_SECURITY =
   ```

```
'Y' AND BUSEC.OPRID = ':USER' AND LEDSEC.OPRID = ':USER' UNION ALL
```

-- If security is set at user level with filters on Business Unit only.

```
SELECT A.BUSINESS_UNIT||'~'||C.SETID||'~'||C.LEDGER LEDGER_ID_FOR_MCAL
FROM PS_SEC_BU_OPR BUSEC, PS_BU_LED_GRP_TBL A, PS_SET_CNTRL_REC B, PS_
LED_GRP_LED_TBL C, PS_INSTALLATION_FS INST
WHERE BUSEC.BUSINESS_UNIT = A.BUSINESS_UNIT AND
A.BUSINESS_UNIT = B.SETCNTRLVALUE AND B.RECNAME = 'LED_GRP_LED_TBL' AND
B.SETID = C.SETID AND
A.LEDGER_GROUP = C.LEDGER_GROUP
AND INST.SECURITY_TYPE = 'O' AND BU_SECURITY = 'Y' AND LEDGER_SECURITY =
'N' AND BUSEC.OPRID = ':USER' UNION ALL
```

-- If security is set at user level with filters on Ledger only.

```
SELECT A.BUSINESS_UNIT||'~'||C.SETID||'~'||C.LEDGER LEDGER_ID_FOR_MCAL
FROM PS_SEC_LEDGER_OPR LEDSEC, PS_BU_LED_GRP_TBL A, PS_SET_CNTRL_REC B,
PS_LED_GRP_LED_TBL C, PS_INSTALLATION_FS INST
WHERE
LEDSEC.LEDGER_GROUP = A.LEDGER_GROUP AND LEDSEC.LEDGER = C.LEDGER AND
A.BUSINESS_UNIT = B.SETCNTRLVALUE AND B.RECNAME = 'LED_GRP_LED_TBL' AND
B.SETID = C.SETID AND
A.LEDGER_GROUP = C.LEDGER_GROUP
AND INST.SECURITY_TYPE = 'O' AND BU_SECURITY = 'N' AND LEDGER_SECURITY =
'Y' AND LEDSEC.OPRID = ':USER' UNION ALL
```

-- If security is set at permission list level with filters on Business Unit and Ledger.

```
SELECT A.BUSINESS_UNIT||'~'||C.SETID||'~'||C.LEDGER LEDGER_ID_FOR_MCAL
FROM PS_SEC_BU_CLS BUSEC, PS_SEC_LEDGER_CLS LEDSEC, PS_BU_LED_GRP_TBL A,
PS_SET_CNTRL_REC B, PS_LED_GRP_LED_TBL C, PS_INSTALLATION_FS INST,
PSOPRDEFN OPR
WHERE BUSEC.BUSINESS_UNIT = A.BUSINESS_UNIT AND LEDSEC.LEDGER_GROUP =
A.LEDGER_GROUP AND LEDSEC.LEDGER = C.LEDGER AND
A.BUSINESS_UNIT = B.SETCNTRLVALUE AND B.RECNAME = 'LED_GRP_LED_TBL' AND
B.SETID = C.SETID AND
A.LEDGER_GROUP = C.LEDGER_GROUP
AND INST.SECURITY_TYPE = 'C' AND BU_SECURITY = 'Y' AND LEDGER_SECURITY =
'Y' AND LEDSEC.OPRCLASS = OPR.OPRCLASS AND BUSEC.OPRCLASS = OPR.OPRCLASS
AND OPR.OPRID = ':USER' UNION ALL
```

-- If security is set at permission list level with filters on Business Unit only.

```
SELECT A.BUSINESS_UNIT||'~'||C.SETID||'~'||C.LEDGER LEDGER_ID_FOR_MCAL
FROM PS_SEC_BU_CLS BUSEC, PS_BU_LED_GRP_TBL A, PS_SET_CNTRL_REC B, PS_
LED_GRP_LED_TBL C, PS_INSTALLATION_FS INST, PSOPRDEFN OPR
WHERE BUSEC.BUSINESS_UNIT = A.BUSINESS_UNIT AND
A.BUSINESS_UNIT = B.SETCNTRLVALUE AND B.RECNAME = 'LED_GRP_LED_TBL' AND
B.SETID = C.SETID AND
A.LEDGER_GROUP = C.LEDGER_GROUP
AND INST.SECURITY_TYPE = 'C' AND BU_SECURITY = 'Y' AND LEDGER_SECURITY =
'N' AND BUSEC.OPRCLASS = OPR.OPRCLASS AND OPR.OPRID = ':USER' UNION ALL
```

-- If security is set at permission list level with filters on Ledger only.

```
SELECT A.BUSINESS_UNIT||'~'||C.SETID||'~'||C.LEDGER LEDGER_ID_FOR_MCAL
FROM PS_SEC_LEDGER_CLS LEDSEC, PS_BU_LED_GRP_TBL A, PS_SET_CNTRL_REC B,
PS_LED_GRP_LED_TBL C, PS_INSTALLATION_FS INST, PSOPRDEFN OPR
WHERE LEDSEC.LEDGER_GROUP = A.LEDGER_GROUP AND LEDSEC.LEDGER = C.LEDGER AND
A.BUSINESS_UNIT = B.SETCNTRLVALUE AND B.RECNAME = 'LED_GRP_LED_TBL' AND
```

```
B.SETID = C.SETID AND
A.LEDGER_GROUP = C.LEDGER_GROUP
AND INST.SECURITY_TYPE = 'C' AND BU_SECURITY = 'N' AND LEDGER_SECURITY =
'Y' AND LEDSEC.OPRCLASS = OPR.OPRCLASS AND OPR.OPRID = ':USER');
```

Note: You need to use the PeopleSoft OLTP Connection Pool for this initialization block.

3. Modify the appropriate initialization block to get a default Calendar for a user, as follows:

   a. Open the initialization block "Mcal Calendar Wid Identification."

      The default SQL code is:

      ```
      select max(MCAL_CAL_WID) from VALUEOF(OLAPTBO).W_MCAL_CONTEXT_G where
      ((ledger_id=valueof(NQ_SESSION.LEDGER)) or (org_id=valueof(NQ_SESSION.OU_
      ORG)')) AND CLASS='GL'
      ```

   b. Replace NQ_SESSION.LEDGER with NQ_SESSION.LEDGER_FSCL_CALENDAR and NQ_SESSION.OU_ORG with NQ_SESSION.OU_ORG_FSCL_CALENDAR.

   c. Edit the Execution Precedence, and remove the existing two initialization blocks (Ledgers and Operating Unit Organizations) that are currently set.

   d. Add the two new initialization blocks you created in steps 1 and 2.

      Note: If Security is completely turned OFF in PeopleSoft, you can disable all these initialization blocks.

5.3.3.5 How to Enable Project Analytics Integration with Financial Subject Areas

You can enable Oracle Financial Analytics for PeopleSoft to use dimension tables in Oracle Project Analytics. You can only perform this integration if you have licensed Oracle Project Analytics.

To enable the integration:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Select each Subject Area, and then in the Configuration Tags tab, clear the Inactive check box for the Enable Project Dimensions configuration tag.
3. Assemble the Subject Areas.
4. Click the Execute button and build the Execution Plan for the Subject Areas that you updated.

5.3.3.6 Configuring Number of Days based Metrics for PeopleSoft

For certain metrics to function properly, you must configure the following two internal metrics in the Oracle BI Applications metadata repository (RPD):

- # of Elapsed Days
- # of Cumulative Elapsed Days

These metrics affect the calculation of other metrics, such as Days Sales Outstanding, Days Payables Outstanding, AP Turnover, AR Turnover, and so forth.

To configure Number of Days based metrics:

1. Using the Administration Tool, open OracleBIAnalyticsApps.rpd.

The OracleBIAnalyticsApps.rpd file is located at:

```
ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_
obisn\repository
```

2. In the Business Model and Mapping layer, go the logical table Fact - Fins - Period Days Count.
3. Under Sources, select the Fact_W_DAY_D_PSFT logical table source.
4. Clear the **Disabled** option in the General tab and click **OK**.
5. Open the other two logical table sources, Fact_W_DAY_D_ORA and Fact_W_DAY_D_PSFT, and select the **Disabled** option.
6. Add the "Fact - Fins - Period Days Count" and "Dim - Company" logical tables to the Business Model Diagram. To do so, right-click the objects and select Business Model Diagram, Selected Tables Only.
7. In the Business Model Diagram, create a new logical join from "Dim - Company" to "Fact - Fins - Period Days Count." The direction of the foreign key should be from the "Dim - Company" logical table to the "Fact - Fins - Period Days Count" table. For example, on a (0,1):N cardinality join, "Dim - Company" will be on the (0/1) side and "Fact - Fins - Period Days Count" will be on the N side.
8. Under the Fact - Fins - Period Days Count logical table, open the "# of Elapsed Days" and "# of Cumulative Elapsed Days" metrics, one at a time.
9. Go to the Levels tab. For the Company dimension, the Logical Level is set to All. Click the X button to remove it. Repeat until the Company dimension does not have a Logical Level setting.
10. Make sure to check Global Consistency to ensure there are no errors, and then save the RPD file.

5.3.3.7 How to Update Dashboard Pages with PeopleSoft Prompts

Data Source specific dashboard prompts are provided with Financial Analytics to accommodate source specific filtering across all application Dashboard pages. You need to add each PeopleSoft dashboard prompt listed in Table 5–17 to its associated dashboard page as part of the application configuration process.

Table 5–17 Financial Analytics Dashboard Pages with Pre-configured PeopleSoft Path and Prompt Names

Dashboard	Dashboard Page	Catalog Shared Folder (Shared Folders/Financials)	PeopleSoft Prompt Name
General Ledger	Overview	/General Ledger/Key Ratios	Oracle PSFT - GL Key Ratios Prompt
General Ledger	Balance Sheet	/General Ledger/Balance Sheet	Oracle PSFT - GL Balance Sheet Prompt
General Ledger	Cash Flow	/General Ledger/Cash Flow	Oracle PSFT - GL Cash Flow Prompt
General Ledger	Budget vs. Actual	/General Ledger/Budget Actual	Oracle PSFT - GL Budget Prompt
General Ledger	Asset Usage	/General Ledger/Asset Usage	Oracle PSFT - GL Asset Usage Prompt
General Ledger	Liquidity	/General Ledger/Liquidity	Oracle PSFT - GL Liquidity Prompt
General Ledger	Financial Structure	/General Ledger/Financial Structure	Oracle PSFT - GL Financial Structure Prompt

Table 5–17 (Cont.) Financial Analytics Dashboard Pages with Pre-configured PeopleSoft Path and Prompt

Dashboard	Dashboard Page	Catalog Shared Folder (Shared Folders/Financials)	PeopleSoft Prompt Name
General Ledger	GL Balance	/General Ledger/Transactions	Oracle PSFT - GL Balance Transactions Prompt
General Ledger	Trial Balance	/General Ledger/Trial Balance	Oracle PSFT - GL Trial Balance Prompt
Payables	Overview	/Payables/Overview	Oracle PSFT - AP Overview Prompt
Payables	AP Balance	/Payables/AP Balance	Oracle PSFT - AP Balance Prompt
Payables	Payments Due	/Payables/Payments Due	Oracle PSFT - AP Payments Due Prompt
Payables	Effectiveness	/Payables/Effectiveness	Oracle PSFT - AP Effectiveness Prompt
Payables	Payment Performance	/Payables/Payment Performance	Oracle PSFT - AP Payment Performance Prompt
Payables	Supplier Report	/Payables/Supplier Report	Oracle PSFT - AP Supplier Report Prompt
Payables	Invoice Details	/Payables/Invoice Details	Oracle PSFT - AP Invoice Details Prompt
Payables	All AP Transactions	/Payables/All AP Transactions	Oracle PSFT - AP Txn Prompt
Profitability	Overview	/Profitability/Overview	Oracle PSFT - GL Profitability Overview Prompt
Profitability	P&L	/Profitability/P&L	Oracle PSFT - GL Profitability P&L Prompt
Profitability	Margins	/Profitability/Margins	Oracle PSFT - GL Profitability Margins Prompt
Profitability	Revenue	/Profitability/Revenue	Oracle PSFT - GL Profitability Revenue Prompt
Profitability	Products	/Profitability/Products	Oracle PSFT - GL Profitability Products Prompt
Profitability	Customers	/Profitability/Customers	Oracle PSFT - GL Profitability Customer Prompt
Receivables	Overview	/Receivables/Overview	Oracle PSFT - AR Overview Prompt
Receivables	AR Balance	/Receivables/AR Balance	Oracle PSFT - AR Balance Prompt
Receivables	Payments Due	/Receivables/Payments Due	Oracle PSFT - AR Payments Due Prompt
Receivables	Effectiveness	/Receivables/Effectiveness	Oracle PSFT - AR Effectiveness Prompt
Receivables	Payment Performance	/Receivables/Payment Performance	Oracle PSFT - AR Payment Performance Prompt
Receivables	Customer Report	/Receivables/Supplier Report	Oracle PSFT - AR Supplier Report Prompt
Receivables	Invoice Details	/Receivables/Invoice Details	Oracle PSFT - AR Invoice Details Prompt
Receivables	All AR Transactions	/Receivables/All AR Transactions	Oracle PSFT - AR Transactions Prompt

To update dashboard pages with PeopleSoft prompts:

These instructions explain how to modify the General Ledger dashboard's Overview page prompt as an example of how to modify a prompt.

1. Access the dashboard page.
2. Click the Page Options button and then select Edit Dashboard to launch Dashboard Editor.
3. Remove the existing dashboard prompt from the top section in Dashboard Editor.

   For the Overview page in the General Ledger dashboard, remove the "Oracle EBS - GL Key Ratios Prompt" from Section 1.

   Note: Remove the prompt, not the Section.

4. From the selection pane in the Saved Content area, browse to the Shared Folders where the dashboard prompt to be used for this dashboard page is stored.

   For the Overview page in the General Ledger dashboard, the catalog path is stored in the following location:

   ```
   /Shared folders/Financials/General Ledger/Key Ratios Prompt name:
   Oracle PSFT - GL Key Ratios Prompt
   ```

5. Drag and drop the dashboard prompt from the shared folder into the section where you removed the prompt in step 3.
6. Click the Save button to save the dashboard page and exit Dashboard Editor.

   This updates the dashboard page with the PeopleSoft prompt.

7. Repeat these steps for all Financial Analytics dashboard pages listed in Table 5–17.

5.3.4 Configuration Steps for Financial Analytics for Oracle’s JD Edwards EnterpriseOne and JD Edwards World

This section contains additional configuration steps that apply to Oracle’s JD Edwards EnterpriseOne and JD Edwards World. It contains the following topics:

- Section 5.3.4.1, "Configuring GL Balances in Oracle’s JD Edwards EnterpriseOne and JD Edwards World"
- Section 5.3.4.2, "Configuring Ledger Type"
- Section 5.3.4.3, "Configuring the DAC Parameter for Receivables"
- Section 5.3.4.4, "Configuring the DAC Parameter when Multicurrency Processing is Disabled in JD Edwards"
- Section 5.3.4.5, "Configuring Number of Days based Metrics"
- Section 5.3.4.6, "How to Update Dashboard Pages with Oracle’s JD Edwards EnterpriseOne and JD Edwards World Prompts"

5.3.4.1 Configuring GL Balances in Oracle’s JD Edwards EnterpriseOne and JD Edwards World

For JD Edwards EnterpriseOne and JD Edwards World GL balances, only Actual Ledger Type is supported. Using a mapplet parameter, you can identify and pass multiple Actual Ledger Types, if required. For Ledger Types AA, A1, and A2, you run the ETL map by passing the value through DAC as AA, A1, A2 (comma separated value) for $$LEDGER_TYPE_ACTUAL. By default, the value for Ledger Type is AA.

W_GL_BALANCE_F supports only Actual ledgers; ledgers with foreign transactions should not be included.

To configure multiple Actual Ledger Types in DAC:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab, and perform a query for the task SDE_JDE_GLBalanceFact.
3. In the Detail area of the selected task, display the Parameters tab. Modify the value in the $$LEDGER_TYPE_ACTUAL parameter, for example:

   ```
   $$LEDGER_TYPE_ACTUAL= 'AA', 'A1', 'A2'
   ```

4. Save your changes.

5.3.4.2 Configuring Ledger Type

Oracle BI Applications extracts JD Edwards EnterpriseOne 'Ledger Type' and maps it into the Ledger Dimension (W_LEDGER_D). This mapping provides:

- The ability to filter or segregate data by Ledger Type for reporting and analysis
- Support for consolidation ('AC') and restatement ledgers ('XA'), in addition to the Actuals and Budget ledgers

Table 5–18 lists the supported and unsupported ledger types.

Table 5–18 Supported and Unsupported Ledgers

Supported Ledgers	Ledgers Not Supported
Actuals, such as 'AA' and 'A1'	Transaction currency - 'CA'
Budget, such as 'BA' and 'B1'	Units - '*U'
Consolidation - 'AC'	Percent of Job complete - 'F%'
Restatement, such as 'XA' and 'YA'	-

Table 5–19 provides an example of how records are populated into the Ledger Dimension. Note that LEDGER_NAME contains the concatenation of Company and Ledger Type. There will be as many entries for company as there are Ledger Types. For example, assuming there are two Ledger Types (AA and A1) and two Companies (00001 and 00050), LEDGER_NAME will contain the four rows shown in the table once the load process is complete.

Table 5–19 Example of Ledger Dimension Table with Support for Ledger Type

LEDGER_NAME	LEDGER_DESC	CURRENCY_CODE	LEDGER_ CATEGORY_CODE	LEDGER_ CATEGORY_NAME
00001~AA	Fixed Asset Company #2	USD	AA	Actuals

Table 5–19 (Cont.) Example of Ledger Dimension Table with Support for Ledger Type

LEDGER_NAME	LEDGER_DESC	CURRENCY_CODE	LEDGER_CATEGORY_CODE	LEDGER_CATEGORY_NAME
00001~A1	Fixed Asset Company #2	USD	A1	User Defined Actuals
00050~AA	SAR TESTING COMPANY	USD	AA	Actuals
00050~A1	SAR TESTING COMPANY	USD	A1	User Defined Actuals

The ETL map is run by passing the Ledger Type value through DAC as comma separated values. By default, the value for Ledger Type is AA.

To configure support for consolidation and restatement ledgers in DAC:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab, and perform a query for the task SDE_JDE_GLBalanceFact.
3. Display the Parameters subtab, and modify the value in the $$LEDGER_TYPE parameter. For example:

   ```
   $$LEDGER_TYPE= 'AA', 'XA', 'AC'
   ```

4. Save your changes.

5.3.4.3 Configuring the DAC Parameter for Receivables

The Transaction Type Code dimension table contains a UDC column for UDCs, which are user defined codes in the JD Edwards EnterpriseOne source system. By default, the UDC is set to "RA" in the dimension table. You can override the value of the $$WRITE_OFF_DOC_TYPE DAC parameter to specify any UDC in the source system.

If you configure the DAC parameter to any other value available in the JD Edwards EnterpriseOne source system, then you must open the domainvalues_src_transaction_type_codes_jde.csv file and modify the UDC column values to reflect the UDC that you entered in the DAC parameter.

The following table is an example of two entries in the CSV file in which you would replace RA in the UDC column with the user defined code specified in the DAC parameter.

DRSY	DRRT	UDC	RECEIPT_TYPE	XACT_CAT_CODE	XACT_CODE	W_XACT_TYPE_CODE	W_XACT_SUBTYPE_CODE
0	DT	RA		ACCT_DOC	RECEIVABLE	APPLICATION	ADJUSTMENT APPLICATION
0	DT	RA	A	ACCT_DOC	RECEIVABLE	APPLICATION	ADJUSTMENT APPLICATION

To configure the $$WRITE_OFF_DOC_TYPE DAC parameter:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Select the Financials - Receivables row.

3. Click Tasks in the Design area on the left side of the screen, and then highlight the SDE_JDE_AR_Transaction_Fact_Extract task.
4. Display the Parameters sub-tab.
5. Select the row with the $$WRITE_OFF_DOC_TYPE parameter, and then double-click in the Value column.
6. In the Enter Parameter Value screen, replace 'RA' with a UDC in the source system. Ensure that the UDC is in single quotes.
7. Open the domainvalues_src_transaction_type_codes_jde.csv file and modify the UDC column values to reflect the UDC that you entered in the DAC parameter.

5.3.4.4 Configuring the DAC Parameter when Multicurrency Processing is Disabled in JD Edwards

LOC_CURR_CODE and ACCT_CURR_CODE in the Data Warehouse are non-nullable fields. Unless they are populated, insertion into the Data Warehouse will fail. If multicurrency processing is disabled in JD Edwards EnterpriseOne or JD Edwards World, then you must set the $$DOC_CURR_CODE parameter in DAC to the domestic or base currency in use. By default, the value of this parameter is set to NULL in DAC.

To configure the $$DOC_CURR_CODE parameter:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Select the Source System Parameters tab.
3. Select the row with the $$DOC_CURR_CODE parameter, and then double-click in the Value column.
4. In the Enter Parameter Value screen, replace NULL with the domestic or base currency in use.

5.3.4.5 Configuring Number of Days based Metrics

For certain metrics to function properly, you must configure the following two internal metrics in the Oracle BI Applications metadata repository (RPD):

- # of Elapsed Days
- # of Cumulative Elapsed Days

These metrics affect the calculation of other metrics, such as Days Sales Outstanding, Days Payables Outstanding, AP Turnover, AR Turnover, and so forth.

To configure Number of Days based metrics:

1. Using the Administration Tool, open OracleBIAnalyticsApps.rpd.

   The OracleBIAnalyticsApps.rpd file is located at:

   ```
   ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_
   obisn\repository
   ```

2. In the Business Model and Mapping layer, go the logical table Fact - Fins - Period Days Count.
3. Under Sources, select the Fact_W_DAY_D_JDE logical table source.
4. Clear the **Disabled** option in the General tab and click **OK**.

5. Open the other two logical table sources, Fact_W_DAY_D_ORA and Fact_W_DAY_D_PSFT, and select the **Disabled** option.
6. Add the "Fact - Fins - Period Days Count" and "Dim - Company" logical tables to the Business Model Diagram. To do so, right-click the objects and select Business Model Diagram, Selected Tables Only.
7. In the Logical Table diagram, create a new logical join from "Dim - Company" to "Fact - Fins - Period Days Count." The direction of the foreign key should be from the "Dim - Company" logical table to the "Fact - Fins - Period Days Count" table. For example, on a (0,1):N cardinality join, "Dim - Company" will be on the (0/1) side and "Fact - Fins - Period Days Count" will be on the N side.
8. Similarly, in the Business Model Diagram, add the following logical tables:
   - Fact - Fins - Period Days Count
   - Dim - Ledger
   - Dim - Receivables Org
   - Dim - Payables Org
9. Click the joins between the three Logical Dimension table joins and the Logical Fact table and delete them.
10. Under the Fact - Fins - Period Days Count logical table, open the "# of Elapsed Days" and "# of Cumulative Elapsed Days" metrics, one at a time.
11. Go to the Levels tab. For the Company dimension, the Logical Level is set to All. Click the X button to remove it. Repeat until the Company dimension does not have a Logical Level setting.
12. Set the Logical Level to All for these three dimensions:
    - Ledger
    - Payables Org
    - Receivables Org
13. Make sure to check Global Consistency to ensure there are no errors, and then save the RPD file.

5.3.4.6 How to Update Dashboard Pages with Oracle's JD Edwards EnterpriseOne and JD Edwards World Prompts

Data-source specific dashboard prompts are provided with Oracle Financial Analytics to accommodate source specific filtering across all application Dashboard pages. You need to add each JD Edwards EnterpriseOne or JD Edwards World dashboard prompt listed in Table 5–20 to its associated dashboard page as part of the application configuration process. Follow the steps in this section to update existing dashboard prompts with the designated JD Edwards EnterpriseOne or JD Edwards World prompt.

Table 5–20 Financial Analytics Dashboard Pages with Preconfigured JD Edwards EnterpriseOne or JD Edwards World Path and Prompt Names

Dashboard	Dashboard Page	Catalog Shared Folder (Shared Folders/Financials)	JD Edwards EnterpriseOne/JD Edwards World Prompt Name
General Ledger	Overview	/General Ledger/Key Ratios	Oracle JDE - GL Key Ratios Prompt
General Ledger	Balance Sheet	/General Ledger/Balance Sheets	Oracle JDE - GL Balance Sheet Prompt
General Ledger	Cash Flow	/General Ledger/Cash Flow	Oracle JDE - GL Cash Flow Prompt
General Ledger	Budget vs. Actual	/General Ledger/Budget Actual	Oracle JDE - GL Budget Prompt
General Ledger	Asset Usage	/General Ledger/Asset Usage	Oracle JDE - GL Asset Usage Prompt
General Ledger	Liquidity	/General Ledger/Liquidity	Oracle JDE - GL Liquidity Prompt
General Ledger	Financial Structure	/General Ledger/Financial Structure	Oracle JDE - GL Financial Structure Prompt
General Ledger	GL Balance	/General Ledger/Transactions	Oracle JDE - GL Balance Transactions Prompt
General Ledger	Trial Balance	/General Ledger/Trial Balance	Oracle JDE - GL Trial Balance Prompt
Payables	Overview	/Payables/Overview	Oracle JDE - AP Overview Prompt
Payables	AP Balance	/Payables/AP Balance	Oracle JDE - AP Balance Prompt
Payables	Payments Due	/Payables/Payments Due	Oracle JDE - AP Payments Due Prompt
Payables	Effectiveness	/Payables/Effectiveness	Oracle JDE - AP Effectiveness Prompt
Payables	Payment Performance	/Payables/Payment Performance	Oracle JDE - AP Payment Performance Prompt
Payables	Supplier Report	/Payables/Supplier Report	Oracle JDE - AP Supplier Report Prompt
Payables	Invoice Details	/Payables/Invoice Details	Oracle JDE - AP Invoice Details Prompt
Payables	All AP Transactions	/Payables/All AP Transactions	Oracle JDE - AP Txn Prompt
Profitability	Overview	/Profitability/Overview	Oracle JDE - GL Profitability Overview Prompt
Profitability	P&L	/Profitability/P&L	Oracle JDE - GL Profitability P&L Prompt
Profitability	Margins	/Profitability/Margins	Oracle JDE - GL Profitability Margins Prompt
Profitability	Revenue	/Profitability/Revenue	Oracle JDE - GL Profitability Revenue Prompt
Profitability	Products	/Profitability/Products	Oracle JDE - GL Profitability Products Prompt
Profitability	Customers	/Profitability/Customers	Oracle JDE - GL Profitability Customers Prompt
Receivables	Overview	/Receivables/Overview	Oracle JDE - AR Overview Prompt
Receivables	AP Balance	/Receivables/AP Balance	Oracle JDE - AR Balance Prompt

Table 5–20 (Cont.) Financial Analytics Dashboard Pages with Preconfigured JD Edwards EnterpriseOne or JD Edwards World Path and Prompt Names

Dashboard	Dashboard Page	Catalog Shared Folder (Shared Folders/Financials)	JD Edwards EnterpriseOne/JD Edwards World Prompt Name
Receivables	Payments Due	/Receivables/Payments Due	Oracle JDE - AR Payments Due Prompt
Receivables	Effectiveness	/Receivables/Effectiveness	Oracle JDE - AR Effectiveness Prompt
Receivables	Payment Performance	/Receivables/Payment Performance	Oracle JDE - AR Payment Performance Prompt
Receivables	Supplier Report	/Receivables/Supplier Report	Oracle JDE - AR Supplier Report Prompt
Receivables	Invoice Details	/Receivables/Invoice Details	Oracle JDE - AR Invoice Details Prompt
Receivables	All AR Transactions	/Receivables/All AR Transactions	Oracle JDE - AR Transaction Prompts

To update dashboard pages with JD Edwards EnterpriseOne or JD Edwards World prompts:

These instructions explain how to modify the General Ledger dashboard's Overview page prompt as an example of how to modify a prompt.

1. Access the dashboard page.
2. Click the Page Options button and then select Edit Dashboard to launch Dashboard Editor.
3. Remove the existing dashboard prompt from the top section in Dashboard Editor.

   For the Overview page in the General Ledger dashboard, remove the "Oracle EBS - GL Key Ratios Prompt" from Section 1.

   Note: Remove the prompt, not the Section.

4. From the selection pane in the Saved Content area, browse to the Shared Folders where the dashboard prompt to be used for this dashboard page is stored.

   For the Overview page in the General Ledger dashboard, the catalog path is stored in the following location:

   ```
   /Shared folders/Financials/General Ledger/Key Ratios Prompt name:
   Oracle JDE - GL Key Ratios Prompt
   ```

5. Drag and drop the dashboard prompt from the shared folder into the section where you removed the prompt in step 3.
6. Click the Save button to save the dashboard page and exit Dashboard Editor.

   This updates the dashboard page with the JD Edwards EnterpriseOne or JD Edwards World prompt.

7. Repeat these steps for all Financial Analytics dashboard pages listed in Table 5–20.

5.3.4.6.1 Additional Dashboard Page Configuration for Ledger Dimension Columns Dashboard prompts defined during the Financial Analytics configuration process will not require any additional changes because Ledger Name will display the new concatenated Ledger Name ("Company~Ledger Type") values.

LEDGER_CATEGORY_NAME and LEDGER_CATEGORY_CODE can be exposed as Dashboard prompts with a few modifications to the original product configuration:

- LEDGER_CATEGORY_NAME is already available in the Presentation Subject Area as part of the Ledger Dimension. Therefore, all that is needed is to define LEDGER_CATEGORY_NAME as a prompt for the required Dashboard page or pages.
- LEDGER_CATEGORY_CODE is available in the RPD Business Model and Mapping (BMM) layer. Therefore, before defining as a Dashboard Page prompt, you will need to expose within the RPD Presentation Subject Area, as a member of the Ledger Dimension.

Table 5–21 summarizes the availability of each column across each area after the standard product configuration, and lists the associated action needed to define as a Dashboard page prompt.

Table 5–21 How to Define Ledger Dimension Columns as Dashboard Page Prompts

Ledger Dimension Column	Dashboard Prompt	Presentation Subject Area	RPD BMM Layer	Action to Define as Dashboard Page Prompt
LEDGER_NAME	Yes	Yes	Yes	None. Already available as a Dashboard prompt.
LEDGER_CATEGORY_NAME	No	Yes	Yes	Define as prompt in Dashboard Page.
LEDGER_CATEGORY_CODE	No	No	Yes	1. Expose in Presentation Subject Area. 2. Define as prompt in Dashboard Page.

6 Configuring Oracle Supply Chain and Order Management Analytics

This section describes how to configure Oracle Supply Chain and Order Management Analytics. It contains the following topics:

- Section 6.1, "Overview of Oracle Supply Chain and Order Management Analytics"
- Section 6.2, "Configuration Required Before A Full Load for Oracle Supply Chain and Order Management Analytics"
- Section 6.3, "Configuration Steps for Controlling Your Data Set"

6.1 Overview of Oracle Supply Chain and Order Management Analytics

The Oracle Supply Chain and Order Management Analytics application enables 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 enables you to evaluate the sales performance of individual sales representatives or departments. Oracle Supply Chain and Order Management Analytics application 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, properly deploy inventory at the right place / 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.

6.2 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. It contains the following topics:

- Section 6.2.1, "Configuration Steps for Oracle Supply Chain and Order Management Analytics for All Source Systems"
- Section 6.2.2, "Configuration Steps for Oracle Supply Chain and Order Management Analytics for Oracle EBS"
- Section 6.2.3, "Configuration Steps for Oracle Supply Chain and Order Management Analytics for Oracle's JD Edwards EnterpiseOne"

6.2.1 Configuration Steps for Oracle Supply Chain and Order Management Analytics for All Source Systems

This section contains configuration steps that apply to all source systems. It contains the following topics:

- Section 6.2.1.1, "How to Disable Project Dimensions"
- Section 6.2.1.2, "Enabling Delete and Primary Extract Sessions for Oracle Supply Chain and Order Management Analytics"

> **Note:** For configuration steps that apply to all analytics modules (for example, Oracle Financial Analytics, Oracle HR Analytics, or Oracle Sales Analytics), see Chapter 3, "Configuring Common Areas and Dimensions."

6.2.1.1 How to Disable Project Dimensions

If you have not licensed or do not want to implement Oracle Projects, you can disable Project dimensions.

To disable project dimensions:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Configuration Tags tab.
3. Query for the tag Enable Project Dimensions.
4. Display the Subject Areas subtab.
5. Select the **Inactive** check box for the Subject Area named 'Supply Chain - Inventory Transactions'.
6. In the upper pane, display the Subject Areas tab, and select the Subject Area named 'Projects'.
7. In lower pane, display the Configuration Tags tab.
8. Select the **Inactive** check box for the Configuration Tag named 'Enable Project Dimensions'.

6.2.1.2 Enabling Delete and Primary Extract Sessions for Oracle Supply Chain and Order Management Analytics

You need to enable the delete and primary extract sessions for Oracle Supply Chain and Order Management Analytics. For more information, see Section 17.8.3.2, "Enabling Delete and Primary Extract Sessions." For general information about primary extract and delete mappings, see Section 17.8, "Configuring Loads."

6.2.2 Configuration Steps for Oracle Supply Chain and Order Management Analytics for Oracle EBS

This section contains configuration steps required before you do a full data load that apply to Oracle EBS.

- Section 6.2.2.1, "About Configuring Domain Values and CSV Worksheet Files for Oracle Supply Chain and Order Management Analytics".
- Section 6.2.2.2, "How to Configure Invoice Type Domain Values".
- Section 6.2.2.3, "How to Configure Pick Types Domain Values".
- Section 6.2.2.4, "How to Configure Order Types Domain Values"
- Section 6.2.2.5, "How to Configure Pick Status Domain Values"
- Section 6.2.2.7, "How to Configure Invoice Status Domain Values"
- Section 6.2.2.8, "How to Configure Order Overall Status Domain Values"
- Section 6.2.2.6, "How to Configure Pay Method Domain Values"
- Section 6.2.2.9, "How to Configure Movement Types Domain Values"
- Section 6.2.2.10, "How to Configure Invoice Source Type Domain Values"
- Section 6.2.2.11, "How to Configure Quantity Types for Product Transactions"
- Section 6.2.2.12, "How to Deploy Objects in Oracle EBS for Exploding the BOM"

> **Note:** If you plan to consume analytics on Bill of Materials (BOM), you need to include the DAC Subject Area "Supply Chain - BOM Items" in your ETL Execution Plan. However, the ETL process for extracting BOM data from Oracle EBS into the OBAW data warehouse is not designed for very high volumes of BOM data. There are known performance issues with initial ETL load for very large BOM data size (for example, more than 100,000 rows of data in EBS BOM). Oracle recommends that you restrict the BOM data size in the initial ETL load by applying appropriate filters (for example, On time or Inventory Organization).

6.2.2.1 About Configuring Domain Values and CSV Worksheet Files for Oracle Supply Chain and Order Management Analytics

Table 6–1 lists the CSV worksheet files and the domain values for Oracle Supply Chain and Order Management Analytics that are located in the $PMServer\LkpFiles folder (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

Table 6–1 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 6.2.2.2, "How to Configure Invoice Type Domain Values."	SDE_ORA_TransactionTypeDimension_SalesInvoiceLines
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 6.2.2.3, "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 6.2.2.4, "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 6.2.2.5, "How to Configure Pick Status Domain Values."	SDE_ORA_StatusDimension_SalesPickLines
domainValues_PayMethodCode_ora.csv	Lists the method code column and the corresponding domain value for the application. For more information about updating the values in this file, see Section 6.2.2.6, "How to Configure Pay Method Domain Values."	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 6.2.2.7, "How to Configure Invoice Status Domain Values."	SDE_ORA_StatusDimension_SalesInvoiceLine

Table 6–1 (Cont.) Domain Values and CSV Worksheet Files for Oracle Supply Chain and Order Management Analytics

Worksheet File Name	Description	Session
domainValues_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 6.2.2.8, "How to Configure Order Overall Status Domain Values."	SDE_ORA_StatusDimension_SalesOrderLineCycle
domainValues_InvoiceSrcTypes_ora11i.csv	Lists the Invoice Source Code and the Source 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 6.2.2.10, "How to Configure Invoice Source Type Domain Values."	SDE_ORA_TransactionSourceDimension

For general information about domain values with CSV worksheet files, see the section about domain values in the section entitled "Customizing the Oracle Business Analytics Warehouse" in *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

Note: Where SQL code is provided in the following sections, you might need to change the language specified by the FND_LOOKUP_VALUES.LANGUAGE = ' ' command.

6.2.2.2 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. Using a text editor, open the domainValues_InvoiceTypes_ora11i.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the TYPE column to the XACT_TYPE_CODE column in the file.

   The data must be copied starting from the second 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.

6.2.2.3 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. Using a text editor open the domainValues_PickTypes_ora11i.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
2. Put 'STANDARD' in the XACT_TYPE_CODE column in the file.

   The data must be copied starting from the second line. The first line is the column header.
3. 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*.
4. Save and close the file.

6.2.2.4 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. Using a text editor, open the domainValues_OrderTypes_ora11i.csv file, located in $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the LOOKUP_TYPE column to the XACT_TYPE_CODE column in the file.

   The data must be copied starting from the second 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.

6.2.2.5 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. Using a text editor, open the domainValues_PickStatus_ora11i.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the LOOKUP_CODE column to the STATUS_CODE column in the file.

   The data must be copied starting from the second 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.

6.2.2.6 How to Configure Pay Method Domain Values

This section explains how to configure Pay Method Status domain values using the domainValues_PayMethodCode_ora.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. Using a text editor, open the domainValues_PayMethodCode_ora.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the LOOKUP_CODE column to the METHOD_CODE column in the file.

   The data must be copied starting from the second 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.

6.2.2.7 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. Using a text editor, open the domainValues_InvoiceStatus_ora11i.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the LOOKUP_CODE column to the STATUS_CODE column in the file.

   The data must be copied starting from the second 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.

6.2.2.8 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. Using a text editor, open the domainValues_OrderOverallStatus_ora11i.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the LOOKUP_CODE column to the STATUS_CODE column in the file.

   The data must be copied starting from the second 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.

6.2.2.9 How 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. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the domainValues_Movement_Types_ora11i.csv file in a text editor.
3. Copy the TRANSACTION_TYPE_NAME to the TRANSACTION_TYPE_NAME column in the file.

   The data must be copied starting from the second 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.

6.2.2.10 How to Configure Invoice Source Type Domain Values

This section explains how to configure Invoice Source Type domain values using the domainValues_InvoiceSrcTypes_ora11i.csv file.

To configure Invoice Source Type domain values:

1. Identify the Invoice Source Types in your Oracle 11i source system by using the following SQL:

   ```
   SELECT DISTINCT DESCRIPTIVE_FLEX_CONTEXT_CODE
   FROM FND_DESCR_FLEX_CONTEXTS
   WHERE DESCRIPTIVE_FLEXFIELD_NAME = 'RA_INTERFACE_LINES' AND APPLICATION_ID=222
   ORDER BY 1;
   ```

2. Using a text editor, open the domainValues_InvoiceSrcType_ora11i.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the TYPE column to the XACT_TYPE_CODE1 column in the file.

   The data must be copied starting from the second line. The first line is the column header.
4. Map each Transaction Source 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.

6.2.2.11 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 Informatica PowerCenter Designer, open the SDE_ORA*Version*_Adaptor.
2. Open the mplt_SA_ORA_ProductTransactionFact mapplet.
3. Double-click the EXP_PROD_XACTS Expression transformation to open the Edit Transformations dialog, and display the Port tab to show the EXT_GR_QTY and EXT_DELIVERY_QTY ports.
4. Edit the quantity types by substituting your desired condition for the prepackaged expression.
5. Click **Apply**.
6. Validate the mapplet, and save your changes to the repository.

6.2.2.12 How to Deploy Objects in Oracle EBS for Exploding the BOM

The Bill of Materials (BOM) functional area enables you to determine the profit margin of the components that comprise the finished goods. BOM enables you to keep up with the most viable vendors in terms of cost and profit, and to keep your sales organization aware of product delivery status, including shortages.

To deploy objects in Oracle EBS for exploding the BOM, ensure that the Oracle EBS source environment meets the minimum patch level for your version, as follows:

- Customers with Oracle EBS version R12 must be at or above patch level 10422612.
- Customers with Oracle EBS version R12.0.x or OPI patch set A must be at or above patch level 10422612:R12.OPI.A.
- Customers with Oracle EBS version R12.1.x or OPI patch set B must be at or above patch level 10422612:R12.OPI.B.
- Customers with Oracle EBS version 11*i* must be at or above patch level 10361119.

Refer to the *System Requirements and Supported Platforms for Oracle Business Intelligence Applications* for full information about supported patch levels for your source system.

For more details, see Section 6.3.2.14, "How to Configure the Bill of Materials Explosion Options."

> **Note:** Systems at or above these minimum patch levels include the package OPI_OBIA_BOMPEXPL_WRAPPER_PKG in the APPS schema, and include the following tables in the OPI schema with alias tables in the APPS schema:
>
> OPI_OBIA_W_BOM_HEADER_DS
>
> OPI_OBIA_BOM_EXPLOSION
>
> OPI_OBIA_BOM_EXPLOSION_S

6.2.3 Configuration Steps for Oracle Supply Chain and Order Management Analytics for Oracle's JD Edwards EnterpiseOne

This section contains configuration steps required before you do a full data load that apply to Oracle's JD Edwards EnterpriseOne. It contains the following topics:

- Section 6.2.3.1, "About Configuring Domain Values and CSV Worksheet Files for JD Edwards EnterpriseOne Supply Chain and Order Management Analytics"
- Section 6.2.3.2, "How to Configure User Defined Code Categories"
- Section 6.2.3.3, "How to Configure Movement Type Domain Values"
- Section 6.2.3.4, "How to Configure Transaction Types Domain Values for Sales Invoice Lines"
- Section 6.2.3.5, "How to Configure Transaction Types Domain Values for Sales Order Lines"
- Section 6.2.3.6, "How to Configure Sales Order Statuses"
- Section 6.2.3.7, "How to Configure Channel Type Groups"
- Section 6.2.3.8, "How to Configure Inventory Consignments"
- Section 6.2.3.9, "How to Configure Inventory Return Locations"
- Section 6.2.3.10, "How to Process Bill of Material Explosion"
- Section 6.2.3.11, "How to Deploy the Stored Procedure for the Left Bound and Right Bound Calculation Option"

6.2.3.1 About Configuring Domain Values and CSV Worksheet Files for JD Edwards EnterpriseOne Supply Chain and Order Management Analytics

Table 6–2 lists the CSV worksheet files and the domain values for Oracle Supply Chain and Order Management Analytics that are located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

Table 6–2 Domain Values and CSV Worksheet Files for Oracle Supply Chain and Order Management Analytics for JD Edwards EnterpriseOne

Worksheet File	Description	Session
file_udc_category_mapping_jde.csv	This file is used to provide JD Edwards EnterpriseOne user defined codes that are to be populated into the W_CODE_D table.	SDE_JDE_Code_Category_Map_Load

Table 6–2 (Cont.) Domain Values and CSV Worksheet Files for Oracle Supply Chain and Order Management Analytics for JD Edwards EnterpriseOne

Worksheet File	Description	Session
domainvalues_Movement_Types_jde.csv	This file provides override information of the Movement Type Description for a specific Movement Type.	SDE_JDE_MvmntTypeDimension
domainvalues_xact_type_codes_scom_jde_sales_ivclns.csv	This file provides the Sales Order Transaction Type Codes that are applicable to sales invoice lines.	SDE_JDE_XactTypeDimension_SalesInvoiceLine
domainvalues_xact_type_codes_scom_jde_sales_ordlns.csv	This file provides the Sales Order Transaction Type Codes that are applicable to Sales Order lines.	SDE_JDE_XactTypeDimension_SalesOrderLine
file_sales_order_status_jde.csv	This file provides Sales Order status information to populate the W_CODE_D (by category) and W_STATUS_D tables.	SDE_JDE_StatusDimension_SalesOrder
file_lkp_chnl_typ_jde.csv	This file serves to provide additional information by Channel Type Code to populate the Channel Type Dimension table.	SDE_JDE_ChannelTypeDimension
file_lkp_consign_inv_org_jde.csv	This file is used to identify the Branch/Plants (Organization ID) that are used for consignment within the inventory process.	SDE_JDE_Inventory_Daily_Bal_Fact
file_lkp_return_loc_jde.csv	This file is used to identify the return locations by Branch/Plant, Location, and Lot Number within the inventory process.	SDE_JDE_Inventory_Daily_Bal_Fact

6.2.3.2 How to Configure User Defined Code Categories

This section explains how to configure the file_udc_category_mapping_jde.csv file.

You populate this file with data before running the SDE_JDE_Category_Map_Load workflow. This workflow will run a mapping of the same name that uses the CSV file as input to create a temporary table called W_JDE_UDC_CATEGORY_MAP_TMP. This temporary table is then used as source input into the SDE_JDE_CODE_DIMENSION workflow to load the W_CODE_D table.

To configure file_udc_category_mapping_jde.csv:

1. Using a text editor, open the file_udc_category_mapping_jde.csv file, located in the $PMServer\SrcFiles folder.
2. Compare the values in this file to the values in Table 3–22 in Section 3.5.1.2, "About Configuring the Code Dimension for Oracle's JD Edwards EnterpriseOne or JD Edwards World UDCs."
3. Add any new values to the CSV file from the Table 3–22 entries by System Code, User Defined Code, and Category. Ensure that each Category value is spelled correctly, as this Category matches what is defined in the mappings for Oracle Supply Chain and Order Management Analytics for JD Edwards EnterpriseOne.

   The values that are added to this CSV file will be added as UDC entries to the W_CODE_D table by Category.

   For more information about the W_CODE_D table and values, see *Oracle Business Analytics Warehouse Data Model Reference*.

4. Use commas to separate the entries under the System Code, UDC column, and Category. Ensure that you do not include any spaces.
5. Save and close the file.

6.2.3.3 How to Configure Movement Type Domain Values

This section explains how to configure the domainvalues_Movement_Types_jde.csv. You populate this flat file with data before running the SDE_JDE_MvmntTypeDimension workflow.

To configure domainvalues_Movement_Types_jde.csv:

1. Identify the Movement Types in your JD Edwards EnterpriseOne source system by using the following SQL:

```
SELECT RTRIM(DRSY) || '~' || DRRT || '~' ||
LTRIM(DRKY) FROM F0005 AS MVMNT_TYPE_CODE
WHERE DRSY = '00' AND DRRT = 'DT' AND DRKY <> ' '
```

2. Using a text editor, open the domainvalues_Movement_Types_jde.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the values from the MVMNT_TYPE_CODE column of the SQL to the MVMNT_TYPE_CODE column in the file.

   The data must be copied starting from the second line. The first line is the column header.
4. In the file, for each MVMNT_TYPE_CODE, add the correct domain values for W_MVMNT_TYPE_CODE and W_MVMNT_TYPE_DESC. Valid domain values are OTHERS, CUSTOMER RETURN, or VENDOR RETURN.

   All values for both W_MVMNT_TYPE_CODE and W_MVMNT_TYPE_DESC will be OTHERS except for those Document Types (Movement Types) that are actually used for Customer Returns or Vendor Returns. For those specific Document Types identified as such, use those values appropriately.

   For more information on Movement Type domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

6.2.3.4 How to Configure Transaction Types Domain Values for Sales Invoice Lines

This section explains how to configure the domainvalues_xact_type_codes_scom_jde_sales_ivclns.csv. You populate this flat file with data before running the SDE_JDE_XactTypeDimension_SalesInvoiceLine workflow.

To configure domainvalues_xact_type_codes_scom_jde_sales_ivclns.csv:

1. Using a text editor, open the domainvalues_xact_type_codes_scom_jde_sales_ivclns.csv file, located in the $PMServer\LkpFiles folder.
2. Identify the Invoice Line transaction types in JD Edwards EnterpriseOne by using the following SQL:

```
SELECT F0005.DRKY FROM F0005
WHERE DRSY = '00' AND DRRT = 'DT'
```

3. Map the values returned with the flat file to correspond with the XACT_TYPE_CODE source column. Associate the values in this file to the Transaction Type

domain values for Invoice Lines as specified in the *Oracle Business Analytics Warehouse Data Model Reference*.

An example configuration is shown here:

```
XACT_TYPE_CODE,XACT_CODE,W_XACT_TYPE_CODE,W_XACT_TYPE_DESC,W_XACT_TYPE_CODE1,W_
XACT_TYPE_DESC1
<DRKY>,SALES_IVCLNS,Standard Invoice,Standard Invoice,Order Management,Order
Management source code
```

> **Note:** The values shown in the example are also the default values for the source code's domain values. If a Document Type <DRKY> is not configured in the flat file, the values presented in this example will be assumed by default for those documents.

For more information about the Transaction Type domain values in W_XACT_TYPE_D, see *Oracle Business Analytics Warehouse Data Model Reference*.

4. Use commas to separate the entries under the System Code, UDC column, and Category. Ensure that you do not include any spaces.
5. Save and close the file.

6.2.3.5 How to Configure Transaction Types Domain Values for Sales Order Lines

This section explains how to configure the domainvalues_xact_type_codes_scom_jde_sales_ordlns.csv. You populate this flat file with data before running the SDE_JDE_XactTypeDimension_SalesOrderLine workflow.

To configure domainvalues_xact_type_codes_scom_jde_sales_ordlns.csv:

1. Using a text editor, open the domainvalues_xact_type_codes_scom_jde_sales_ordlns.csv file, located in the $PMServer\LkpFiles folder.
2. Identify the Order Line transaction types in JD Edwards EnterpriseOne by using the following SQL:

```
SELECT F0005.DRKY FROM F0005
WHERE DRSY = '00' AND DRRT = 'DT'
```

3. Map the values returned with the flat file to correspond with the XACT_TYPE_CODE source column. Associate the values in this column to the Transaction Type domain values for Order lines as specified in the *Oracle Business Analytics Warehouse Data Model Reference*.

An example configuration is shown here:

```
XACT_TYPE_CODE,XACT_CODE,W_XACT_TYPE_CODE,W_XACT_TYPE_DESC,W_XACT_TYPE_CODE1,W_
XACT_TYPE_DESC1, W_XACT_TYPE_CODE2,W_XACT_TYPE_DESC2
<DRKY>,SALES_ORDLNS,Regular,Regular,EXTERNAL,External order,Self Ship,Internal
inventory order
```

> **Note:** The values shown in the example are also the default values for the source code's domain values. If a Document Type <DRKY> is not configured in the flat file, the values presented in this example will be assumed by default for those documents.

For more information about the W_XACT_TYPE_D domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

4. Use commas to separate the entries under the System Code, UDC column, and Category. Ensure that you do not include any spaces.
5. Save and close the file.

6.2.3.6 How to Configure Sales Order Statuses

This section explains how to configure the file_sales_order_status_jde.csv. You populate this flat file with data before running the SDE_JDE_StatusDimension_SalesOrder workflow. This flat file is the source for the table W_STATUS_DS and is available in the $PMServer\SrcFiles folder.

To configure file_sales_order_status_jde.csv:

1. Using a text editor, open the file_sales_order_status_jde.csv file, located in the $PMServer\SrcFiles folder.
2. Identify the Sales Order status combinations by analyzing your order activity rules setup, which will have data similar to this:

DocType	LineType	Last Status	Description	Next Status	Description
SO	S	520	Enter Sales Order	540	Print Pickslips
SO	S	520	Enter Sales Order	560	Shipment Confirmation
SO	S	520	Enter Sales Order	535	In Warehouse Management
SO	S	900	Backordered in SO Entry	540	Print Pickslips

3. Identify the distinct combination of DocType, LineType, Last Status, and Next Status from the F4211 table for all of your sales orders.
4. Render each of these combinations with a prefix of:

```
SALES_ORDLNS~<DocType>~<LineType>~<LastStatus>~NextStatus>
SALES_PCKLNS~<DocType>~<LineType>~<LastStatus>~NextStatus>
SALES_SCHLNS~<DocType>~<LineType>~<LastStatus>~NextStatus>
```

5. Copy the values rendered into the flat file to correspond with the STATUS_CODE, STATUS_NAME flat file column for each of the STATUS_CLASSes - SALES_ORDLNS, SALES_PCKLNS, and SALES_SCHLNS respectively.

   An example configuration is shown here:

```
STATUS_CLASS,STATUS_CODE,STATUS_NAME,STATUS_CAT,STATUS_CAT_DESC
SALES_ORDER_PROCESS,SALES_ORDLNS~SO~S~620~998,SALES_ORDLNS~SO~S~620~998,Being
Processed,Being Processed
SALES_ORDER_PROCESS,SALES_ORDLNS~SO~S~620~999,SALES_
ORDLNS~SO~S~620~999,Closed,Closed
SALES_ORDER_PROCESS,SALES_ORDLNS~SO~S~985~999,SALES_
ORDLNS~SO~S~985~999,Closed,Closed
SALES_PICKING_PROCESS,SALES_PCKLNS~SO~S~520~540,SALES_PCKLNS~SO~S~520~540,Not
Yet Picked,Not Yet Picked
SALES_PICKING_PROCESS,SALES_PCKLNS~SO~S~520~560,SALES_PCKLNS~SO~S~520~560,Not
Yet Picked,Not Yet Picked
```

Note: ■The code does not automatically identify statuses as ENTERED, BOOKED, BLOCKED, and so forth. These statuses are user-configured in OLTP. These statuses are derived from the setup in the Status Dimension flat file. For example, back-ordered records may be treated as Blocked. The code does not automatically handle this, therefore you should configure the status dimension flat file accordingly to treat back-ordered status combinations as BLOCKED.

- A set of hard-coded statuses are used to identify whether a Sales Order line is cancelled or not. These statuses are 980, 982, 983, 984, 985, 986, 987, 988, 990, 995, 996 and 997. No other Sales Order status is used to identify a cancelled Sales Order line. If you want a certain status combination to be treated as closed, then define it as closed in the Status Dimension flat file. When defined as closed, they will be overlooked by financial and operational backlog flags.
- You do not have to configure certain statuses that are already hard-coded in the system like Closed, Cancelled, and so forth, as mentioned previously. The code automatically identifies them. However, a hard-coded status line should be present in the flat file to track these statuses, which are described in the following steps.

6. Append your configuration with these mandatory hard coded statuses:

```
FULFILLMENT_STATUS,ORDER BOOKED,ORDER BOOKED,Order Booked,Order Booked
FULFILLMENT_STATUS,ORDER ENTERED,ORDER ENTERED,Order Entered,Order Entered
FULFILLMENT_STATUS,ORDER FULLY CANCELLED,ORDER FULLY CANCELLED,Order Fully
 Cancelled,Order Fully Cancelled
FULFILLMENT_STATUS,ORDER FULLY PICKED,ORDER FULLY PICKED,Order Fully
 Picked,Order Fully Picked
FULFILLMENT_STATUS,ORDER FULLY SCHEDULED,ORDER FULLY SCHEDULED,Order Fully
 Scheduled,Order Fully Scheduled
FULFILLMENT_STATUS,ORDER FULLY SHIPPED,ORDER FULLY SHIPPED,Order Fully
 Shipped,Order Fully Shipped
FULFILLMENT_STATUS,ORDER PARTIALLY CANCELLED,ORDER PARTIALLY CANCELLED,Order
 Partially Cancelled,Order Partially Cancelled
FULFILLMENT_STATUS,ORDER PARTIALLY SCHEDULED,ORDER PARTIALLY SCHEDULED,Order
 Partially Scheduled,Order Partially Scheduled
FULFILLMENT_STATUS,ORDER PARTIALLY SHIPPED,ORDER PARTIALLY SHIPPED,Order
 Partially Shipped,Order Partially Shipped
FULFILLMENT_STATUS,ORDER NOT ELIGIBLE,ORDER NOT ELIGIBLE,Order Not
 Eligible,Order Not Eligible
SALES_INVOICE_PROCESS,SALES_INVCLNS~CANCELLED,SALES_INVCLNS~CANCELLED,
 Cancelled,Cancelled
SALES_INVOICE_PROCESS,SALES_INVCLNS~COMPLETED,SALES_INVCLNS~COMPLETED,
 Completed,Completed
SALES_INVOICE_PROCESS,SALES_INVCLNS~OPEN,SALES_INVCLNS~OPEN,Open,Open
SALES_INVOICE_PROCESS,SALES_INVCLNS~PENDING,SALES_INVCLNS~PENDING,Pending,
 Pending
SALES_ORDER_PROCESS,SALES_ORDLNS~BEING PROCESSED,SALES_ORDLNS~BEING PROCESSED,
 Being Processed,Being Processed
SALES_ORDER_PROCESS,SALES_ORDLNS~BLOCKED,SALES_ORDLNS~BLOCKED,Blocked,Blocked
SALES_ORDER_PROCESS,SALES_ORDLNS~BOOKED,SALES_ORDLNS~BOOKED,Booked,Booked
SALES_ORDER_PROCESS,SALES_ORDLNS~CANCELLED,SALES_ORDLNS~CANCELLED,Cancelled,
 Cancelled
```

```
SALES_ORDER_PROCESS,SALES_ORDLNS~CLOSED,SALES_ORDLNS~CLOSED,Closed,Closed
SALES_ORDER_PROCESS,SALES_ORDLNS~ENTERED,SALES_ORDLNS~ENTERED,Entered,Entered
SALES_PICKING_PROCESS,SALES_PCKLNS~CANCELLED,SALES_
PCKLNS~CANCELLED,Cancelled,Cancelled
SALES_PICKING_PROCESS,SALES_PCKLNS~FULLY PICKED,SALES_PCKLNS~FULLY PICKED,Fully
 Picked,Fully Picked
SALES_PICKING_PROCESS,SALES_PCKLNS~FULLY SHIPPED,SALES_PCKLNS~FULLY SHIPPED,
 Fully Shipped,Fully Shipped
SALES_PICKING_PROCESS,SALES_PCKLNS~NOT RELEVANT,SALES_PCKLNS~NOT RELEVANT,Not
 Relevant,Not Relevant
SALES_PICKING_PROCESS,SALES_PCKLNS~NOT YET PICKED,SALES_PCKLNS~NOT YET PICKED,
 Not Yet Picked,Not Yet Picked
SALES_PICKING_PROCESS,SALES_PCKLNS~BACKORDERED,SALES_PCKLNS~BACKORDERED,
 Backordered,Backordered
SALES_PICKING_PROCESS,SALES_PCKLNS~PURGED,SALES_PCKLNS~PURGED,Purged,Purged
SALES_SCHEDULE_PROCESS,SALES_SCHLNS~BLOCKED,SALES_SCHLNS~BLOCKED,Blocked,
 Blocked
SALES_SCHEDULE_PROCESS,SALES_SCHLNS~CANCELLED,SALES_SCHLNS~CANCELLED,Cancelled,
 Cancelled
SALES_SCHEDULE_PROCESS,SALES_SCHLNS~CLOSED,SALES_SCHLNS~CLOSED,Closed,Closed
SALES_SCHEDULE_PROCESS,SALES_SCHLNS~ENTERED,SALES_SCHLNS~ENTERED,Entered,
 Entered
SALES_SCHEDULE_PROCESS,SALES_SCHLNS~NOT VALID,SALES_SCHLNS~NOT VALID,Not
 Valid,Not Valid
```

7. Save and close the file.

6.2.3.7 How to Configure Channel Type Groups

This section explains how to configure the file_lkp_chnl_typ_jde.csv. You populate this flat file with data before running the SDE_JDE_ChannelTypeDimension workflow.

To configure the file_lkp_chnl_typ_jde.csv file:

1. Identify the Channel Types in your JD Edwards EnterpriseOne source system by using the following SQL:

   ```
   SELECT DRKY FROM F0005 WHERE DRSY = '90CB' AND DRRT = 'TC'
   ```

2. Using a text editor, open the file_lkp_chnl_typ_jde.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

3. Copy the values from the DRKY column of the SQL to the CHNL_TYPE_CODE column in the file.

   The data must be copied starting from the second line. The first line is the column header.

4. In the file, add the correct domain values for Channel Type Group Code (W_CHTY_GRP_CODE), Channel Type Subgroup Code (W_CHTY_SUBG_CODE), and W_INBOUND_TYPE_FLG for each Channel Type Code (CHNL_TYPE_CODE).

   For more information on Channel Type domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

5. Save and close the file.

6.2.3.8 How to Configure Inventory Consignments

This section explains how to configure the file_lkp_consign_inv_org_jde.csv. You populate this flat file with data before running the SDE_JDE_Inventory_Daily_Bal_Fact workflow.

To configure file_lkp_consign_inv_org_jde.csv:

1. Identify the Inventory Consignment Branch/Plants (Warehouse and Inventory Organization IDs) in your JD Edwards EnterpriseOne source system.
2. Using a text editor, open the file_lkp_consign_inv_org_jde.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Manually enter the first value for a valid Inventory Consignment Branch/Plant on the second line in the file. The first line is the column header.

   > **Note:** The data for each Branch/Plant must be entered as 12 characters and right-justified with leading blanks. For example, Branch/Plant (INVENTORY_ORG_ID) ABC should be entered as ---------ABC with nine leading blanks.

4. Enter each Inventory Consignment Branch/Plant on a separate line thereafter.

   For more information on consignments, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

6.2.3.9 How to Configure Inventory Return Locations

This section explains how to configure the file_lkp_return_loc_jde.csv. You populate this flat file with data before running the SDE_JDE_Inventory_Daily_Bal_Fact workflow.

To configure file_lkp_return_loc_jde.csv:

1. Identify the Inventory Return Locations in each Branch/Plant (Warehouse/Inventory Organization ID) in your JD Edwards EnterpriseOne source system.
2. Using a text editor, open the file_lkp_return_loc_jde.csv file, located in the $PMServer\LkpFiles directory (for example *INFA_HOME*\server\infa_shared\LkpFiles).
3. Manually enter the first value for a valid Inventory Return Location on the second line in the file. The first line is the column header.

Note: The data for each Return Location must be entered in the following format on each line:

STORAGE_LOC~Branch/Plant~Location~Lot Number

where:

- STORAGE_LOC~ is a constant
- Branch/Plant value is 12 characters and right-justified, blank filled
- Location is left-justified and 20 characters right-blank filled
- Lot Number is entered with no trailing blanks

For example, an Inventory Return Location with a Branch/Plant of ABC, a Location of 123 and a Lot Number of 789, would be entered in the following format:

```
STORAGE_LOC~         ABC~123                 ~789
```

4. Enter each Inventory Return Location on a separate line thereafter.

   For more information on return locations, see *Oracle Business Analytics Warehouse Data Model Reference.*

5. Save and close the file.

6.2.3.10 How to Process Bill of Material Explosion

This section explains how to process the Bill of Materials (BOM) for exploding to a multi-level structure to ultimately populate both the W_BOM_HEADER_D and W_BOM_ITEM_F tables.

JD Edwards EnterpriseOne maintains BOM information in a single level format, but Oracle BI Applications requires it in multi-level format. Therefore, before loading data into Oracle BI Application tables, the single level structure must be exploded into a multi-level structure.

Because all of the BOM information is stored in one single table in JD Edwards EnterpriseOne source and there are no defined levels for the BOM, the system has to loop through iteratively to get the BOM exploded. Also, Oracle BI Applications maintains all the revisions to the components as a new version of the BOM along with their effective dates. Considering these facts, it is not feasible to use ETL to convert the single level BOM to a multi-level BOM. Therefore the logic from an existing JD Edwards EnterpriseOne UBE (R30460) was used to create a new UBE for the explosion.

This new UBE (R30461) extracts the manufactured end products and converts the single-level BOM format into a multi-level BOM format. In addition, it also extracts some required information like Left bounds/Right bounds and level parents (1 - 10).

The UBE loads the multi-level BOM structure for manufactured end products with each revision into two work files respectively for BOM header and item (component). The ETL then extracts the data from the two work files and loads it into the Oracle BI Applications tables.

> **Note:** If you plan to consume analytics on Bill of Materials, it is mandatory to run this UBE before starting the ETL. This UBE and the related JD Edwards EnterpriseOne objects are created solely for the benefit of analytics and therefore will not be available in the existing source system.

6.2.3.11 How to Deploy the Stored Procedure for the Left Bound and Right Bound Calculation Option

The SIL_BOMItemFact mapping contains the stored procedure called `COMPUTE_BOUNDS` which traverses the exploded BOM tree structure and calculates the left bound and right bound. By default, the `COMPUTE_BOUNDS` stored procedure is off. If you want to turn on the procedure, see Section 6.3.2.15, "How to Configure the Left Bound and Right Bound Calculation Option."

Before you run an ETL using a BOM, you must compile and deploy the SQL code in Compute_Bounds_Ora11i.sql. You deploy stored procedures by copying the stored procedure files from your Oracle Business Intelligence installation and deploying them to the target data warehouse.

In JD Edwards EnterpriseOne, the left and right bounds are calculated by the UBE (R30461).

> **Note:** Some sessions may fail if this procedure is not compiled in the database before running the workflows.

To deploy the COMPUTE_BOUNDS stored procedure:

1. Navigate to the *MW_HOME*\biapps\dwrep\Informatica\Stored_Procedure_Scripts folder.
2. Select the appropriate folder for your target database type (for example, Oracle or DB2).
3. Copy the source code in the *DB_type*\Compute_Bounds_*DB_type*.sql file into the target data warehouse schema.

   For example, for an Oracle database, copy the source SQL code from Oracle\Compute_Bounds_Ora11i.sql.
4. Compile the stored procedures in the target data warehouse database.

> **Note:** If you have problems deploying stored procedures, see your database reference guide, or contact your database administrator.

6.3 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle Supply Chain and Order Management Analytics. It contains the following topics:

- Section 6.3.1, "Configuration Steps for Oracle Supply Chain and Order Management Analytics for All Source Systems"
- Section 6.3.2, "Configuration Steps for Oracle Supply Chain and Order Management Analytics for Oracle EBS"

6.3.1 Configuration Steps for Oracle Supply Chain and Order Management Analytics for All Source Systems

This section contains configuration steps that apply to all source systems. It contains the following topics:

- Section 6.3.1.1, "Tracking Multiple Attribute Changes in Bookings"
- Section 6.3.1.2, "Process of Aggregating Oracle Supply Chain and Order Management Analytics Tables"
- Section 6.3.1.3, "About Tracking Multiple Products for Oracle Supply Chain and Order Management Analytics"
- Section 6.3.1.4, "Adding Dates to the Order Cycle Time Table for Post-Load Processing"
- Section 6.3.1.5, "About Configuring the Backlog Period Date for Oracle Supply Chain and Order Management Analytics"
- Section 6.3.1.6, "Example of How Backlog Data Is Stored in the Backlog History Table"
- Section 6.3.1.7, "About Configuring the Customer Status History Fact for Post-Load Processing In Oracle Supply Chain and Order Management Analytics"
- Section 6.3.1.8, "Configuring the Customer Status History Fact table"
- Section 6.3.1.9, "How to Configure the Customer Status History Fact Table"
- Section 6.3.1.10, "About Configuring the Inventory Monthly Balance Table"
- Section 6.3.1.11, "How to Configure the Inventory Monthly Balance"
- Section 6.3.1.12, "About Configuring the Product Transaction Aggregate Table"
- Section 6.3.1.13, "How to Configure the Product Transaction Aggregate Table"

6.3.1.1 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 Account, 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_ACCOUNT_ID)
```

To track dimensional attribute changes in bookings:

1. In Informatica PowerCenter Designer, open the SDE_ORA115*Version*_Adaptor or SDE_ORAR12_Adaptor folder.
2. Open one of the following mappings:
   - mplt_SA_ORA_SalesOrderLinesFact
   - mplt_SA_ORA_SalesScheduleLinesFact

3. Double-click the appropriate Expression transformation to open the Edit Transformation box:

   - EXP_SALES_ORDLNS
   - EXP_SALES_SCHLNS

4. In the Ports tab, edit the expression for the EXT_BOOKING_ID port, 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 all attributes and put the concatenated value in the VAR_BOOKING_ID column.

5. Validate and save your changes to the repository.

6.3.1.2 Process of Aggregating Oracle Supply Chain and Order Management Analytics Tables

This section contains Oracle Supply Chain and Order Management Analytics configuration points for aggregating the Sales Invoice Lines and Sales Order Lines tables.

The aggregation processes use the following Teradata parameters:

- Hint_Tera_Pre_Cast
- Hit_Tera_Post_Cast

To aggregate the Sales Invoice Lines and Sales Order Lines tables, perform the following tasks:

- Configure the Sales Invoice Lines Aggregate Table
- Configure the Sales Order Lines Aggregate Table

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 need to configure the Sales Invoice Lines aggregate table in order to run initial ETL and incremental ETL.

For your initial ETL run, you need to configure the TIME_GRAIN parameter for the time aggregation level in the Sales Invoice Lines aggregate fact table.

For the incremental ETL run, you need to configure the time aggregation level.

To aggregate the Sales Invoice Lines table for your incremental run, you need to configure the TIME_GRAIN parameter.

The TIME_GRAIN parameter has a preconfigured value of Month. The possible values for the TIME_GRAIN parameter are:

- 'WEEK'
- 'MONTH'
- 'QUARTER'
- 'YEAR'

The aggregation processes use the following Teradata parameters:

- Hint_Tera_Pre_Cast
- Hit_Tera_Post_Cast

The Sales Invoice Lines aggregate table is fully loaded from the base table in the initial ETL 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 ETL 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 ETL 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 ETL 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 ETL 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).

How to Configure the Sales Invoice Lines Aggregate Table

To load the Sales Invoice Lines aggregate table (W_SALES_INVOICE_LINE_A), you need to configure the parameterfileDW.txt file, and run the initial and then the incremental workflows. Using a text editor, open the file parameterfileDW.txt, located in the Dac\Informatica\parameters\input folder on the DAC Server computer. For example:

C:\orahome\10g*version*\bifoundation\dac\Informatica\parameters\input

To configure the Sales Invoice Lines Aggregate Table:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab.
3. For each of the following tasks, display the Parameters tab and specify an appropriate value in the Value field for the TIME_GRAIN parameter:
   - SIL_SalesInvoiceLinesAggregate_Derive_PreLoadImage
   - SIL_SalesInvoiceLinesAggregate_Derive_PreSoftDeleteImage
   - PLP_SalesInvoiceLinesAggregate_Derive_PostLoadImage

- PLP_SalesInvoiceLinesAggregate_Load

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 order lines issued for your sales orders. You need to configure the Sales Order Lines aggregate table in order to run initial ETL and incremental ETL.

For your initial ETL run, you need to configure the TIME_GRAIN parameter for the time aggregation level in the Sales Order Lines aggregate fact table.

For the incremental ETL run, you need to configure the time aggregation level.

To aggregate the Sales Invoice Lines table for your incremental run, you need to configure the TIME_GRAIN parameter.

The TIME_GRAIN parameter has a preconfigured value of Month. The possible values for the GRAIN parameter are:

- 'WEEK'
- 'MONTH'
- 'QUARTER'
- 'YEAR'

The aggregation processes use the following Teradata parameters:

- Hint_Tera_Pre_Cast
- Hit_Tera_Post_Cast

The Sales Order Lines aggregate table is fully loaded from the base table in the initial ETL 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 ETL 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 ETL 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 ETL 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 ETL 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).

How to Configure the Sales Order Lines Aggregate Table

To load the Sales Order Lines aggregate table (W_SALES_ORDER_LINE_A), you need to configure the post-load-processing parameter file and the source system parameter files, and run the initial and then the incremental workflows.

To configure the Sales Order Lines Aggregate Table:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab.
3. For each of the following tasks, display the Parameters tab and specify an appropriate value in the Value field for the TIME_GRAIN parameter:
   - SIL_SalesOrderLinesAggregate_Derive_PreLoadImage
   - SIL_SalesOrderLinesAggregate_Derive_PreSoftDeleteImage
   - PLP_SalesOrderLinesAggregate_Derive_PostLoadImage
   - PLP_SalesOrderLinesAggregate_Load

6.3.1.3 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 6–3. 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 6–3 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 6–3 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 6–4.

Table 6–4 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

6.3.1.4 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 Informatica PowerCenter Designer, open the Configuration for Post Load Processing folder (that is, the PLP folder).
2. In Warehouse Designer, 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 need to create a new column, VALIDATED_ON_DT, and modify the target definition of the W_SALES_CYCLE_LINE_F table.
3. In Source Analyzer, modify the table definition of the source table to include this new column.

   Continuing with the example, you would include the VALIDATED_ON_DT column in the W_SALES_CYCLE_LINE_F source table.
4. In Mapping Designer, modify the PLP_SalesCycleLinesFact_Load and PLP_SalesCycleLinesFact_Load_Full mappings to select the new column from any of the following source tables, and load it to the W_SALES_CYCLE_LINE_F target table:
   - W_SALES_ORDER_LINE_F
   - W_SALES_INVOICE_LINE_F
   - W_SALES_PICK_LINE_F
   - W_SALES_SCHEDULE_LINE_F
5. Modify the Source Qualifier SQL Override for the mapping, and map the column in the transformation to map it to the target table.

6.3.1.5 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.

6.3.1.6 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 6–5.

Table 6–5 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 6–6.

Table 6–6 Oracle 11i and Oracle R12: Backlog History Table Entry as of February 2, 2001

SALES_ORDER_NUM	BACKLOG _DK	BACKLOG_PERIOD_ DK	OPEN_QTY
(Sales Order Number)	(Backlog Date)	(Backlog Period Date)	(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 6–7.

Table 6–7 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 6–8.

Table 6–8 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
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 6–8, 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 6–9 provides a list of all applicable mappings and their corresponding Expression transformations that you must modify.

Table 6–9 Oracle 11i and Oracle R12: Backlog History Applicable Mappings and Expression Transformations

Mapping	Expression Transformation
PLP_SalesBacklogLinesfact_ LoadOrderLines	EXP_SALES_ORNLNS_BACKLOG

Table 6–9 (Cont.) Oracle 11i and Oracle R12: Backlog History Applicable Mappings and Expression Transformations

Mapping	Expression Transformation
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).

6.3.1.7 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.

6.3.1.8 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 BI 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 Informatica 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, and so on.

For information about how to configure the $$WH_DATASOURCE_NUM_ID value, see:Section 6.3.1.9, "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 BI 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 Informatica 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 BI 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 value, see:Section 6.3.1.9, "How to Configure the Customer Status History Fact Table."

6.3.1.9 How to Configure the Customer Status History Fact Table

This section explains the steps involved in configuring the Customer Status History Fact table using the $$WH_DATASOURCE_NUM_ID and $$PERIOD variables (for more information about these variables, see Section 6.3.1.8, "Configuring the Customer Status History Fact table").

To modify the value of $$WH_DATASOURCE_NUM_ID:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab and locate the parameter $$WH_DATASOURCE_NUM_ID.
3. In the Edit subtab, enter an appropriate value in the Value field.
4. Save the changes.

To modify the value of $$PERIOD:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab and query for the following two tasks:
   - PLP_CustomerStatusHistoryFact_New_Customers_Load
   - PLP_CustomerStatusHistoryFact_Status_Revalidate
3. For each task, display the Parameters subtab, and enter an appropriate value in the Value field.

   Make sure you the set same value for both tasks.
4. Save the changes.

6.3.1.10 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**

  The GRAIN parameter has a preconfigured value of Month. The possible values for the GRAIN parameter are:

  - DAY
  - WEEK
  - MONTH
  - QUARTER
  - YEAR

- **KEEP_PERIOD**

  The KEEP_PERIOD parameter has a preconfigured value of Month. Values for the KEEP_PERIOD parameter include:

  - DAY
  - WEEK
  - MONTH
  - QUARTER
  - YEAR

- **NUM_OF_PERIOD**

  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.

> **Note:** If you need "Inventory Turns" data for a period older than 3 months, you must change the parameter values for KEEP_PERIOD and NUM_OF_PERIOD. For example, if you want data for the last 3 years, then set KEEP_PERIOD to YEAR and NUM_OF_PERIOD to 3.

6.3.1.11 How to Configure the Inventory Monthly Balance

Before you run the initial ETL session or incremental ETL sessions to load the Inventory Monthly Balance table, configure the Inventory Monthly Balance as follows.

To configure the Inventory Monthly Balance:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab.
3. For each of the following tasks, display the Parameters subtab and create the specified parameter name and parameter value

   - PLP_InventoryMonthlyBalance $$GRAIN 'MONTH'
   - PLP_InventoryDailyBalance_Trim $$KEEP_PERIOD 'MONTH'
   - PLP_InventoryDailyBalance_Trim $$NUM_OF_PERIOD 3

To incrementally refresh the Inventory Monthly Balance table:

1. Delete the records from the Monthly Balance (W_INVENTORY_MONTHLY_BAL_F) aggregate table for a certain time.

   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.

   Running the PLP_InventoryMonthlyBalance workflow mapping implements this step.

2. 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 a certain grain level.

   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 the Monthly 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.

   Running the PLP_InventoryMonthlyBalance workflow implements this step.

3. Remove 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.

   Running the PLP_InventoryDailyBalance_Trim workflow implements this step.

> **Note:** The trimming process is to reduce data size in the table. It is important to emphasize that you will not be able to see the old daily balance records. But, you will still be able to see the month-end balance. Therefore, ensure that you adjust the NUM_OF_PERIOD values to reflect your data volume and data recency requirement.

6.3.1.12 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 ETL run and then the incremental ETL run.

For your initial ETL run, you need to configure the aggregation level, and the length of history kept in the Product Transaction fact table.

For your initial ETL run, you need to configure the aggregation grain, using the GRAIN parameter.

For the incremental ETL 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).

6.3.1.13 How to Configure the Product Transaction Aggregate Table

Before you run the initial ETL and then the incremental ETL 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. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab.
3. Locate the Task named PLP_ProductTransactionAggregate, display the Parameters subtab, and ensure that the following three parameters are set as specified:

   - $$REFRESH_PERIOD = 'MONTH'
   - $$GRAIN = 'MONTH'
   - $$NUM_OF_PERIOD = 3

   Note: If any of these parameters do not exist, create them as Data Type = Text with the specified Values.

To configure the Product Transaction aggregate table for the initial ETL run:

1. 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 at a certain grain level.

   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.

   Running the PLP_ProductTransactionAggregatel workflow implements this step.

To configure the Product Transaction aggregate table for the incremental ETL run:

1. Delete the refreshed records from the Product Transaction aggregate (W_PRODUCT_XACT_A) table for a certain time.

   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.

Running the PLP_ProductTransactionAggregate workflow implements this step.

2. 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 a certain grain level.

   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.

   Running the PLP_ProductTransactionAggregate workflow implements this step.

6.3.2 Configuration Steps for Oracle Supply Chain and Order Management Analytics for Oracle EBS

This section contains configuration steps that apply to Oracle EBS. It contains the following topics:

- Section 6.3.2.1, "About Configuring Sales Order Lines Data Storage"
- Section 6.3.2.2, "How to Include Non-Booked Orders in the Sales Order Lines and Sales Schedule Lines Tables"
- Section 6.3.2.3, "How to Configure Sales Schedule Lines Data Storage"
- Section 6.3.2.4, "About Loading Bookings at the Schedule Line Level"
- Section 6.3.2.5, "How to Configure Early and Late Tolerances for Shipments"
- Section 6.3.2.6, "How to Configure Sales Invoice Lines Data Storage"
- Section 6.3.2.7, "How to Configure the Sales Invoice Extract"
- Section 6.3.2.8, "How to Configure the Order Lines"
- Section 6.3.2.9, "How to Configure Oracle Financial Analytics for Oracle Supply Chain and Order Management Analytics"
- Section 6.3.2.10, "About Tracking Attribute Changes in Bookings"
- Section 6.3.2.11, "About Viewing the Data Warehouse Changes by Salesperson ID"
- Section 6.3.2.12, "How to Configure Different Types of Backlog Calculations"
- Section 6.3.2.13, "Adding Closed Orders to Backlog Calculations"
- Section 6.3.2.14, "How to Configure the Bill of Materials Explosion Options"
- Section 6.3.2.15, "How to Configure the Left Bound and Right Bound Calculation Option"

6.3.2.1 About Configuring 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.

6.3.2.2 How to Include Non-Booked Orders in the Sales Order Lines and Sales Schedule Lines Tables

By default, only booked orders are extracted from the Oracle source system as shown in Figure 6–1. Therefore, all orders loaded into the Sales Order Lines, Sales Schedule Lines, and Sales Booking Lines tables are booked.

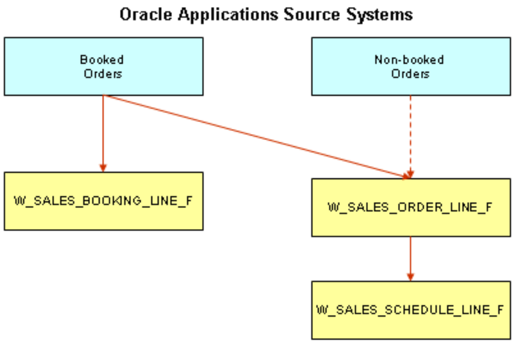
However, if you want to load non-booked orders into the Sales Order Lines and Sales Schedule Lines tables, you have to configure the extract so that it does not filter out non-booked orders. In Oracle 11i and Oracle R12, the `OE_ORDER_LINES_ALL.BOOKED_FLAG = 'Y'` condition indicates that an order is booked; therefore, this statement is used to filter out non-booked orders. To load all orders, including non-booked orders, remove the filter condition from the Source SQL query in the following objects:

- SDE_ORA_SalesOrderLinesFact (mapping)
- SDE_ORA_SalesOrderLinesFact_Primary (mapping)
- SDE_ORA_SalesOrderLinesFact_Full (session)

If you include non-booked orders in the Sales Order Lines and Sales Schedule Lines tables, you have to exclude non-booked orders when you populate the Sales Booking Lines table from the Sales Order Lines or from the Sales Schedule Lines. You can do this by adding the `W_SALES_ORDER_LINE_F.BOOKING_FLG = 'Y'` condition to the Source SQL query in the following mappings:

- SIL_SalesBookingLinesFact_Load_OrderLine_Credit, SIL_SalesBookingLinesFact_Load_OrderLine_Debit
- SIL_SalesBookingLinesFact_Load_ScheduleLine_Credit, SIL_SalesBookingLinesFact_Load_ScheduleLine_Debit

Figure 6–1 Handling Booked and Non-Booked Orders

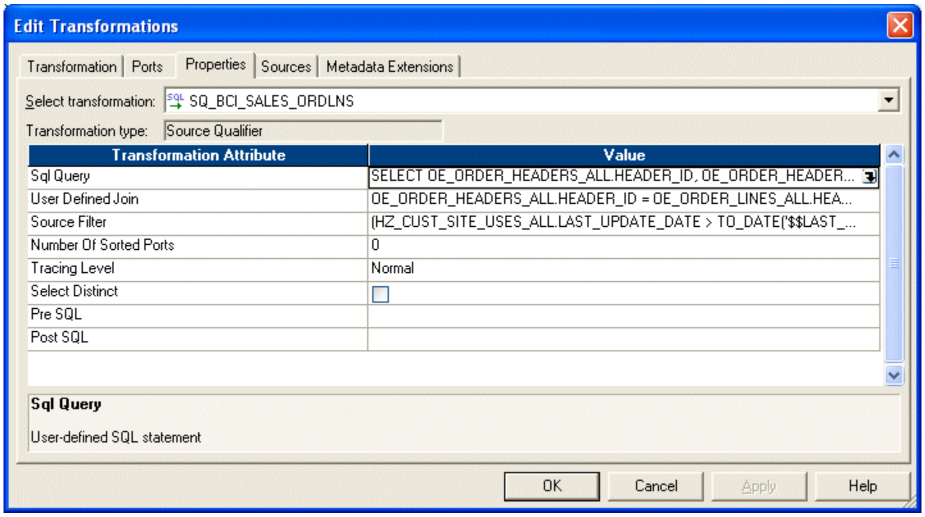


By default, only booked orders are loaded into the Sales Order Lines (W_SALES_ORDER_LINES_F), Sales Schedule Lines (W_SALES_SCHEDULE_LINE_F), and Sales Booking Lines (W_SALES_BOOKING_LINE_F) tables. However, you can also load non-booked orders in Sales Order Lines (W_SALES_ORDERS_LINES_F) and Sales Schedule Lines (W_SALES_SCHEDULE_LINE_F), while loading only booked orders in Sales Booking Lines (W_SALES_BOOKING_LINE_F).

To include non-booked orders in the Sales Order Lines and Sales Schedule Lines tables (for incremental load):

1. In Informatica PowerCenter Designer, open the SDE_ORA115*Version*_Adaptor or SDE_ORAR12_Adaptor folder.

2. Open and check out the mplt_BC_ORA_SalesOrderLinesFact mapplet in the Mapplet Designer.
3. Double-click the `SQ_BCI_SALES_ORDLNS` Source Qualifier to open the Edit Transformations box.
4. Display the Properties tab.



Select transformation: SQ_BCI_SALES_ORDLNS

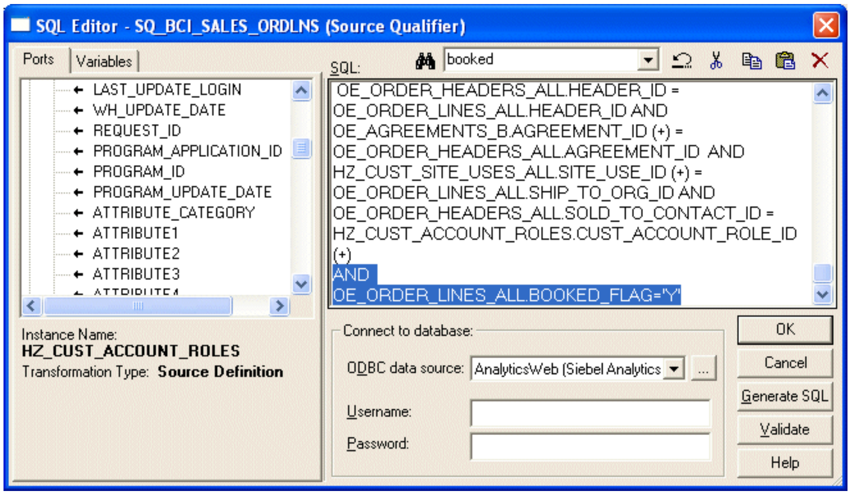
Transformation type: Source Qualifier

Transformation Attribute	Value
Sql Query	SELECT OE_ORDER_HEADERS_ALL.HEADER_ID, OE_ORDER_HEADER...
User Defined Join	OE_ORDER_HEADERS_ALL.HEADER_ID = OE_ORDER_LINES_ALL.HEA...
Source Filter	(HZ_CUST_SITE_USES_ALL.LAST_UPDATE_DATE > TO_DATE('$$LAST_...
Number Of Sorted Ports	0
Tracing Level	Normal
Select Distinct	□
Pre SQL	
Post SQL	

Sql Query

User-defined SQL statement

5. For the Sql Query Transformation Attribute, the User Defined Join Transformation Attribute, and the Source Filter Transformation Attribute, do the following:
   a. Select the down arrow in the Value field to display the SQL Editor box.



Find: booked

```
OE_ORDER_HEADERS_ALL.HEADER_ID =
OE_ORDER_LINES_ALL.HEADER_ID AND
OE_AGREEMENTS_B.AGREEMENT_ID (+) =
OE_ORDER_HEADERS_ALL.AGREEMENT_ID  AND
HZ_CUST_SITE_USES_ALL.SITE_USE_ID (+) =
OE_ORDER_LINES_ALL.SHIP_TO_ORG_ID AND
OE_ORDER_HEADERS_ALL.SOLD_TO_CONTACT_ID =
HZ_CUST_ACCOUNT_ROLES.CUST_ACCOUNT_ROLE_ID
(+)
AND
OE_ORDER_LINES_ALL.BOOKED_FLAG='Y'
```

Instance Name:
HZ_CUST_ACCOUNT_ROLES
Transformation Type: **Source Definition**

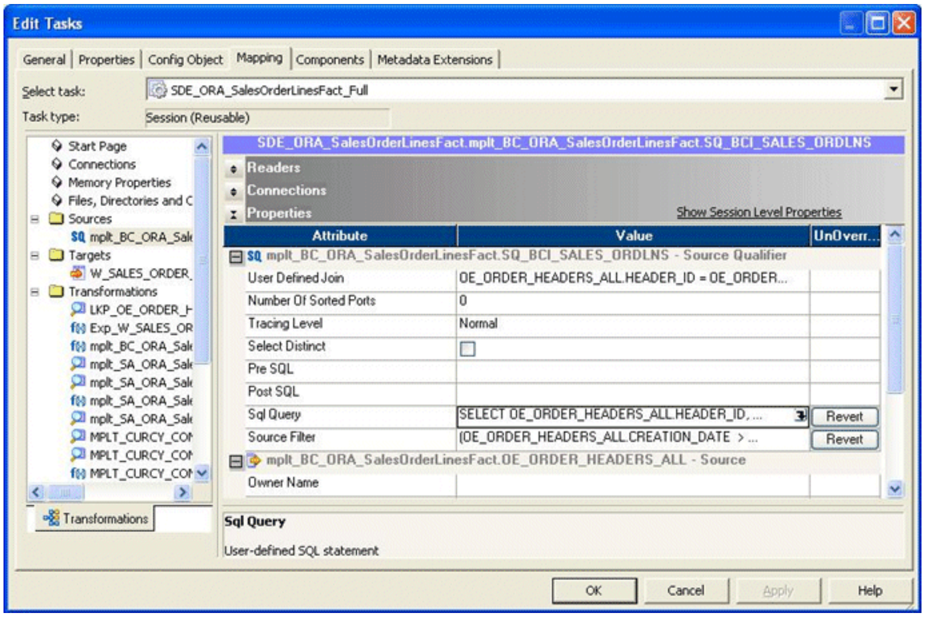
Connect to database:
ODBC data source: AnalyticsWeb (Siebel Analytics
Username:
Password:

   b. In the SQL box, remove the line 'AND OE_ORDER_LINES_ALL.BOOKED_FLAG='Y' (or 'OE_ORDER_LINES_ALL.BOOKED_FLAG='Y' AND') from the Sql Query Transformation Attribute, the User Defined Join Transformation Attribute, and the Source Filter Transformation Attribute. The User Defined Join Transformation Attribute and the Source Filter Transformation Attribute might not have the line.
   c. Click OK to save the changes.
6. Validate and save your changes to the repository, and check in the mapplet.

7. Open the mplt_BC_ORA_SalesOrderLinesFact_Primary in the Mapplet Designer. Repeat steps 3 - 6 for Sales Order Lines Primary table.

To include non-booked orders in the Sales Order Lines and Sales Schedule Lines tables (for Full load):

1. In Informatica PowerCenter Workflow Manager, open the SDE_ORA115*Version*_Adaptor or SDE_ORAR12_Adaptor folder.
2. Drag the SDE_ORA_SalesOrderLinesFact_Full session to the Task Developer.
3. Check out and double-click the session to open the Edit Tasks window.
4. Display the Mapping tab and click mplt_BC_ORA_SalesOrderLinesFact.SQ_BCI_SALES_ORDLNS in the left pane.



Select task: SDE_ORA_SalesOrderLinesFact_Full

Task type: Session (Reusable)

SDE_ORA_SalesOrderLinesFact.mplt_BC_ORA_SalesOrderLinesFact.SQ_BCI_SALES_ORDLNS

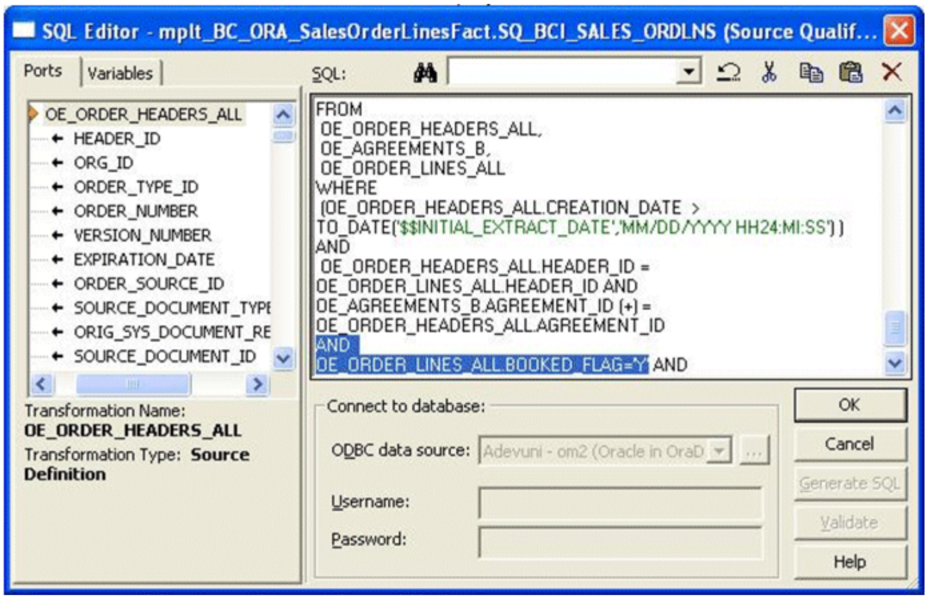
Attribute	Value	UnOverr...
mplt_BC_ORA_SalesOrderLinesFact.SQ_BCI_SALES_ORDLNS - Source Qualifier		
User Defined Join	OE_ORDER_HEADERS_ALL.HEADER_ID = OE_ORDER...	
Number Of Sorted Ports	0	
Tracing Level	Normal	
Select Distinct	□	
Pre SQL		
Post SQL		
Sql Query	SELECT OE_ORDER_HEADERS_ALL.HEADER_ID, ...	Revert
Source Filter	(OE_ORDER_HEADERS_ALL.CREATION_DATE > ...	Revert
mplt_BC_ORA_SalesOrderLinesFact.OE_ORDER_HEADERS_ALL - Source		
Owner Name		

Sql Query

User-defined SQL statement

5. For the Sql Query Attribute, the Source Filter Attribute, and the User Defined Join Attribute under Properties in the right pane, do the following:

   a. Select the down arrow in the Value field to display the SQL Editor box.



```
FROM
 OE_ORDER_HEADERS_ALL,
 OE_AGREEMENTS_B,
 OE_ORDER_LINES_ALL
WHERE
 (OE_ORDER_HEADERS_ALL.CREATION_DATE >
TO_DATE('$$INITIAL_EXTRACT_DATE','MM/DD/YYYY HH24:MI:SS') )
AND
 OE_ORDER_HEADERS_ALL.HEADER_ID =
OE_ORDER_LINES_ALL.HEADER_ID AND
OE_AGREEMENTS_B.AGREEMENT_ID (+) =
OE_ORDER_HEADERS_ALL.AGREEMENT_ID
AND
OE_ORDER_LINES_ALL.BOOKED_FLAG='Y' AND
```

Transformation Name: **OE_ORDER_HEADERS_ALL**

Transformation Type: **Source Definition**

ODBC data source: Adevuni - om2 (Oracle in OraD

Username:

Password:

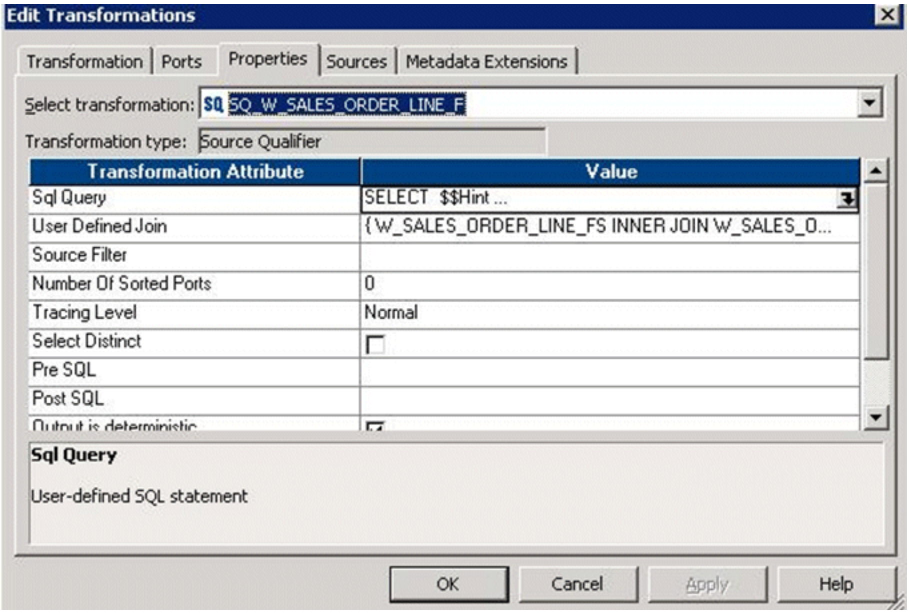
b. In the SQL box, remove the line 'AND OE_ORDER_LINES_ALL.BOOKED_FLAG='Y' (or 'OE_ORDER_LINES_ALL.BOOKED_FLAG='Y' AND') from the Sql Query Attribute, the Source Filter Attribute, and the User Defined Join Attribute. The User Defined Join Attribute or the Source Filter Attribute might not have the line.

c. Click OK to save the changes.

6. Validate and save your changes to the repository, and check in the session.

To include only booked orders in the Sales Booking Lines table:

1. In Informatica PowerCenter Designer, open the SILOS folder.
2. Open and check out the SIL_SalesBookingLinesFact_Load_OrderLine_Credit mapping in the Mapping Designer.
3. Double-click the `SQ_W_SALES_ORDER_LINE_F` Source Qualifier to open the Edit Transformations box.
4. Display the Properties tab.



Select transformation: SQ_W_SALES_ORDER_LINE_F

Transformation type: Source Qualifier

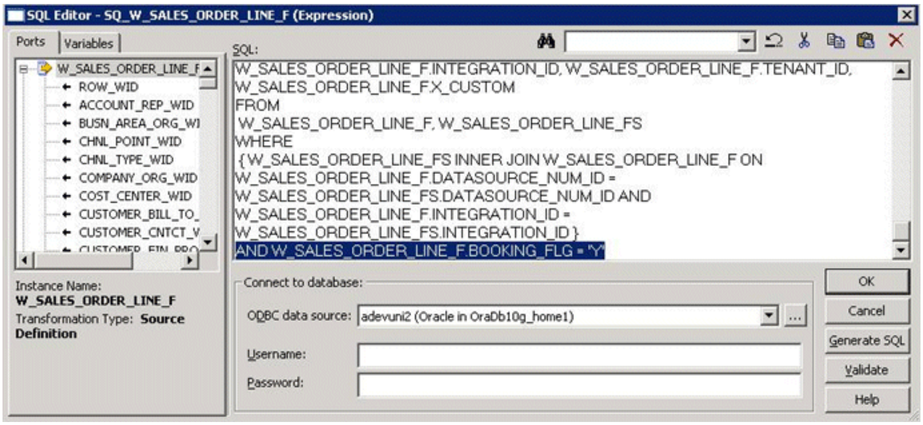
Transformation Attribute	Value
Sql Query	SELECT $$Hint ...
User Defined Join	{W_SALES_ORDER_LINE_FS INNER JOIN W_SALES_O...
Source Filter	
Number Of Sorted Ports	0
Tracing Level	Normal
Select Distinct	☐
Pre SQL	
Post SQL	
Output is deterministic	☑

Sql Query

User-defined SQL statement

5. Perform the following steps for the Source Filter Transformation Attribute and the Sql Query Transformation Attribute, if it is not empty.

   a. Select the down arrow in the Value field to display the SQL Editor box.



```
W_SALES_ORDER_LINE_F.INTEGRATION_ID, W_SALES_ORDER_LINE_F.TENANT_ID,
W_SALES_ORDER_LINE_F.X_CUSTOM
FROM
 W_SALES_ORDER_LINE_F, W_SALES_ORDER_LINE_FS
WHERE
{W_SALES_ORDER_LINE_FS INNER JOIN W_SALES_ORDER_LINE_F ON
W_SALES_ORDER_LINE_F.DATASOURCE_NUM_ID =
W_SALES_ORDER_LINE_FS.DATASOURCE_NUM_ID AND
W_SALES_ORDER_LINE_F.INTEGRATION_ID =
W_SALES_ORDER_LINE_FS.INTEGRATION_ID}
AND W_SALES_ORDER_LINE_F.BOOKING_FLG = 'Y'
```

Instance Name: W_SALES_ORDER_LINE_F

Transformation Type: Source Definition

ODBC data source: adevuni2 (Oracle in OraDb10g_home1)

Username:

Password:

   b. In the SQL box, add ' W_SALES_ORDER_LINE_F.BOOKING_FLG = 'Y'' (plus AND if there are existing filters) to the Sql Query Transformation Attribute and the Source Filter Transformation Attribute.

      Note: Do not add this line to the Sql Query Transformation Attribute if it is empty. In that case, only add the line to the Source Filter Transformation Attribute.

   c. Click OK to save the changes.

6. Validate and save your changes to the repository, and check in the mapping.

7. Repeat steps 2 - 6 for the SIL_SalesBookingLinesFact_Load_OrderLine_Debit, SIL_SalesBookingLinesFact_Load_ScheduleLine_Credit, and SIL_SalesBookingLinesFact_Load_ScheduleLine_Debit mappings.

6.3.2.3 How to Configure Sales Schedule Lines Data Storage

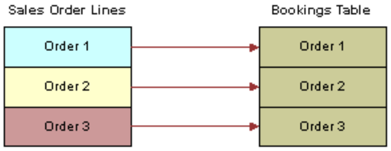
Sales schedule lines detail when each order's items are slated for shipment. Each sales order is broken into sales order lines, and each sales order line can have multiple schedule lines.

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.

6.3.2.4 About Loading Bookings at the Schedule Line Level

As initially configured for Oracle 11i and Oracle R12, 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 Figure 6–2.

Figure 6–2 Sales Order Lines and Bookings Table



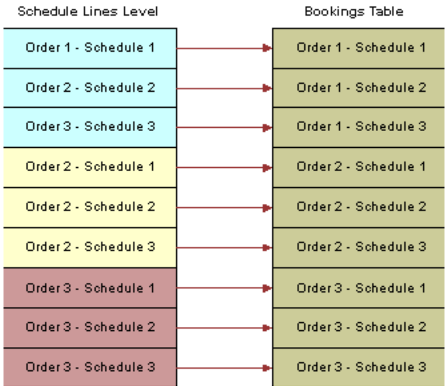
There are two Subject Areas in the SDE_ORA115*Version*_Adaptor or SDE_ORAR12_ Adaptor containers:

- Enterprise Sales - Booking Lines & Order Lines
- Enterprise Sales - Booking Lines & Schedule Lines

The Execution Plans that are installed with Oracle BI Applications use the Enterprise Sales - Booking Lines & Order Lines Subject Area by default. If you want to load booking lines at the schedule line level, create a new Execution Plan and include the Enterprise Sales - Booking Lines & Schedule Lines Subject Area instead of Enterprise Sales - BookingLines & Order Lines.

Bookings may be recorded at the Sales Schedule Line level instead of the Sales Order Line 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, as shown in Figure 6–3. 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.

Figure 6–3 Bookings Recorded at the Schedule Line Level



6.3.2.5 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 mapplet. The mplt_SA_ORA_SalesPickLinesFact mapplet is used by the SDE_ORASalesPickLinesFact mapping.

This mapplet 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. Open the file parameterfileOLTP.txt and locate at the section [SDE_ORA_SalesPickLinesFact].

2. Edit the parameter for the tolerance you want to modify.

   For example:

   - If you want to allow two days after the scheduled pick date before the pick is flagged as late, set the $$PICK_LATE_TIME_TOL=2.

   - To set the number of days before a pick is flagged as early, set the value for $$PICK_EARLY_TIME_TOL

   - To set the number of days before a pick is flagged as late, set the value for $$PICK_LATE_TIME_TOL.

   - If you want to change the shipping tolerances, set the values for the shipping parameters ($$SHIP_LATE_TIME_TOL and $$SHIP_EARLY_TIME_TOL).

3. Validate and save your changes to the parameter file.

6.3.2.6 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.

6.3.2.7 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 and Oracle R12.

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 Informatica PowerCenter Designer, open the SDE_ORA115*Version*_Adaptor or SDE_ORAR12_Adaptor folder.
2. Open the mplt_BC_ORA_SalesInvoiceLinesFact mapplet in the Mapplet Designer.
3. Double-click the `SQ_BCI_SALES_IVCLNS` Source Qualifier to open the Edit Transformations box.
4. Display the Properties tab.
5. For the SQL Query Transformation Attribute, select the down arrow in the Value field to display the SQL Editor box.
6. In the SQL box, remove the line 'AND RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG='Y''.
7. Validate and save your changes to the repository.
8. Repeat steps 2 - 7 for the mplt_BC_ORA_SalesInvoiceLinesFact_Primary mapplet.

6.3.2.8 How to Configure the Order Lines

The Backlogs and Cycle Lines in the out-of-the-box ETL component assumes that the Backlog, Picking and Cycle Lines tables have been updated with shipping and invoicing information (for example, using Oracle EBS Interface programs). If the Oracle Order Line Tables have not been updated with shipping and invoicing information, you need to update the out-of-the-box ETL components as follows:

To configure the Order Lines:

1. In Informatica PowerCenter Designer, open the PLP\Mappings folder.
2. Using the Mapping Designer, edit the PLP_SalesCycleLinesFact_Load mapping and the PLP_SalesCycleLinesFact_Load_Full mappings as follows:
   a. Open the mapping in Mapping Designer.
   b. Double-click the `SQ_W_SALES_ORDER_LINE_F` Source Qualifier to open the Edit Transformations box.
   c. Display the Properties tab.
   d. For the SQL Query Transformation Attribute, select the down arrow in the Value field to display the SQL Editor box.
   e. In the SQL box, replace the SQL text 'X.TOTAL_SHIPPED_QTY' with 'PICKLINE.TOTAL_SHIPPED_QTY'.
   f. In the SQL box, replace the SQL text 'X.TOTAL_INVOICED_QTY' with 'IVCLINE.TOTAL_INVOICE_QTY'.
   g. Validate and save your changes to the repository.
3. In DAC, do the following:
   a. Go to the Design view, and select the appropriate custom container from the drop-down list.
   b. Display the Configuration Tags tab.
   c. Query for the tag 'Sales PLP Optional Tasks'.
   d. Display the Subject Areas subtab.

   e. Activate the appropriate Subject Areas by clearing the Inactive check box.

4. In PowerCenter Designer, open the Source Qualifier in SDE_ORA_SalesPickLinesFact.
5. Modify the SQL Query as follows:
   a. Add to the following join condition:

      ```
      AND WSH_DELIVERY_DETAILS.DELIVERY_DETAIL_ID=WSH_DELIVERY_ASSIGNMENTS.
      DELIVERY_DETAIL_ID (+) AND WSH_DELIVERY_ASSIGNMENTS.DELIVERY_ID=
      WSH_NEW_DELIVERIES.DELIVERY_ID (+)
      ```

   b. Nest the following filter condition within the parenthesis:

      ```
      OR WSH_NEW_DELIVERIES.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE',
      'MM/DD/YYYY HH24:MI:SS')
      ```

   c. Change `select OE_ORDER_LINES_ALL.ACTUAL_SHIPMENT_DATE` to `select WSH_NEW_DELIVERIES.INTIAL_PICKUP_DATE`.
   d. Select `WSH_NEW_DELIVERIES.LAST_UPDATE_DATE` and link it to `EXP_SALES_PCKLNS.LAST_UPDATE_DATE1`.

6.3.2.9 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 5.3.2.5, "How to Configure AR Balance ID for Oracle Receivables Analytics and Oracle General Ledger and Profitability Analytics"
- Section 5.3.2.6, "How to Configure the AR Adjustments Extract for Oracle Receivables Analytics"
- Section 5.3.2.7, "How to Configure the AR Schedules Extract"
- Section 5.3.2.8, "How to Configure the AR Cash Receipt Application Extract for Oracle Receivables Analytics"
- Section 5.3.2.9, "How to Configure the AR Credit-Memo Application Extract for Oracle Receivables Analytics"

6.3.2.10 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(INP_LINE_ID)||'~'||TO_CHAR(INP_INV_ITEM_ID)||'~'||TO_CHAR(INP_WAREHOUSE_
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 ETL 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 mapplet (mplt_SA_ORA_SalesOrderLinesFact). The following section describes what happens if you modify the Booking ID to include the sales representative.

6.3.2.11 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 modify the VAR_BOOKING_ID to use the value:

```
TO_CHAR(INP_LINE_ID)||'~'||TO_CHAR(INP_INV_ITEM_ID)||'~'||to_char(INP_WAREHOUSE_
ID)
```

For example, to edit the VAR_BOOKING_ID value, do the following:

1. In Informatica PowerCenter Designer, open the mplt_SA_ORA_SalesOrderLinesFact mapplet in the Mapplet Designer.
2. Double-click the MAPI_SALES_ORDLNS transformation to open the Edit Transformation box.
3. Display the Ports tab.
4. Select the EXP_SALES_ORDLNS transformation.
5. Edit the expression for the VAR_BOOKING_ID port.

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 6–10.

Table 6–10 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 6–10 is entered into the IA Bookings table (W_SALES_BOOKING_LINE_F) as shown in Table 6–11.

Table 6–11 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 6–12.

Table 6–12 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 SIL_SalesBookingLinesFact_Load_OrderLine_Credit, which also writes to the booking table, now does a debooking for the old line and SIL_SalesBookingLinesFact_Load_OrderLine_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 6–13.

Table 6–13 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

6.3.2.12 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.

6.3.2.13 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 Informatica PowerCenter Designer, open the SDE_ORA115*Version*_Adaptor or SDE_ORAR12_Adaptor folder.
2. Open the mplt_BC_ORA_SalesOrderLinesFact mapplet in the Mapplet Designer.
3. Double-click the EXP_SALES_ORDLNS, and display the Ports tab.
4. Edit the VAR_OPR_BACKLOG_FLG and remove the `OPEN_FLAG='Y'`.
5. Edit the VAR_FIN_BACKLOG_FLG and remove the `OPEN_FLAG='Y'`.
6. Open the mplt_BC_ORA_SalesScheduleLinesFact mapplet in the Mapplet Designer.
7. Double-click the EXP_SALES_SCHLNS, and display the Ports tab.
8. Edit the VAR_OPR_BACKLOG_FLG and remove the `OPEN_FLAG='Y'`.
9. Edit the VAR_FIN_BACKLOG_FLG and remove the `OPEN_FLAG='Y'`.
10. Validate and save your changes to the repository.
11. Open the PLP folder.
12. Open the mappings PLP_SalesBacklogLinesFact_LoadOrderLines and PLP_SalesBacklogLinesFact_LoadScheduleLines.
13. Remove the condition `W_STATUS_CODE <> 'Closed'` from the SQL Query in the Source Qualifier.

6.3.2.14 How to Configure the Bill of Materials Explosion Options

The Bill of Materials (BOM) functional area enables you to analyze the components that comprise the finished goods. BOM enables you to determine how many products use a certain component. It also enables you to get visibility into the complete BOM hierarchy for a finished product. In order to explode BOM structures, certain objects need to be deployed in your EBS system.

Note that to run the ETL as the apps_read_only user, you must first run the following DCL commands from the APPS schema:

```
Grant insert on opi.opi_obia_w_bom_header_ds to &read_only_user;
Grant analyze any to &read_only_user;
```

> **Note:** If you plan to consume analytics on Bill of Materials (BOM), you need to include the DAC Subject Area "Supply Chain - BOM Items" in your ETL Execution Plan. However, the ETL process for extracting BOM data from Oracle EBS into the OBAW data warehouse is not designed for very high volumes of BOM data. There are known performance issues with initial ETL load for very large BOM data size (for example, more than 100,000 rows of data in EBS BOM). Oracle recommends that you restrict the BOM data size in the initial ETL load by applying appropriate filters (for example, On time or Inventory Organization).

You can explode the BOM structure with three different options:

- **All**. All the BOM components are exploded regardless of their effective date or disable date. To *explode* a BOM component is to expand the BOM tree structure.
- **Current**. The incremental extract logic considers any changed components that are currently effective, any components that are effective after the last extraction date, or any components that are disabled after the last extraction date.
- **Current and Future**. All the BOM components that are effective now or in the future are exploded. The disabled components are left out.

These options are controlled by the EXPLODE_OPTION variable. The EXPLODE_OPTION variable is preconfigured with a value of 2, explode Current BOM structure.

In the SDE_ORA_BOM_Explosion mapping, the `OPI_OBIA_BOMPEXPL_KPG` stored procedure calls `bompexpl.exploder_userexit` stored procedure to explode the BOM structure. Table 6–14 lists the variables for the `bompexpl.exploder_userexit` stored procedure.

Table 6–14 Variables for the bompexpl.exploder_userexit Stored Procedure

Input Variable	Preconfigured Value	Description
VERIFY_FLAG	0	A verify flag with a value of 1 only applies to standard BOM.
ORG_ID	ORGANIZATION_ID	Organization ID
ORDER_BY	1	Controls the order of the records. 1—Operation Sequence Number, Item Number 2—Item Number, Operation Sequence Number
GRP_ID	Negative Sequence ID -1, -2, and so on.	Unique value to identify current explosion.
SESSION_ID	Negative Sequence ID -1, -2, and so on.	Unique value to identify current session.
LEVELS_TO_EXPLODE	10	Levels to explode.
BOM_OR_ENG	1	1—BOM 2—ENG
IMPL_FLAG	1	1—Implemented Only 2—Implemented and Non-implemented

Table 6–14 (Cont.) Variables for the bompexpl.exploder_userexit Stored Procedure

Input Variable	Preconfigured Value	Description
PLAN_FACTOR	2	1—Yes 2—No
EXPLODE_OPTION	2	1—All 2—Current 3—Current and Future
MODULE	2	1—Costing 2—BOM 3—Order Entry 4—ATO 5—WSM
CST_TYPE_ID	0	Cost type ID for cost explosion.
STD_COMP_FLAG	0	1—Explode only standard components 2—All components
EXPL_QTY	1	Explosion quantity
ITEM_ID	ROUND(TO_ DECIMAL(PRODUCT_ID))	Item ID of assembly to explode.
ALT_DESG	ALTERNATE_BOM_ DESIGNATOR	Alternate routing designator
COMP_CODE	NULL	Concatenated component code
REV_DATE	TO_CHAR(CREATION_DT, 'YYYY/MM/DD HH24:MI')	Explosion date YYYY/MM/DD HH24:MI

There are five different BOM types in a source system: 1- Model, 2 - Option Class, 3 - Planning, 4 - Standard, and 5 - Product Family. By default, only the Standard BOM type is extracted and exploded.

To configure the BOM explosion to the All option:

1. In Informatica PowerCenter Designer, open the SDE_ORA*Version*_Adaptor.
2. Go to Mapplet Designer, and open mplt_BC_ORA_BOMHeaderDimension.
3. Double-click the SQL qualifier `SQ_BOM_INVENTORY_COMPONENTS` to open the Edit Transformations dialog, display the Properties tab, and open the value for SQL Query.
4. Modify the following default `WHERE` condition from:

```
((

   /* CURRENT valid component changed */

   INV.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY
HH24:MI:SS') AND (INV.EFFECTIVITY_DATE <= TO_DATE('$$CURRENT_DATE','MM/DD/YYYY
HH24:MI:SS') and (INV.DISABLE_DATE > TO_DATE('$$CURRENT_DATE','MM/DD/YYYY
HH24:MI:SS') OR INV.DISABLE_DATE IS NULL))

   OR

   /* Component that became effective after last extract date and before
today's extract, for CURRENT Option*/

   INV.EFFECTIVITY_DATE between TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY
HH24:MI:SS') and TO_DATE('$$CURRENT_DATE','MM/DD/YYYY HH24:MI:SS')

   OR

   /* Component that become disabled after last extract date and before today's
extract, for CURRENT and CURRENT-FUTURE Option*/

   INV.DISABLE_DATE between TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY
HH24:MI:SS') and TO_DATE('$$CURRENT_DATE','MM/DD/YYYY HH24:MI:SS')

   )

   OR BOM.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY
HH24:MI:SS'))

GROUP BY
```

To:

```
   (INV.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY
HH24:MI:SS')

   OR BOM.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY
HH24:MI:SS'))

GROUP BY
```

5. Click Apply, and Validate the mapping and save your changes to the repository.
6. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
7. Display the Tasks tab.
8. Query for task SDE_ORA_BOMItemFact, and then display the parameters subtab.
9. Select the parameter $$EXPLODE_OPTION, edit the Value field and change it to 1, and save.

To configure the BOM explosion to the Current and Future option:

1. In Informatica PowerCenter Designer, open the SDE_ORA*Version*_Adaptor.
2. Go to Mapplet Designer, and open mplt_BC_ORA_BOMHeaderDimension.
3. Double-click the SQL qualifier `SQ_BOM_INVENTORY_COMPONENTS` to open the Edit Transformations dialog, and display the Properties tab, and open the value for SQL Query.
4. Modify the following default `WHERE` condition from:

```
((

/* CURRENT valid component changed */

INV.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY HH24:MI:SS')
AND (INV.EFFECTIVITY_DATE <= TO_DATE('$$CURRENT_DATE','MM/DD/YYYY HH24:MI:SS')
and (INV.DISABLE_DATE > TO_DATE('$$CURRENT_DATE','MM/DD/YYYY HH24:MI:SS') OR
INV.DISABLE_DATE IS NULL))

OR
/* Component that became effective after last extract date and before today's
extract, for CURRENT Option*/

INV.EFFECTIVITY_DATE between TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY
HH24:MI:SS') and TO_DATE('$$CURRENT_DATE','MM/DD/YYYY HH24:MI:SS')

OR

/* Component that become disabled after last extract date and before today's
extract, for CURRENT and CURRENT-FUTURE Option*/

INV.DISABLE_DATE between TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY HH24:MI:SS')
and TO_DATE('$$CURRENT_DATE','MM/DD/YYYY HH24:MI:SS')

)

OR BOM.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY
HH24:MI:SS'))

GROUP BY
```

To:

```
((

INV.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY HH24:MI:SS')

AND ((INV.DISABLE_DATE > TO_DATE('$$CURRENT_DATE','MM/DD/YYYY HH24:MI:SS') OR
INV.DISABLE_DATE IS NULL))

OR

INV.DISABLE_DATE between TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY HH24:MI:SS')
and TO_DATE('$$CURRENT_DATE','MM/DD/YYYY HH24:MI:SS')

)

OR BOM.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY
HH24:MI:SS'))

 GROUP BY
```

5. Click Apply, and Validate the mapping and save your changes to the repository.
6. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
7. Display the Tasks tab.
8. Query for task SDE_ORA_BOMItemFact, and then display the parameters subtab.
9. Select the parameter $$EXPLODE_OPTION, edit the Value field and change it to 3, and save.

To configure the BOM type:

1. In Informatica PowerCenter Designer, open the SDE_ORA*Version*_Adaptor.
2. Open the mplt_BC_ORA_BOMHeaderDimension mapplet.

3. Double-click the SQL qualifier `SQ_BOM_INVENTORY_COMPONENTS` to open the Edit Transformations dialog, and display the Properties tab, and open the value for SQL Query.

4. Modify the `BOM_ITEM_TYPE` section in the `WHERE` statement to change the number to your BOM type. For example, change the number to 3 for a Planning BOM type, as follows:

   ```
   Where INV.BOM_ITEM_TYPE = 3 AND
   M.BOM_ITEM_TYPE = 3 AND
   ```

   > **Note:** You can also remove these two filters to extract all types of BOM.

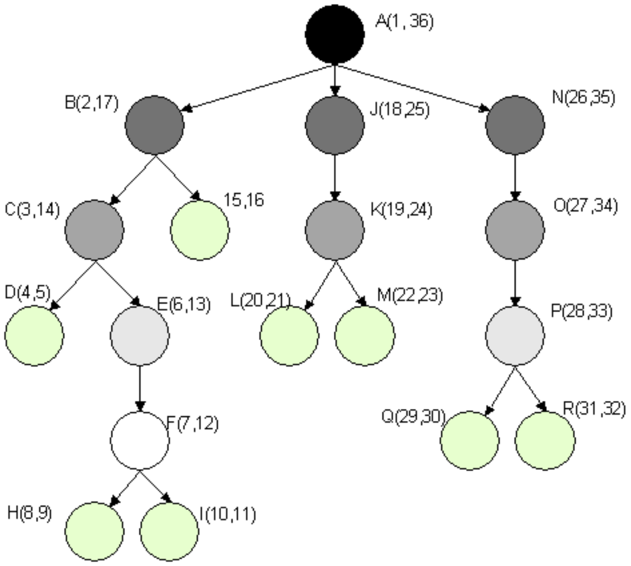
5. Click Apply, and Validate the mapping and save your changes to the repository.

6.3.2.15 How to Configure the Left Bound and Right Bound Calculation Option

You can use the left bound and the right bound calculation to expedite some reporting requirements. For example, you can find the components in a subassembly within a finished product. Left bound and right bound are stored in the W_BOM_ITEM_F table for each BOM node, and they have one row of data in the W_BOM_ITEM_F table. The `COMPUTE_BOUNDS` stored procedure traverses the exploded BOM tree structure and calculates the left bound and right bound. By default, the `COMPUTE_BOUNDS` stored procedure is off and the W_BOM_ITEM_F.LEFT_BOUNDS and W_BOM_ITEM_F.RIGHT_BOUNDS columns are empty.

Figure 6–4 illustrates a sample BOM structure with the left bound and right bound values listed for each node. To find all the components under node B, you select the components with a top product key value of A, the left bound value is greater than 2, and the right bound value is less than 17.

Figure 6–4 Sample BOM Structure



You can use the following procedure to turn on the left bound and the right bound calculation and populate the W_BOM_ITEM_F.LEFT_BOUNDS and W_BOM_ITEM_F.RIGHT_BOUNDS columns.

Note: Before you run an ETL using a BOM, you must compile and deploy the SQL code in Compute_Bounds_Ora11i.sql. See Section 6.2.3.11, "How to Deploy the Stored Procedure for the Left Bound and Right Bound Calculation Option" for more information.

To configure the left bound and right bound calculation option:

1. In Informatica PowerCenter Designer, navigate to the SILOS folder and edit the SIL_BOMItemFact mapping.
2. Double-click the `COMPUTE_BOUNDS` stored procedure transformation to open the Edit Transformations dialog, and display the Properties tab.
3. Change the value of the **Call Text** attribute to 'compute_bounds_ora11i(1)'.
4. Click **Apply**.
5. Validate the mapping, and save your changes to the repository.

7 Configuring Oracle Human Resources Analytics

This section describes how to configure Oracle Human Resources Analytics. It contains the following topics:

Note: All references to Oracle 11*i* also apply to Oracle R12.

7.1 Overview of Oracle Human Resources Analytics

Oracle Human Resources contains information for workforce profile, employee events, compensation, recruitment, leave and absence, and learning.

The Oracle HR Analytics application has the following functional areas:

- **Workforce Profile.** The Workforce Profile functional area is the information foundation for workforce analysis. It provides the comprehensive core workforce information to support analysis on head count, retention, workforce diversity, employee performance, and contingent labor utilization. Key Workforce Profile information such as employee, organization, supervisor, performance band, and service band are shared with other HR functional areas. Sensitive personal attributes like birth date, age, and marital status are organized in a separate folder to allow for restricted access.

  Configurable HR event analysis is another key feature of Workforce Profile functional area. Customers can configure various employee assignment actions to support analysis in voluntary/involuntary termination, hires, transfers, promotions, or layoffs, and so on. In addition, changes in an employee's job, organization, location, supervisor and salary are tracked to support workforce movement analysis.

Specifically, the following analysis is supported by Workforce Profile functional area:

  - Headcount analysis
  - Workforce diversity
  - Employee attrition and retention
  - Employee performance
  - Span of control
  - Internal mobility

- **Compensation.** HR Compensation functional area analyzes employee payroll information that is vital to success in today's economy. It enables you to analyze salaries, benefits, and other rewards that comprise your employee compensation plan. The delivered compensation metrics allow you to correlate employee pay with performance and perform compensation parity analyses at different levels of granularity. It proactively detects over or under-compensated employees, which can have big impact 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 and evaluate the effectiveness of the compensation plan by identifying emerging trends within the organization, or within specific areas of compensation.
- **Recruitment.** Recruitment functional area provides executives, recruiting managers and line managers the intelligence in assessing the efficiency and effectiveness of the recruitment process in sourcing and attracting quality new hires. It delivers over 100 metrics to monitor the entire recruitment life cycle. Specifically, the following recruitment analysis are supported by the Recruitment functional area:
  - Job vacancy analysis
  - Recruitment events analysis
  - Quality of hire
  - Source of hire
  - Applicant pool analysis
  - Referral analysis
- **Learning.** Learning is a key component of Talent Management. The Learning functional area focuses on the analysis of course offerings, delivery methods, course utilization, and learner enrollment and completion. By combining learning and workforce metrics, the Learning functional area provides critical insight into the effectiveness of learning delivery and how learning contributes to workforce development and employee performance.
- **Leave and Absence.** Absenteeism impedes workforce productivity and increase workforce cost. Leave & Absence functional area analyzes workforce leave and absence trends, working days lost, and absenteeism hot spots.
- **Human Resource Effectiveness.** Human Resources Effectiveness allows senior HR executives to monitor key HR effectiveness metrics at the enterprise level. It combines key HR metrics with the organization's financial data. The correlation of workforce and financial metrics provides insight into how workforce trends directly impact the organization's operations and financial strength.

- **U.S. Statutory Compliance.** The U.S. Statutory Compliance functional area delivers monitoring reports to support compliance with US legislative EEO, AAP, and Vets100 reporting.

7.2 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. It contains the following topics:

- Section 7.2.1, "Configuration Steps for Oracle HR Analytics for All Source Systems"
- Section 7.2.2, "Configuration Steps for Oracle HR Analytics for Oracle EBS"
- Section 7.2.3, "Configuration Steps for Flexflelds"
- Section 7.2.4, "Configuration Steps for Oracle HR Analytics for PeopleSoft"
- Section 7.3, "Configuration Steps for Controlling Your Data Set"

Notes

- Before you run Execution Plans for HR Subject Areas, ensure that you have used DAC to set the appropriate Source System Parameters. To do so, in DAC, go to the Design view, and select the appropriate custom container from the drop-down list. Then, display the Source System Parameters tab.
- When you load OLTP data into the HR Workforce Snapshot Fact tables, data is loaded from 01/01/2004 12:00:00 AM out-of-the-box. To load data from an earlier date, use the $$HR_WRKFC_SNAPSHOT_DT parameter on the Source System Parameters tab in DAC to specify a different date. Note that loading more data into the snapshot fact tables can adversely affect ETL performance.

7.2.1 Configuration Steps for Oracle HR Analytics for All Source Systems

This section contains configuration steps required before you do a full data load that apply to all source systems. It contains the following topics:

- Section 7.2.1.1, "Configuring Band Dimensions"
- Section 7.2.1.2, "About Domain Values and CSV Worksheet Files for Oracle HR Analytics"

> **Note:** For configuration steps that apply to all analytics modules (for example, Oracle Financial Analytics, Oracle HR Analytics, or Oracle Sales Analytics), see Chapter 3, "Configuring Common Areas and Dimensions."

7.2.1.1 Configuring Band Dimensions

This section contains information about how to configure the band dimensions for all source systems. It contains the following topics:

- Section 7.2.1.1.1, "Overview of Band Dimensions"
- Section 7.2.1.1.2, "Configuring file_age_band.csv for Age Band Dimension"
- Section 7.2.1.1.3, "Configuring the file_req_age_band.csv for Job Requisition Age Band Dimension"

- Section 7.2.1.1.4, "Configuring the file_perf_age_band.csv for Performance Band Dimension"
- Section 7.2.1.1.5, "Configuring file_emp_pow_band.csv and file_cwk_pow_band.csv for Period of Work Band Dimension"

7.2.1.1.1 Overview of Band Dimensions To enable data analysis based on various groups of a given attribute, Oracle BI Applications provides an option to configure your choice of groups, or bands, for these four attribute families:

- Person Age
- Job Requisition Age
- Performance Ratings
- Period of Service

The band data that you configure is stored in four corresponding dimension tables. Table 7–1 provides a description of each of these tables.

Table 7–1 Dimension Tables that Store Band Data

Dimension Table	Description
W_AGE_BAND_D	Age Band table. This table breaks down the ages of people into different bands to help determine the age ranges the people fall into. The table has two levels: ■ LEVEL_ID = AGE_BAND. This level defines the age bands. ■ LEVEL_ID = AGE. This level defines the age (in months) for a person.
W_JOB_RQSTN_AGE_BAND_D	Job Requisition Age Band table. This table breaks down the age of the job requisition into different bands to help determine the age range the job requisition falls into. The table has two levels: ■ LEVEL_ID = RQSTN_AGE_BAND. This level defines the job requisition age bands. ■ LEVEL_ID = RQSTN_AGE. This level defines the job requisition age (in months).
W_PERFORMANCE_BAND_D	Performance Band table. This table breaks down the performance ratings into different bands to help determine the level of quality of a candidate. The table has two levels: ■ LEVEL_ID = PERF_BAND. This level defines the performance rating bands. ■ LEVEL_ID = PERF_RTNG. This level defines each performance ratings (in integers up to 100) for a person.
W_PRD_OF_WRK_BAND_D	Period of Work Band table. This table breaks down employees and contingent workers into different bands to help determine the time that the employees or the contingent workers have been employed. The table has three levels: ■ Two levels define the bands: LEVEL_ID = POW_BAND_EMP defines the employees' period of work band; LEVEL_ID = POW_BAND_CWK defines the contingent workers' period of work band. ■ LEVEL_ID = POW. This level defines the period of work (in months) for a person.

Each band dimension uses a CSV file to define the band definitions. The CSV files that you need to configure for the band dimensions are:

- file_age_band.csv
- file_req_age_band.csv
- file_perf_age_band.csv
- file_emp_pow_band.csv
- file_cwk_pow_band.csv

> **Note:** If these bands are changed after the initial ETL run, the data in subsequent ETL runs will be inaccurate. Also, any changes to these files will not retroactively change data that has already run through the system.

All band dimensions use one common CSV file, file_row_gen_band.csv, to generate the individual elements that fall into each band. No configuration is required of this common file.

All CSV files for the band dimensions, including the common file, are stored in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).

7.2.1.1.2 Configuring file_age_band.csv for Age Band Dimension The Age Band dimension uses file_age_band.csv. To configure this file, open the CSV file in a text editor and provide inputs based on the column descriptions in Table 7–2.

Table 7–2 Age Band Dimension Columns

Parameter Name	Description
AGE_BAND_CODE	This is the short name of the band and should be unique.
AGE_BAND_DESC	This is a description of the band that is displayed to end users. It is intended to be easier to read than the AGE_BAND_CODE.
BAND_MIN_MONTHS	This is the minimum number of months to qualify for the band (inclusive). If this field is blank, min years is required. If both min months and min years are present, min years is ignored.
BAND_MAX_MONTHS	This is the maximum number of months to qualify for the band (non inclusive). If this field is blank and max years is blank then 1800 is implied, if both max months and max years are present, max years is ignored.
BAND_MIN_YEARS	This is the minimum number of years to qualify for the band (inclusive). If this field is blank, min months is required, if both min months and min years are present, min years is ignored.
BAND_MAX_YEARS	This is the maximum number of years to qualify for the band (non inclusive). If this field is blank and max months is blank then 150 is implied, if both max months and max years are present, max years is ignored.

When defining the data in the CSV file, ensure that:

- The first band starts with a value of 0 for BAND_MIN_MONTHS or BAND_MIN_YEARS.
- The last band does not have a value for BAND_MAX_MONTHS or BAND_MAX_YEARS.
- The bands are contiguous to avoid losing rows in reports, which display by Age Band

Table 7–3 is an example of file_age_band.csv inputs.

Table 7–3 Example of file_age_band.csv Inputs

AGE_BAND_CODE	AGE_BAND_DESC	BAND_MIN_ MONTHS	BAND_MAX_ MONTHS
AGE_BAND_1	<20 Years	0	240
AGE_BAND_2	20-30 Years	240	360
AGE_BAND_3	30-40 Years	360	480
AGE_BAND_4	40-50 Years	480	600
AGE_BAND_5	50-60 Years	600	720
AGE_BAND_6	>=60 Years	720	

7.2.1.1.3 Configuring the file_req_age_band.csv for Job Requisition Age Band Dimension The Job Requisition Age Band dimension uses file_req_age_band.csv. To configure this file, open the CSV file in a text editor and provide inputs based on the column descriptions in Table 7–4.

Table 7–4 Job Requisition Age Band Dimension Columns

Parameter Name	Description
REQ_AGE_BAND_CODE	This is the short code of the band and should be unique.
REQ_AGE_BAND_DESC	This is a description of the band that is displayed for end users. It is intended to be easier to read than the REQ_AGE_BAND_ CODE.
REQ_BAND_MIN_ MONTHS	This is the minimum number of months to qualify for the band (inclusive). If this field is blank, min years is required. If both min months and min years are present, min years is ignored.
REQ_BAND_MAX _ MONTHS	This is the maximum number of months to qualify for the band (non inclusive). If this field is blank and max years is blank then 1800 is implied. If both max months and max years are present, max years is ignored.
REQ_BAND_MIN_YEARS	This is the minimum number of years to qualify for the band (inclusive). If this field is blank, min months is required. If both min months and min years are present, min years is ignored.
REQ_BAND_MAX_YEARS	This is the maximum number of years to qualify for the band (non inclusive). If this field is blank and max months is blank then 150 is implied. If both max months and max years are present, max years is ignored.

When defining the data in the CSV file, ensure that:

- The first band starts with a value of 0 for REQ_BAND_MIN_MONTHS or REQ_ BAND_MIN_YEARS
- The last band does not have a value for REQ_BAND_MAX_MONTHS or REQ_ BAND_MAX_YEARS
- The bands should be contiguous to avoid losing rows in reports which display by Requisition Age Band

Table 7–5 Example of file_req_age_band.csv Inputs

REQ_AGE_BAND_CODE	REQ_AGE_BAND_DESC	REQ_BAND_MIN_MONTHS	REQ_BAND_MAX_MONTHS
REQ_AGE_BAND_1	<1 Month	0	1
REQ_AGE_BAND_2	1 to 3 Months	1	3
REQ_AGE_BAND_3	3 to 6 Months	3	6
REQ_AGE_BAND_4	6 to 12 Months	6	12
REQ_AGE_BAND_5	12 to 24 Months	12	24
REQ_AGE_BAND_6	>= 24 Months	24	

7.2.1.1.4 Configuring the file_perf_age_band.csv for Performance Band Dimension

The Performance Band dimension uses file_perf_age_band.csv. To configure this file, open the CSV file in a text editor and provide inputs based on the column descriptions in Table 7–6.

Table 7–6 Performance Band Dimension Columns

Parameter Name	Description
PERF_BAND_CODE	This is the short code of the band and should be unique.
PERF_BAND_DESC	This is a description of the band that is displayed to end users. It is intended to be easier to read than the PERF_BAND_CODE.
BAND_MIN_NORM_RATING	This is the minimum rating to qualify for the band (inclusive). This field is required.
BAND_MAX_NORM_RATING	This is the maximum rating to qualify for the band (non inclusive). If this field is blank then 100 is implied.

When defining the data in the CSV file, ensure that:

- The first band starts with a value of 0 for BAND_MIN_NORM_RATING.
- The last band does not have a value for BAND_MAX_NORM_RATING.
- The bands are contiguous to avoid losing rows in reports which display by Performance Band

See the related domain value configuration sections for EBS and PeopleSoft:

- For information about configuring the domainValues_perf_nrml_rating_ora*Version*.csv, see Section 7.2.2.13, "How to Configure the domainValues_perf_nrml_rating_oraVersion.csv."
- For information about configuring the domainValues_perf_nrml_rating_psft.csv, see Section 7.2.4.19, "How to Configure the domainValues_perf_nrml_rating_psft.csv."

Table 7–7 Example of file_perf_age_band.csv Inputs

PERF_BAND_CODE	PERF_BAND_DESC	BAND_MIN_NORM_RATING	BAND_MAX_NORM_RATING
PERF_BAND_1	<50 Percent	0	50
PERF_BAND_2	50 to 60 Percent	50	60
PERF_BAND_3	60 to 70 Percent	60	70
PERF_BAND_4	70 to 80 Percent	70	80

Table 7–7 (Cont.) Example of file_perf_age_band.csv Inputs

PERF_BAND_CODE	PERF_BAND_DESC	BAND_MIN_NORM_ RATING	BAND_MAX_NORM_ RATING
PERF_BAND_5	80 Percent and Above	80	

7.2.1.1.5 Configuring file_emp_pow_band.csv and file_cwk_pow_band.csv for Period of Work Band Dimension The Period of Work Band dimension uses these two files:

- file_emp_pow_band.csv. Use this file to configure the employee information.
- file_cwk_pow_band.csv. Use this file to configure the contingent worker information.

To configure the file_emp_pow_band.csv, open the file in a text editor and provide inputs based on the column descriptions in Table 7–8.

Table 7–8 Employee Columns in the file_emp_pow_band.csv for the Period of Work Band Dimension

Column Name	Description
EMP_POW_BAND_CODE	This is the short code of the band and should be unique.
EMP_POW_BAND_DESC	This is a description of the band that is displayed to end users. It is intended to be easier to read than the EMP_POW_BAND_CODE.
EMP_BAND_MIN_MONTHS	This is the minimum number of months to qualify for the band (inclusive). If this field is blank, min years is required. If both min months and min years are present, min years is ignored.
EMP_BAND_MAX_MONTHS	This is the maximum number of months to qualify for the band (non inclusive). If this field is blank and max years is blank then 1800 is implied, if both max months and max years are present, max years is ignored.
EMP_BAND_MIN_YEARS	This is the minimum number of years to qualify for the band (inclusive). If this field is blank, min months is required. If both min months and min years are present, min years is ignored.
EMP_BAND_MAX_YEARS	This is the maximum number of years to qualify for the band (non inclusive). If this field is blank and max months is blank then 150 is implied. If both max months and max years are present, max years is ignored.

When defining the data in the CSV file, ensure that:

- The first band starts with a value of 0 for EMP_BAND_MIN_MONTHS or EMP_BAND_MIN_YEARS
- The last band does not have value for EMP_BAND_MAX_MONTHS or EMP_BAND_MAX_YEARS
- The bands are contiguous to avoid losing rows in reports which display by Period of Placement band

Table 7–9 Example of file_emp_pow_band.csv Inputs

EMP_POW_BAND_ CODE	EMP_POW_BAND_ DESC	EMP_BAND_MIN_ MONTHS	EMP_BAND_MAX_ MONTHS
EMP_BAND_1	<1 Year	0	12
EMP_BAND_2	1 to 2 Years	12	24

Table 7–9 (Cont.) Example of file_emp_pow_band.csv Inputs

EMP_POW_BAND_ CODE	EMP_POW_BAND_ DESC	EMP_BAND_MIN_ MONTHS	EMP_BAND_MAX_ MONTHS
EMP_BAND_3	2 - 5 Years	24	60
EMP_BAND_4	5 to 10 Years	60	120
EMP_BAND_5	10 to 20 Years	120	240
EMP_BAND_6	>= 20 Years	240	

For contingent worker data, use the file_cwk_pow_band.csv file. Open the file in a text editor and provide inputs based on the column descriptions in Table 7–10.

Table 7–10 Contingent Worker Columns in the file_cwk_pow_band.csv for the Period of Work Band Dimension

Column Name	Description
CWK_POW_BAND_CODE	This is the short code of the band and should be unique.
CWK_POW_BAND_DESC	This is a description of the band that is displayed to end users. It is intended to be easier to read than the CWK_POW_BAND_ CODE.
CWK_BAND_MIN_ MONTHS	This is the minimum number of months to qualify for the band (inclusive). If this field is blank, min years is required. If both min months and min years are present, min years is ignored.
CWK_BAND_MAX_ MONTHS	This is the maximum number of months to qualify for the band (non inclusive). If this field is blank and max years is blank, then 1800 is implied. If both max months and max years are present, max years is ignored.
CWK_BAND_MIN_YEARS	This is the minimum number of years to qualify for the band (inclusive). If this field is blank, min months is required. If both min months and min years are present, min years is ignored.
CWK_BAND_MAX_YEARS	This is the maximum number of years to qualify for the band (non inclusive). If this field is blank and max months is blank, then 150 is implied. If both max months and max years are present, max years is ignored.

When defining the data in the CSV file, ensure that:

- The first band starts with a value of 0 for CWK_BAND_MIN_MONTHS or CWK_ BAND_MIN_YEARS
- The last band does not have a value for CWK_BAND_MAX_MONTHS or CWK_ BAND_MAX_YEARS
- The bands are contiguous to avoid losing rows in reports which display by Period of Placement band

Table 7–11 Example of file_cwk_pow_band.csv Inputs

CWK_POW_BAND_ CODE	CWK_POW_BAND_ DESC	CWK_BAND_MIN_ MONTHS	CWK_BAND_MAX_ MONTHS
CWK_BAND_1	<1 Month	0	1
CWK_BAND_2	1 Month	0	1
CWK_BAND_3	1 to 3 Months	1	3
CWK_BAND_4	3 to 6 Months	3	6

Table 7–11 (Cont.) Example of file_cwk_pow_band.csv Inputs

CWK_POW_BAND_CODE	CWK_POW_BAND_DESC	CWK_BAND_MIN_MONTHS	CWK_BAND_MAX_MONTHS
CWK_BAND_5	12 to 24 Months	12	24
CWK_BAND_6	>= 24 Months	24	

7.2.1.2 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 $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

For more information on configuring domain values with CSV worksheet files, see Section 17.13, "About Domain Values" and Section 17.14, "Configuring the Domain Value Set with CSV Worksheet Files."

For more information on domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

> **Note:** Incorrect mappings may result in inaccurate calculations of Oracle Business Intelligence metrics.
>
> Some sessions may fail if these procedures are not compiled in the database before running the workflows.

> **Note:** When editing CSV files, ensure that you:
>
> - Do not change the case of values in the CSV files for columns with the name format W_<xyz>_CODE; however, if required, carefully change the case of values for the columns with the name format W_xyz_DESC or W_<xyz >_NAME.
>
>   For example, do not change 'CONTRACTOR' to 'Contractor'.
> - Do not add new values, other than the values present in the CSV file, to the columns with the name format W_ columns.
>
>   In other words, if you add new rows to the spreadsheet, then the columns with the name format W_ values must map to those in the default spreadsheet. If you add new columns with the name format W_ values, then you must customize the affected mappings.

7.2.2 Configuration Steps for Oracle HR Analytics for Oracle EBS

This section contains configuration steps required before you do a full data load that apply to Oracle EBS. It contains the following topics:

- Section 7.2.2.1, "Domain Values and CSV Worksheet Files for Oracle EBS"
- Section 7.2.2.2, "How to Configure the domainValues_AbsenceEvent_Status_oraVersion.csv"
- Section 7.2.2.3, "How to Configure the domainValues_Employment_Cat_oraVersion.csv"
- Section 7.2.2.4, "How to Configure the domainValues_Employee_Ethnic_Group_Code_oraVersion.csv"

- Section 7.2.2.5, "How to Configure the domainValues_Employment_Full_Time_Flg_oraVersion.csv"
- Section 7.2.2.6, "How to Configure the domainValues_Employee_Sex_MF_oraVersion.csv"
- Section 7.2.2.7, "How to Configure the domainValues_Employment_Status_oraVersion.csv"
- Section 7.2.2.8, "How to Configure the domainValues_Employee_Veteran_Status_Code_oraVersion.csv"
- Section 7.2.2.9, "How to Configure the domainValues_Flsa_oraVersion.csv"
- Section 7.2.2.10, "How to Configure the domainValues_HRPosition_Active_Pos_Flg_oraVersion.csv"
- Section 7.2.2.11, "How to Configure the domainValues_Pay_Type_Flg_oraVersion.csv"
- Section 7.2.2.12, "How to Configure the domainValues_Pay_Type_Grp_Code_oraVersion.csv"
- Section 7.2.2.13, "How to Configure the domainValues_perf_nrml_rating_oraVersion.csv"
- Section 7.2.2.14, "How to Configure the domainValues_Recruitment_Event_Reason_oraVersion.csv"
- Section 7.2.2.15, "How to Configure the domainValues_Recruitment_Source_Type_oraVersion.csv"
- Section 7.2.2.16, "How to Configure the domainValues_Requisition_Category_oraVersion.csv"
- Section 7.2.2.17, "How to Configure the domainValues_Status_Recruitment_oraVersion.csv"
- Section 7.2.2.18, "How to Configure the domainValues_Status_Vacancy_oraVersion.csv"
- Section 7.2.2.19, "How to Configure the domainValues_Wrkfc_EventType_oraVersion.csv"
- Section 7.2.2.20, "How to Configure the domainValues_LearningEnrollment_Status_oraVersion.csv"
- Section 7.2.2.21, "How to Configure the domainValues_Wrkfc_Appraisals_oraVersion.csv"
- Section 7.2.2.22, "How to Configure the file_Normal_Hours_Conversion_Factor_oraVersion.csv"
- Section 7.2.2.23, "How to Configure Address Types for the Employee Dimension"
- Section 7.2.2.24, "How to Configure Phone Types for HR Profile"
- Section 7.2.2.25, "How to Configure Education Degree Codes for Employee Dimension"
- Section 7.2.2.26, "How to Configure Flags for the Pay Type Dimension"
- Section 7.2.2.27, "How to Configure Classification Names for Payroll"

7.2.2.1 Domain Values and CSV Worksheet Files for Oracle EBS

Table 7–12 lists the CSV worksheet files and the domain values for Oracle HR Analytics in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

Table 7–12 Domain Values and CSV Worksheet Files for Oracle E-Business Suite

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_AbsenceEvent_Status_ora*Version*.csv	W_ABSENCE_EVENT_D.APPROVAL_STATUS	Lists the absence approval status and their corresponding domain value of Approval Status.	SDE_ORA_AbsenceEvent
domainValues_Employment_Cat_ora*Version*.csv	W_EMPLOYMENT_D.W_EMPLOYEE_CAT_CODE	Lists the User Person Types and their corresponding domain values of 'Employment Category Code' for the Oracle 11i Application.	SDE_ORA_Employment Dimension, SDE_ORA_Employment Dimension_Full, SDE_ORA_Employment Dimension_Workforce
domainValues_Employee_Ethnic_Group_Code_ora*Version*.csv	W_EMPLOYEE_D.W_ETHNIC_GRP_CODE	Lists the Ethnic codes and their corresponding domain values of 'Ethnic Group Code' for the Oracle 11i Application.	SDE_ORA_Employee Dimension, SDE_ORA_Employee Dimension_Full
domainValues_Employment_Full_Time_Flg_ora*Version*.csv	W_EMPLOYMENT_D.W_FULL_TIME_FLG	Lists the Employment Categories and their corresponding domain values of 'Full Time Flag' for the Oracle 11i Application.	SDE_ORA_Employment Dimension, SDE_ORA_Employment Dimension_Full, SDE_ORA_Employment Dimension_Workforce
domainValues_Employee_Sex_MF_ora*Version*.csv	W_EMPLOYEE_D.W_SEX_MF_CODE	Lists the Sex codes and their corresponding domain values of 'Sex Code' for the Oracle 11i Application.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employment_Status_ora*Version*.csv	W_EMPLOYMENT_D.W_EMPLOYMENT_STAT_CODE	Lists the Per System Statuses and their corresponding domain values of 'Employment Status' for the Oracle 11i Application.	SDE_ORA_Employment Dimension, SDE_ORA_Employment Dimension_Full, SDE_ORA_EmploymentDimension_Workforce
domainValues_Employee_Veteran_Status_Code_ora*Version*.csv	W_EMPLOYEE_D.W_VETERAN_STATUS_CODE	Lists the Veteran codes and their corresponding domain values of 'Veteran Status Code' for the Oracle 11i Application.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_EventTypes_ora*Version*.csv **Note:** This file is not used for Version 7.9.6 or later.	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 the Oracle 11i Application.	-

Table 7–12 (Cont.) Domain Values and CSV Worksheet Files for Oracle E-Business Suite

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_Flsa_ora*Version*.csv	W_EMPLOYMENT_D.W_EXEMPT_FLG, W_JOB_D.W_FLSA_STAT_CODE, W_HR_POSITION_D.W_EXEMPT_FLG	Lists the FLSA Statuses and their corresponding domain values of 'Exempt Flag' for the Oracle R12 Application.	SDE_ORA_EmploymentDimension_Workforce, SDE_ORA_JobDimension, SDE_ORA_JobDimension_Full, SDE_ORA_HRPositionDimension
domainValues_HRPosition_Active_Pos_Flg_ora*Ver*sion.csv	W_HR_POSITION_D.W_ACTIVE_POSITION_FLG	Lists the Position Statuses and their corresponding domain values of 'Active Position Flag' for the Oracle 11i Application.	SDE_ORA_HRPosition Dimension, SDE_ORA_HRPositionDimension
domainValues_Pay_Type_Flg_ora*Version*.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 the Oracle 11i Application.	SDE_ORA_PayTypeDimension, SDE_ORA_PayTypeDimension_Full
domainValues_Pay_Type_Grp_Code_ora*Version*.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 the Oracle 11i Application.	SDE_ORA_PayTypeDimension, SDE_ORA_PayTypeDimension_Full
domainValues_Perf_nrml_rating_ora*Version*.csv	W_PERFORMANCE_BAND_D.PERF_BAND_CODE, PERF_BAND_NAME	Lists the performance ratings and their corresponding domain values of Performance Band Code/Name.	SIL_PerformanceBandDimension
domainValues_Recruitment_Event_Reason_ora*Version*.csv	W_RCRTMNT_EVENT_TYPE_D.W_EVENT_CODE, W_RCRTMNT_EVENT_TYPE_D.W_SUB_STAGE_CODE, W_RCRTMNT_EVENT_TYPE_D.W_STAGE_CODE, W_RCRTMNT_EVENT_TYPE_D.W_REASON_CODE, W_RCRTMNT_EVENT_TYPE_D.W_REASON_TYPE_CODE	Lists the Job Requisition Statuses and various Recruitment Events and their corresponding domain values of 'Recruitment Event Code', 'Recruitment Sub Stage Code', 'Recruitment Stage Code', 'Recruitment Event Reason Code' and 'Recruitment Event Reason Type Code' for Oracle EBS.	SDE_ORA_RecruitmentEventTypeDimension_ApplicantEvents, SDE_ORA_RecruitmentEventTypeDimension_ApplicantEvents_Full, SDE_ORA_RecruitmentEventTypeDimension_ApplicantTerminated, SDE_ORA_RecruitmentEventTypeDimension_ApplicantTerminated_Full, SDE_ORA_RecruitmentEventTypeDimension_Hire_Leave, SDE_ORA_RecruitmentEventTypeDimension_Hire_Leave_Full, SDE_ORA_RecruitmentEventTypeDimension_RequisitionStatuses, SDE_ORA_RecruitmentEventTypeDimension_RequisitionStatuses_Full, SDE_ORA_RecruitmentEventTypeDimension_Unspecified_Full

Table 7–12 (Cont.) Domain Values and CSV Worksheet Files for Oracle E-Business Suite

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_Recruitment_Source_Type_ora*Version*.csv	W_RCRTMNT_SOURCE_D.W_SOURCE_TYPE_CODE	Lists the Recruitment sources (details) and their corresponding domain values of 'Recruitment Source Type Code' for Oracle EBS.	SDE_ORA_RecruitmentSourceDimension_Detail, SDE_ORA_RecruitmentSourceDimension_Detail_Full, SDE_ORA_RecruitmentSourceDimension_Type, SDE_ORA_RecruitmentSourceDimension_Type_Full
domainValues_Requisition_Category_ora*Version*.csv	W_JOB_RQSTN_D.W_CATEGORY_CODE	Lists the Job Requisition Categories and their corresponding domain values of 'Job Requisition Category Code' for Oracle EBS.	SDE_ORA_JobRequisitionDimension, SDE_ORA_JobRequisitionDimension_Full
domainValues_Status_Recruitment_ora*Version*.csv	W_STATUS_D.W_STATUS_CLASS, W_STATUS_D.W_STATUS_CODE	Lists the Recruitment Statuses and their corresponding domain values of 'Status Class' and 'Status Code' for Oracle EBS.	SDE_ORA_StatusDimension_RecruitmentStatus, SDE_ORA_StatusDimension_RecruitmentStatus_Full
domainValues_Status_Vacancy_ora*Version*.csv	W_STATUS_D.W_STATUS_CLASS, W_STATUS_D.W_STATUS_CODE	Lists the Job Requisition Statuses and their corresponding domain values of 'Status Class' and 'Status Code' for Oracle EBS.	SDE_ORA_StatusDimension_RequisitionStatus, SDE_ORA_StatusDimension_RequisitionStatus_Full
domainValues_Wrkfc_EventType_ora*Version*.csv	W_WRKFC_EVENT_TYPE_D.W_EVENT_GRP_CODE, W_WRKFC_EVENT_TYPE_D.W_EVENT_SUBG_CODE, W_WRKFC_EVENT_TYPE_D.EVENT_CODE, W_WRKFC_EVENT_TYPE_D.PROMOTION_FLG, W_WRKFC_EVENT_TYPE_D.TRANSFER_FLG	Lists the Event Types, Event Codes and Meanings and their corresponding domain values of 'Event Group', 'Event Sub-Group', and 'Event' for the Oracle R12 Application.	SDE_ORA_PersistedStage_WorkforceEventDimension_Domain

Table 7–12 (Cont.) Domain Values and CSV Worksheet Files for Oracle E-Business Suite

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_LearningEnrollment_Status_ora*Version*.csv	W_STATUS_D.W_STATUS_CODE	Maps source statuses onto a warehouse status. Measures/dates/FKs are set accordingly.	SDE_ORA_StatusDimension_LearningEnrollmentStatus, SDE_ORA_StatusDimension_LearningEnrollmentStatus_Full
domainValues_Wrkfc_Appraisals_ora*Version*.csv	W_ORA_WEVT_PERF_PS (Filter)	Defines which appraisals to include or exclude (for example, include "Annual Performance Reviews," exclude any other type).	SDE_ORA_PersistedStage_WorkforceEvent_Performance_Full, SDE_ORA_PersistedStage_WorkforceEvent_Performance_Incr
file_Normal_Hours_Conversion_Factor_ora*Version*.csv	W_ORA_WEVT_ASG_PS.NORMAL_HOURS_MONTH	Used to convert working hours to an annualized figure, which is then divided by 12 to store hours by month.	SDE_ORA_PersistedStage_WorkforceEvent_Assignment_Full, SDE_ORA_PersistedStage_WorkforceEvent_Assignment_Incr

7.2.2.2 How to Configure the domainValues_AbsenceEvent_Status_oraVersion.csv

This section explains how to configure the domainValues_AbsenceEvent_Status_ora*Version*.csv file.

Note: This task is Oracle EBS R12-specific. For Oracle EBS 11.5.10 source systems, this configuration is not required.

1. Identify Approval Status values in your Oracle source system by using the following SQL:

```
SELECT DISTINCT APPROVAL_STATUS
FROM PER_ABSENCE_ATTENDANCES
```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the domainValues_AbsenceEvent_Status_ora*Version*.csv file in a text editor. In the file, verify whether the information provided in the following table is available, starting from the sixth line.

APPROVAL_STATUS	ABSENCE_STATUS_CODE	ABSENCE_STATUS_NAME
Y	APPROVED	Approved
N	UNAPPROVED	Not Approved
-	REQUESTED	Requested or In Progress

You can update or customize values for the ABSENCE_STATUS_CODE and ABSENCE_STATUS_NAME domain values.

7.2.2.3 How to Configure the domainValues_Employment_Cat_oraVersion.csv

This section explains how to configure the domainValues_Employment_Cat_ora*Version*.csv file. A sub category is added to the employment dimension.

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','CWK','OTHER','EMP_APL','EX_EMP','EX_EMP_
APL','RETIREE','PRTN')
ORDER BY 1,2
```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the domainValues_Employment_Cat_ora*Version*.csv file in a text editor.
3. From the SQL query result, copy the User Person Types and System Person Types data to the domainValues_Employment_Cat_ora*Version*.csv file in the USER_PERSON_TYPE and SYS_PERSON_TYPE columns. Copy data starting from the sixth line.
4. In the domainValues_Employment_Cat_ora*Version*.csv file, map each combination of the User Person Type and System Person Type data (results of the SQL query) to one delivered combination of the W_EMPLOYEE_CATEGORY_CODE and W_EMPLOYEE_SUB_CAT_CODE domain value.

   You can map more than one combination of the User Person Type and System Person Type data to the same combination of the W_EMPLOYEE_CATEGORY_CODE and W_EMPLOYEE_SUB_CAT_CODE domain value by adding a new row. For example:

USER_PERSON_TYPE	SYS_PERSON_TYPE	W_EMPLOYEE_CATEGORY_CODE
Contractor	OTHER	CONTINGENT
Contingent Worker	CWK	CONTINGENT

5. Save and close the file.

7.2.2.4 How to Configure the domainValues_Employee_Ethnic_Group_Code_oraVersion.csv

This section explains how to configure the domainValues_Employee_Ethnic_Group_Code_ora*Version*.csv file.

1. Identify the Ethnic Group Codes in your Oracle source system by using the following SQL:

```
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','13','BA','BC','BO','C','I'
,'O','P','W')
 ORDER BY 1
```

2. Using a text editor, open the domainValues_Employee_Ethnic_Group_Code_ora*Version*.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the PER_INFORMATION1 to the ETHNIC_CODE column in the file. The data must be copied starting from the sixth line.
4. Map each Ethnic Code to one domain value.
5. Save and close the file.

7.2.2.5 How to Configure the domainValues_Employment_Full_Time_Flg_oraVersion.csv

This section explains how to configure the domainValues_Employment_Full_Time_Flg_ora*Version*.csv file. The employment dimension is enhanced to include the Regular and Temporary worker configuration.

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. From in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the domainValues_Employment_Full_Time_Flg_ora*Version*.csv file in a text editor.
3. From the SQL query result, copy the Employment Category data to the domainValues_Employment_Full_Time_Flg_ora*Version*.csv file in the EMPLOYMENT_CATEGORY column. Copy data starting from the sixth line.
4. Map an Employment Category in the Oracle E-Business Suite source system to two columns: FULL_TIME_FLAG and W_REG_TEMP<_CODE | _DESC> on the Oracle E-Business Suite on the Employment Category.

   In the domainValues_Employment_Full_Time_Flg_ora*Version*.csv file, map each EMPLOYMENT_CATEGORY value to one domain value. Use the columns to the right of the EMPLOYMENT_CATEGORY value to define whether the EMPLOYMENT_CATEGORY value corresponds to a full time employment category (FULL_TIME_FLAG) or not and if worker status is (REG) Regular Worker or (TEMP) Temporary Worker (W_REG_TEMP_CODE, W_REG_TEMP_DESC).
5. Save and close the file.

7.2.2.6 How to Configure the domainValues_Employee_Sex_MF_oraVersion.csv

This section explains how to configure the domainValues_Employee_Sex_MF_ora*Version*.csv file.

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. Using a text editor, open the domainValues_Employee_Sex_MF_ora*Version*.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the SEX column to the SEX column in the file. The data must be copied starting from the sixth line.
4. Map each Sex Code to one domain value.
5. Save and close the file.

7.2.2.7 How to Configure the domainValues_Employment_Status_oraVersion.csv

This section explains how to configure the domainValues_Employment_Status_ora*Version*.csv file.

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. Using a text editor, open the domainValues_Employment_Status_ora*Version*.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the PER_SYSTEM_STATUS to the PER_SYSTEM_STATUS column in the file. The data must be copied starting from the sixth line.
4. Map each PER_SYSTEM_STATUS to one domain value.
5. Save and close the file.

7.2.2.8 How to Configure the domainValues_Employee_Veteran_Status_Code_oraVersion.csv

This section explains how to configure the domainValues_Employee_Veteran_Status_Code_ora*Version*.csv file.

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_INFORMATION_CATEGORY = 'US'
 AND PER_INFORMATION5 IS NOT NULL
 ORDER BY 1
```

2. Using a text editor, open the domainValues_Employee_Veteran_Status_Code_ora*Version*.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the PER_INFORMATION5 column to the VETERAN_STATUS_CODE column in the file. The data must be copied starting from the sixth line.
4. Map each Veteran Status Code to one domain value.
5. Save and close the file.

7.2.2.9 How to Configure the domainValues_Flsa_oraVersion.csv

This section explains how to configure the domainValues_Flsa_ora*Version*.csv file.

1. Identify the FLSA Statuses in your Oracle source system by using the following SQL:

```
SELECT
 DISTINCT CASE WHEN JOB_INFORMATION_CATEGORY IN ('US','CA')
 THEN JOB_INFORMATION3 END FLSA_STAT_CODE
FROM PER_JOBS
ORDER BY 1
```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the domainValues_Employment_Exempt_Flg_ora*Version*.csv file in a text editor.
3. From the SQL query result, copy the FLSA_STAT_CODE data to the domainValues_Flsa_ora*Version*.csv file in the FLSA_STATUS_CODE column. Copy data starting from the sixth line.
4. Map each FLSA_STATUS_CODE value to one domain value.
5. Save and close the file.

7.2.2.10 How to Configure the domainValues_HRPosition_Active_Pos_Flg_oraVersion.csv

This section explains how to configure the domainValues_HRPosition_Active_Pos_Flg_ora*Version*.csv file.

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. Using a text editor, open the domainValues_HRPosition_Active_Pos_Flg_ora*Version*.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the STATUS to the STATUS column in the file. The data must be copied starting from the sixth line.
4. Map each position STATUS to one domain value.
5. Save and close the file.

7.2.2.11 How to Configure the domainValues_Pay_Type_Flg_oraVersion.csv

This section explains how to configure the domainValues_Pay_Type_Flg_ora*Version*.csv file.

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. Using a text editor, open the domainValues_Pay_Type_Flg_ora*Version*.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
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 sixth line.
4. Map each _DEBIT_OR_CREDIT to one domain value.
5. Save and close the file.

7.2.2.12 How to Configure the domainValues_Pay_Type_Grp_Code_oraVersion.csv

This section explains how to configure the domainValues_Pay_Type_Grp_Code_ora*Version*.csv file.

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. Using a text editor, open the domainValues_Pay_Type_Grp_Code_ora*Version*.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
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 sixth line. Use commas to separate the entries.

   Note: Do not change lines one to four in the domainValues_Pay_Type_Grp_Code_ora*Version*.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.

7.2.2.13 How to Configure the domainValues_perf_nrml_rating_oraVersion.csv

This section explains how to configure the domainValues_perf_nrml_rating_ora*Version*.csv file. Using the Oracle source system's performance functionality determines the configuration options for the domainValues_perf_nrml_rating_ora*Version*.csv file.

1. Identify the Performance Ratings in your Oracle source system by using the following SQL:

```
SELECT perf_rating,rating_level_id,rating_desc,
  CASE WHEN max_rating > 0 THEN
    ROUND(100 * rnk_rating / max_rating, 0)
  END NRMLIZED_RATING,
  NULL PERF_BAND_CODE
FROM
 (SELECT to_char(prl.step_value)   perf_rating,prl.rating_level_id
     rating_level_id,
     prl.name rating_desc,
     prl.step_value rnk_rating,
     MAX(prl.step_value) KEEP (DENSE_RANK LAST ORDER BY prl.step_value) OVER
       (PARTITION BY prl.rating_scale_id)    max_rating
FROM per_rating_levels prl
 WHERE prl.rating_scale_id IN
(
SELECT DISTINCT
 lvl.rating_scale_id
   FROM
 per_rating_levels  lvl,
 per_appraisals     appr
WHERE appr.overall_performance_level_id = lvl.rating_level_id)
UNION ALL
SELECT
lookup_code      perf_rating,
to_number(null)  rating_level_id,
meaning          rating_desc,
RANK() OVER (ORDER BY lookup_code)   rnk_rating,
   SUM(1) OVER ()                    max_rating
  FROM hr_lookups
WHERE lookup_type = 'PERFORMANCE_RATING'
)
ORDER BY 2, 1
```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the domainValues_perf_nrml_rating_ora*Version*.csv file in a text editor.
3. From the SQL query result, copy the PERF_RATING, RATING_LEVEL_ID, and RATING_DESC data into the domainValues_perf_nrml_rating_(ver).csv file. Copy data starting from the sixth line.
4. In the domainValues_perf_nrml_rating_ora*Version*.csv file, map each PERF_RATING, RATING_LEVEL_ID, and RATING_DESC value to one NRMLIZED_RATING or to one PERF_BAND_CODE domain value.
   - The NRMLIZED_RATING values are from 0 to 100 (percent).
   - The PERF_BAND_CODE values should correspond to the PERF_BAND_CODE column in the $PMServer\srcfiles\file_perf_band.csv file (for example, in *INFA_HOME*\server\infa_shared\SrcFiles).

   > **Note:** Executing the SQL statements provided in step 1 will convert rating levels into the normalized rating values and the converted values are written to the NRMLIZED_RATING column; you can change the defaults if required.

5. Save and close the file.

7.2.2.14 How to Configure the domainValues_Recruitment_Event_Reason_oraVersion.csv

This section explains how to configure the domainValues_Recruitment_Event_Reason_ora*Version*.csv file.

> **Note:** You must configure this csv file for the recruitment Subject Area to function properly.

1. Identify the assignment statuses in your Oracle source system by using the following SQL:

```
SELECT TO_CHAR(ASSIGNMENT_STATUS_TYPE_ID) STATUS_CODE, USER_STATUS STATUS_NAME
FROM PER_ASSIGNMENT_STATUS_TYPES
WHERE PER_SYSTEM_STATUS in
 ('ACTIVE_APL','INTERVIEW1','INTERVIEW2','OFFER','ACCEPTED','TERM_APL')
```

2. Identify the Job requisition statuses in your Oracle Source system by using the following SQL:

```
SELECT LOOKUP_CODE, MEANING
FROM HR_STANDARD_LOOKUPS
WHERE LOOKUP_TYPE = 'VACANCY_STATUS'
```

3. Identify the Assignment Change Code and Reasons in your Oracle source system by using the following SQL:

```
SELECT LOOKUP_CODE, MEANING
FROM HR_STANDARD_LOOKUPS
WHERE LOOKUP_TYPE = 'APL_ASSIGN_REASON'
```

4. Identify the Application Termination Code and Reasons in your Oracle source system by using the following SQL:

   ```
   SELECT LOOKUP_CODE, MEANING
   FROM HR_STANDARD_LOOKUPS
   WHERE LOOKUP_TYPE = 'TERM_APL_REASON'
   ```

5. Using a text editor, open the domainValues_Recruitment_Event_Reason_ora*Version*.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

6. Copy the assignment status type and name from the results of using the SQL statement in step 1 to the STATUS_CODE and STATUS_NAME columns in the file respectively. The data must be copied starting from the sixth line. Use commas to separate the entries.

7. Copy the lookup code and meaning from the results of using the SQL statement in step 2 to the STATUS_CODE and STATUS_NAME columns in the file respectively. Append this data to the data copied in step 6. Use commas to separate the entries.

8. The Lookup Code and Meaning in step 3 and step 4 are used to populate the REASON_CODE and REASON_NAME columns in the file. Map the values of STATUS_CODE and STATUS_NAME to valid values of REASON_CODE and REASON_NAME columns in the file respectively. Use commas to separate the entries.

9. To see the STATUS_CODE and REASON_CODE combinations currently used in your Oracle source system for active application statuses, use the following SQL:

   ```
   SELECT DISTINCT
   STATUS_CODE,
   STATUS_NAME,
   REASON_CODE,
   REASON_NAME
   FROM (
   SELECT ASG.ASSIGNMENT_STATUS_TYPE_ID STATUS_CODE,
   STS.USER_STATUS STATUS_NAME,
   ASG.CHANGE_REASON REASON_CODE,
   LKP.MEANING REASON_NAME
   FROM PER_ALL_ASSIGNMENTS_F ASG,
   PER_ASSIGNMENT_STATUS_TYPES STS,
   HR_STANDARD_LOOKUPS LKP
   WHERE ASG.ASSIGNMENT_STATUS_TYPE_ID = STS.ASSIGNMENT_STATUS_TYPE_ID
   AND ASG.CHANGE_REASON = LKP.LOOKUP_CODE(+)
   AND LKP.LOOKUP_TYPE(+) = 'APL_ASSIGN_REASON'
   AND ASG.ASSIGNMENT_TYPE = 'A'
   UNION ALL
   SELECT ASG.ASSIGNMENT_STATUS_TYPE_ID STATUS_CODE,
   STS.USER_STATUS STATUS_NAME,
   ASG.STATUS_CHANGE_REASON REASON_CODE,
   LKP.MEANING REASON_NAME
   FROM IRC_ASSIGNMENT_STATUSES ASG,
   PER_ASSIGNMENT_STATUS_TYPES STS,
   HR_STANDARD_LOOKUPS LKP
   WHERE ASG.ASSIGNMENT_STATUS_TYPE_ID = STS.ASSIGNMENT_STATUS_TYPE_ID
   AND ASG.STATUS_CHANGE_REASON = LKP.LOOKUP_CODE(+)
   AND LKP.LOOKUP_TYPE(+) = 'APL_ASSIGN_REASON'
   )
   ```

> **Note:** Use a REASON_CODE, REASON_NAME of 'Any' in case you have not assigned reason code for some assignment changes. The application first searches for events with the assignment status type and reason combination. If the reason is not configured for a particular assignment status type, it uses the event corresponding to REASON_CODE 'Any.'

10. The REASON_CODE and REASON_NAME for all the job requisition STATUS_CODE and STATUS_NAME values obtained from step 2 should be 'Unspecified.'
11. Map all the SOURCE_CODE, SOURCE_NAME, REASON_CODE, REASON_NAME combinations with the following domain columns:
    - W_EVENT_CODE - Recruitment Event Code
    - W_EVENT_DESC - Recruitment Event Description
    - W_SUB_STAGE_CODE - Recruitment Sub Stage Code
    - W_SUB_STAGE_DESC - Recruitment Sub Stage Description
    - W_STAGE_CODE - Recruitment Stage Code
    - W_STAGE_DESC - Recruitment Stage Description
    - W_REASON_CODE - Recruitment Reason Code
    - W_REASON_DESC - Recruitment Reason Description
    - W_REASON_TYPE_CODE - Recruitment Reason Type Code
    - W_REASON_TYPE_DESC - Recruitment Reason Type Description
    - W_APL_EVENT_FLG - Applicant Event Flag. 'Y' for Applicant Events, 'N' for others
    - W_REQ_EVENT_FLG - Job Requisition Event Flag. 'Y' for job Requisition Events, 'N' for others

    > **Note:** EVENT_SEQ_NUM column orders the events in this file. For example, if there are 6 events called Job Requisition Approved, Application Received, Applicant Interviewed, Applicant Offered, Applicant Offer Accepted and Job Requisition Closed, the event sequence can be 10,20,30,40,50,60 in that order.

    For a list of published domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
12. Save and close the file.

> **Note:** An additional CSV file called file_Recruitment_Event_Reason_Hard_Coded_ora*Version*.csv is used to populate the recruitment event dimension. It creates the automatically generated recruitment events. In addition, when there is no reason specified for an event like hire, termination, first performance evaluation and first transfer, this file is used to create a default event for it.
>
> You do not need to configure anything for this file. Instead, use the file domainValues_Recruitment_Event_Reason_ora*Version*.csv to specify the configurable data in the recruitment event dimension, as described in the preceding procedure.

7.2.2.15 How to Configure the domainValues_Recruitment_Source_Type_oraVersion.csv

This section explains how to configure the domainValues_Recruitment_Source_Type_ora*Version*.csv file.

1. Identify the Recruitment Source Types in your Oracle source system by using the following SQL:

   ```
   SELECT
      HR_STANDARD_LOOKUPS.LOOKUP_TYPE,
      HR_STANDARD_LOOKUPS.LOOKUP_CODE,
      HR_STANDARD_LOOKUPS.MEANING
   FROM HR_STANDARD_LOOKUPS
   WHERE LOOKUP_TYPE = 'REC_TYPE'
   ```

2. Using a text editor, open the domainValues_Recruitment_Source_Type_ora*Version*.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the Lookup Code and Meaning to the SOURCE_TYPE_CODE and SOURCE_TYPE_NAME columns in the file respectively. The data must be copied starting from the sixth line. Use commas to separate the entries.
4. Map each Recruitment Source Type (LOOKUP_CODE) to one domain value for each of the two domain columns - W_SOURCE_TYPE_CODE and W_SOURCE_TYPE_DESC. The Recruitment Source Type Description (MEANING) is extracted with the Source Type to help you map the domain values.
5. Map each of these additional source types, which are added to W_RCRTMNT_SOURCE_D table by the extract mapping:

   - Type = IREC; Name = iRecruitment
   - Type = UNKNOWN; Name = unknown

   To review the list of published values, refer to your configuration or the *Oracle Business Analytics Warehouse Data Model Reference*.
6. Save and close the file.

7.2.2.16 How to Configure the domainValues_Requisition_Category_oraVersion.csv

This section explains how to configure the domainValues_Requisition_Category_ora*Version*.csv file.

1. Identify the Requisition Categories in your Oracle source system by using the following SQL:

```
SELECT
   HR_STANDARD_LOOKUPS.LOOKUP_TYPE,
   HR_STANDARD_LOOKUPS.LOOKUP_CODE,
   HR_STANDARD_LOOKUPS.MEANING
FROM
   HR_STANDARD_LOOKUPS
WHERE
   LOOKUP_TYPE = 'VACANCY_CATEGORY'
```

2. Using a text editor, open the domainValues_Requisition_Category_ora*Version*.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the Lookup Code and Meaning to the CATEGORY_CODE and CATEGORY_NAME columns in the file respectively. The data must be copied starting from the sixth line. Use commas to separate the entries.
4. Map each Requisition Category (LOOKUP_CODE) to one domain value for each of the two domain columns - W_CATEGORY_CODE and W_ CATEGORY_NAME. The Requisition Category Description (MEANING) is extracted with each Category Code to help you map the domain values.

   To review the list of published values, refer to your configuration or *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

7.2.2.17 How to Configure the domainValues_Status_Recruitment_oraVersion.csv

This section explains how to configure the domainValues_Status_Recruitment_ora*Version*.csv file.

1. Identify the Recruitment Statuses in your Oracle source system by using the following SQL:

```
SELECT
  PER_ASSIGNMENT_STATUS_TYPES.PER_SYSTEM_STATUS AS LOOKUP_CODE,
   PER_ASSIGNMENT_STATUS_TYPES.USER_STATUS AS MEANING FROM
   PER_ASSIGNMENT_STATUS_TYPES
WHERE
   PER_SYSTEM_STATUS in
('ACTIVE_APL','INTERVIEW1','INTERVIEW2','OFFER','ACCEPTED','TERM_APL')
 Group By PER_ASSIGNMENT_STATUS_TYPES.PER_SYSTEM_STATUS,
PER_ASSIGNMENT_STATUS_TYPES.USER_STATUS
```

2. Using a text editor, open the domainValues_Status_Recruitment_ora*Version*.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the Lookup Code and Meaning to the PER_SYSTEM_STATUS and USER_STATUS columns in the file, respectively. The data must be copied starting from the sixth line. Use commas to separate the entries.
4. Map each Recruitment Status (LOOKUP_CODE) to one domain value for each of the two domain columns - W_STATUS_CODE and W_STATUS_NAME. The W_STATUS_CLASS domain value column must be the value RECRUITMENT_STATUS for all Recruitment Status codes you put in the file. The Recruitment Status Description - USER_STATUS (MEANING) is extracted with each Status Code to help you map the domain values.

   To review the list of published values, refer to your configuration or *Oracle Business Analytics Warehouse Data Model Reference*.

5. Save and close the file.

7.2.2.18 How to Configure the domainValues_Status_Vacancy_oraVersion.csv

This section explains how to configure the domainValues_Status_Vacancy_ora*Version*.csv file.

1. Identify the Job Requisition Statuses in your Oracle source system by using the following SQL:

```
SELECT
   HR_STANDARD_LOOKUPS.LOOKUP_TYPE,
   HR_STANDARD_LOOKUPS.LOOKUP_CODE,
   HR_STANDARD_LOOKUPS.MEANING
FROM
   HR_STANDARD_LOOKUPS
WHERE
   LOOKUP_TYPE = 'VACANCY_STATUS'
```

2. Using a text editor, open the domainValues_Status_Recruitment_ora*Version*.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the Lookup Code and Meaning to the STATUS_CODE and STATUS_NAME columns in the file, respectively. The data must be copied starting from the sixth line. Use commas to separate the entries.
4. Map each Requisition Status (LOOKUP_CODE) to one domain value for each of the two domain columns - W_STATUS_CODE and W_STATUS_NAME. The W_STATUS_CLASS domain value column must be the value VACANCY_STATUS for all Requisition Status codes you put in the file. The Requisition Status Description (MEANING) is extracted with each Status Code to help you map the domain values.

   To review the list of published values, refer to your configuration or *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

7.2.2.19 How to Configure the domainValues_Wrkfc_EventType_oraVersion.csv

This section explains how to configure the domainValues_Wrkfc_EventType_ora*Version*.csv file. The Workforce Event dimension is available in this release. During data loading, data is loaded into the persisted staging table, W_ORA_DMN_WEVT_TYP_PS, which is used to input data to the dimension table, W_WRKFC_EVENT_TYPE_D.

The default domainValues_Wrkfc_EventType_ora*Version*.csv file contains event groups, sub groups, and events, which are supported by the Workforce Event dimension. Using the EVENT_TYPE, EVENT_REASON, ORG_CHANGE_FLG, JOB_CHANGE_FLG, POS_CHANGE_FLG, GRD_CHANGE_FLG, LOC_CHANGE_FLG, and SUP_CHANGE_FLG columns, map a combination of source system attribute changes to a conformed warehouse event.

By default, events are defined in the domainValues_Wrkfc_EventType_ora*Version*.csv file. These events are listed under the column heading W_EVENT_NAME:

- Assignment Change
- Transfer
- Promotion

- Assignment Start
- New Hire
- Rehire
- Voluntary Termination
- Involuntary Termination
- Assignment End
- FTE Change
- Headcount Change
- Performance Review
- Person Type Change
- Salary Review

To configure the domainValues_Wrkfc_EventType_ora*Version*.csv file:

1. Identify the *potential* list of Workforce Event Reasons in your Oracle source system by using the following SQL:

```
/* Change the language condition to your implementation language */
SELECT 'ASG' EVENT_TYPE
  ,LOOKUP_CODE EVENT_REASON
  ,'ANY' ORG_CHANGE
  ,'ANY' JOB_CHANGE
  ,'ANY' POS_CHANGE
  ,'ANY' GRD_CHANGE
  ,'ANY' LOC_CHANGE
  ,'ANY' SUP_CHANGE
  ,'ASG' W_EVENT_CODE
  ,'ASSIGNMENT EVENT' W_EVENT_DESC
  ,'ASG~OTHER' W_EVENT_SUBG_CODE
  ,'ASSIGNMENT CHANGE' W_EVENT_SUBG_DESC
  ,'ASG~OTHER' W_EVENT_GRP_CODE
  ,'ASSIGNMENT CHANGE' W_EVENT_GRP_DESC
  ,'N' PROMOTION_EVENT_FLG
  ,'N' TRANSFER_EVENT_FLG
 FROM HR_STANDARD_LOOKUPS
  WHERE LOOKUP_TYPE IN ('EMP_ASSIGN_REASON', 'CWK_ASSIGN_REASON')
 UNION ALL
 SELECT 'TERM' EVENT_TYPE
  ,LOOKUP_CODE EVENT_REASON
  ,'N' ORG_CHANGE
  ,'N' JOB_CHANGE
  ,'N' POS_CHANGE
  ,'N' GRD_CHANGE
  ,'N' LOC_CHANGE
  ,'N' SUP_CHANGE
  ,'TERM' W_EVENT_CODE
  ,'TERMINATION' W_EVENT_DESC
  ,'TERM~VOLUNTARY' W_EVENT_SUBG_CODE
  ,'VOLUNTARY TERMINATION' W_EVENT_SUBG_DESC
  ,'TERM~VOLUNTARY' W_EVENT_GRP_CODE
  ,'VOLUNTARY TERMINATION' W_EVENT_GRP_DESC
  ,'N' PROMOTION_EVENT_FLG
  ,'N' TRANSFER_EVENT_FLG
 FROM HR_STANDARD_LOOKUPS
  WHERE LOOKUP_TYPE IN ('LEAV_REAS', 'HR_CWK_TERMINATION_REASONS')
```

```
ORDER BY 1, 2
```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the domainValues_Wrkfc_EventType_ora*Version*.csv file in a text editor.

3. Map each Event Type (LOOKUP_TYPE) data to one domain value for these three domain columns: W_EVENT_GRP_CODE, W_EVENT_SUBG_CODE, and W_EVENT_CODE. The Event Category (LOOKUP_TYPE) and Event Description (MEANING) data are also extracted with the Event Type data to help you map the domain values.

> **Note:** When you are editing the CSV files, ensure that you do not remove the following event types:
>
> - ASG_START
>
>   - HIRE
>
>   - REHIRE
>
> - ASG_END
>
>   - FTE
>
>   - HDC
>
>   - PERF
>
>   - PTYP
>
> In cases where (on the source system) there is no enforced event reason (for example, Assignment Budget Value change), HR analytics provides a mechanism to capture the event. Do not alter or remove these settings from the default domainValues_Wrkfc_EventType_ora*Version*.csv file.
>
> Do not delete the existing examples; however, modify them if required based on the following rules:
>
> For the ASG event types, you can split out by reason and change flags, provide the event_reason value as 'Unassigned' for events that do not have a reason. However, do not modify rows with default value 'Any'.
>
> For the TERM event types, you can split out by reason, but set the change flag value to 'N', provide the event_reason value as 'Unassigned' for events that do not have a reason. However, do not modify rows with default value 'Any'.
>
> For the ASG_END, FTE, HDC, PERF, PTYP, and SAL event types, set the change flags value to 'N' and the Reason value to 'Not Applicable'.

4. Configure data options based on your requirements.

Table 7–13 Configuration Options for the domainValues_Wrkfc_EventType_oraVersion.csv file

Configuration Option or Flags	Values	Description
EVENT_REASON	Any	If you want to configure the *_FLG columns instead of the EVENT_REASON column as you do not consider the event reason given against the source transaction as important to the actual event; then for the EVENT_REASON column, provide value Any. For examples on this configuration option, see the domainValues_Wrkfc_EventType_ora*Version*.csv file. The *_FLG configuration options are provided in this table.
ORG_CHANGE_FLG	Any Y	A change in the ORGANIZATION_ID value between the current and previous effective date records (for the same ASSIGNMENT_ID), in the source table PER_ALL_ASSIGNMENTS_F.
JOB_CHANGE_FLG	Any Y	A change in the JOB_ID value between the current and previous effective date records (for the same ASSIGNMENT_ID), in the source table PER_ALL_ASSIGNMENTS_F.
POS_CHANGE_FLG	Any Y	A change in the POSITION_ID value between the current and previous effective date records (for the same ASSIGNMENT_ID), in the source table PER_ALL_ASSIGNMENTS_F.
GRD_CHANGE_FLG	Any Y	A change in the GRADE_ID value between the current and previous effective date records, in the source table PER_ALL_ASSIGNMENTS_F.
LOC_CHANGE_FLG	Any Y	A change in the LOCATION_ID value between the current and previous effective date records (for the same ASSIGNMENT_ID), in the source table, PER_ALL_ASSIGNMENTS_F.
SUP_CHANGE_FLG	Any Y	A change in the SUPERVISOR_ID value between the current and previous effective date records (for the same ASSIGNMENT_ID), in the source table PER_ALL_ASSIGNMENTS_F. If you do not consider the change in assignment attribute (for example, a Job change) for a source transaction as important or influential to the actual event, then for the *_FLG column, provide the value Any. If you do consider the change in assignment attribute important, then provide the value Y. For examples on these configuration options, see the domainValues_Wrkfc_EventType_ora*Version*.csv file.
Unassigned	Not Available	In cases where (on the source system) it is optional to provide an event reason (for example, Termination Leaving Reason), HR analytics provides a mechanism to capture this reason as unassigned whilst still allowing the conformed mapping. For examples on this configuration option, see the domainValues_Wrkfc_EventType_ora*Version*.csv file, see TERM Unassigned.
Not Applicable	Not Available	In cases where (on the source system) there is no enforced event reason (for example, Assignment Budget Value change), HR analytics provides a mechanism to capture the event. Do not alter or remove these settings from the default domainValues_Wrkfc_EventType_ora*Version*.csv file.

Table 7–13 (Cont.) Configuration Options for the domainValues_Wrkfc_EventType_oraVersion.csv file

Configuration Option or Flags	Values	Description
Identifying a Promotion Event	Not Available	To identify a promotion on the source system, often only the transaction reason is not enough, to support this scenario, formally map a promotion assignment event reason, for example, PR_PROM, or use a combination of one or more of the *_FLG columns. Additionally, set the PROMOTION_EVENT_FLG column value to 'Y' to explicitly indicate that this selection is a promotion event. For examples on this configuration option, in the domainValues_Wrkfc_EventType_ora*Version*.csv file, see information provided in the ASG PR_PROM column.
Identifying a Transfer Event	Identifying a Transfer Event	To identify a transfer on the source system, often only the transaction reason is not enough, to support this scenario, formally map a transfer assignment event reason if you have one, or use a combination of one or more of the *_FLG columns. Additionally, set the TRANSFER_EVENT_FLG column value to 'Y' to explicitly indicate that this selection is a transfer event. For examples on this configuration option, in the domainValues_Wrkfc_EventType_ora*Version*.csv file, see information provided in the column where ORG_CHANGE_FLG = Y.
Identifying a Termination Event (Voluntary or Involuntary)	Not Available	To define the employees' voluntary and involuntary leaving reason mappings, you must provide rows in the domainValues_Wrkfc_EventType_ora*Version*.csv file as employees' voluntary and involuntary leaving reasons is a sub-concept and this information is not captured in the Oracle E-Business Suite source system. For examples on this configuration option, in the domainValues_Wrkfc_EventType_ora*Version*.csv file, see TERM.

5. Save and close the file.

7.2.2.20 How to Configure the domainValues_LearningEnrollment_Status_oraVersion.csv

This section explains how to configure the domainValues_LearningEnrollment_Status_ora*Version*.csv file.

1. Identify the Learning Enrollment Statuses in your Oracle source system by using the following SQL:

```
SELECT DISTINCT
BOOKING_STATUS_TYPE_ID STATUS_CODE,
NAME STATUS_NAME
FROM
OTA_BOOKING_STATUS_TYPES
ORDER BY 1,2
```

2. Using a text editor, open the domainValues_LearningEnrollment_Status_ora*Version*.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the Status Code and Status Name to the STATUS_CODE and STATUS_NAME columns in the file, respectively. The data must be copied starting from the sixth line. Use commas to separate the entries.
4. Map each Status Code to one domain value. Status Names are also extracted with Status Codes to help you map the domain values.
5. Save and close the file.

7.2.2.21 How to Configure the domainValues_Wrkfc_Appraisals_oraVersion.csv

This section explains how to configure the domainValues_Wrkfc_Appraisals_ora*Version*.csv file.

1. Identify the Appraisal Templates in your Oracle source system by using the following SQL:

   ```
   SELECT
   APPRAISAL_TEMPLATE_ID APPRAISAL_TEMPLATE_ID,
   NAME APPRAISAL_TEMPLATE_NAME,
   'Y' EXCLUDE_FLAG
   FROM PER_APPRAISAL_TEMPLATES;
   ```

2. Using a text editor, open the domainValues_Wrkfc_Appraisals_ora*Version*.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Change the EXCLUDE_FLAG to 'N' for appraisals that carry performance ratings you want to display (for example, annual appraisals).
4. Save and close the file.

7.2.2.22 How to Configure the file_Normal_Hours_Conversion_Factor_oraVersion.csv

This section explains how to configure the file_Normal_Hours_Conversion_Factor_ora*Version*.csv file.

1. Identify the Frequency Codes in your Oracle source system by using the following SQL:

   ```
   SELECT LOOKUP_CODE NORMAL_HOURS_FREQ, MEANING
   FROM HR_STANDARD_LOOKUPS A
   WHERE LOOKUP_TYPE = 'FREQUENCY'
   AND ENABLED_FLAG = 'Y';
   ```

2. Using a text editor, open the file_Normal_Hours_Conversion_Factor_ora*Version*.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Add/amend the annual and monthly conversion factors to convert assignment working hours from the entered units in per_all_assignments_f into annual and monthly values.

   For example, an assignment working 40 hours / W (Week) might have the following conversion factors:

   - Annual = 52 (52 weeks per year - so hours per year = 40 * 52 = 2080)
   - Monthly = 52/12 = 4.3 (4.3 weeks per month)

4. Save and close the file.

7.2.2.23 How to Configure Address Types for the Employee Dimension

There are three address fields in the Employee dimension table:

- Permanent address
- Mail address
- Work address

For each of these addresses, Oracle Business Intelligence Applications uses only the primary ones. In addition, the following logic is used to determine the various types of addresses:

- Permanent: Address Type = 'H' (in other words, this is also the home address).
- Mail: Address Type = 'M'. If this is not available, use Permanent address (which can be the best alternate for a mailing address).
- Work: Address Type = 'Default'. If this is not available, use Permanent address (which can be the best alternate for a 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 need 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 Informatica PowerCenter Designer, open the SDE_ORA*Version*_Adaptor directory.
2. In Mapplet Designer, open the mplt_SA_ORA_EmployeeDimension mapplet.
3. For each of the Lookup Procedures Lkp_Work_Address, Lkp_Mail_Address, Lkp_Perm_Address, do the following:
   a. Double click the Lookup Procedure to display the Edit Transformations dialog.
   b. Display the Properties tab.
   c. Change the Value of the Lookup Sql Override attribute to the following:

```
SELECT W_EMPLOYEE_D_ADDRESSES_TMP.STREET_ADDRESS as STREET_ADDRESS, W_
EMPLOYEE_D_ADDRESSES_TMP.TOWN_OR_CITY as TOWN_OR_CITY, W_EMPLOYEE_D_
ADDRESSES_TMP.STATE as STATE,
W_EMPLOYEE_D_ADDRESSES_TMP.REGION as REGION, W_EMPLOYEE_D_ADDRESSES_
TMP.COUNTRY
as COUNTRY, W_EMPLOYEE_D_ADDRESSES_TMP.POSTAL_CODE as POSTAL_CODE, W_
EMPLOYEE_D_ADDRESSES_TMP.DATE_FROM as DATE_FROM, W_EMPLOYEE_D_ADDRESSES_
TMP.ADDRESS_ID as ADDRESS_ID,
W_EMPLOYEE_D_ADDRESSES_TMP.PERSON_ID as PERSON_ID, W_EMPLOYEE_D_ADDRESSES_
TMP.DATASOURCE_NUM_ID as DATASOURCE_NUM_ID, W_EMPLOYEE_D_ADDRESSES_
TMP.DATE_FROM as EFFECTIVE_START_DATE
FROM W_EMPLOYEE_D_ADDRESSES_TMP
WHERE PRIMARY_FLAG = 'Y' AND ADDRESS_TYPE ='W'
```

The outputs of these 3 address lookups are evaluated in the expression Exp_SA_Employee_Addresses. The ports EXT_MAIL_ST_ADDRESS, EXT_MAIL_CITY, EXT_MAIL_STATE_CODE, EXT_MAIL_STATE_NAME, EXT_MAIL_STATE_REGION, EXT_MAIL_COUNTRY and so on have logic to use the respective home address columns

if the mailing address columns are null. When a mail address is not available, if you do not want to assume that the mail address (M) is the same as the home address (H), then you would modify the logic in the outgoing ports EXT_MAIL_ST_ADDRESS, MAIL_CITY, MAIL_STATE_CODE, MAIL_STATE_NAME, MAIL_STATE_REGION, MAIL_COUNTRY and so on as follows:

From:

```
IIF(ISNULL(LKP_MAIL_ADDRESS_ID),LKP_PERM_STREET_ADDRESS,LKP_MAIL_STREET_ADDRES
S)
```

To:

```
MAIL_STREET_ADDRESS
```

Note: The above example for the EXT_MAIL_ST_ADDRESS port only. Repeat the same for EXT_MAIL_CITY, EXT_MAIL_STATE_CODE and so on using the correct port names and the correct ADDRESS_ID column.

4. Validate and save changes to the repository.

   If you are using version control for the Informatica Repository, then check in your changes.

7.2.2.24 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 determine 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 need to modify the existing logic.

To configure Phone Type

1. In Informatica PowerCenter Designer, open the SDE_ORA*Version*_Adaptor directory.
2. In Mapplet Designer, open the mplt_SA_ORA_EmployeeDimension mapplet.
3. For each of the Lookup Procedures Lkp_Work_Phone, Lkp_Pager_Phone, Lkp_Mobile_Phone, Lkp_Fax_Phone, do the following:

   a. Double click the Lookup Procedure to display the Edit Transformations dialog.

   b. Display the Properties tab.

   c. Edit the Value of the Lookup Sql Override attribute.

      For example, if you have a specific phone type for the primary work phone, 'WP', you would change the expression for the SQL override in lookup Lkp_Work_Phone as follows:

      From:

```
SELECT W_EMPLOYEE_D_PHONES_TMP.PHONE_NUMBER as PHONE_NUMBER, W_EMPLOYEE_D_
PHONES_TMP.PARENT_ID as PARENT_ID, W_EMPLOYEE_D_PHONES_TMP.PHONE_TYPE as
PHONE_TYPE, W_EMPLOYEE_D_PHONES_TMP.DATASOURCE_NUM_ID as DATASOURCE_NUM_ID,
W_EMPLOYEE_D_PHONES_TMP.DATE_FROM as DATE_FROM
FROM W_EMPLOYEE_D_PHONES_TMP
```

```
WHERE PHONE_TYPE='W1'
```

To:

```
SELECT W_EMPLOYEE_D_PHONES_TMP.PHONE_NUMBER as PHONE_NUMBER, W_EMPLOYEE_D_
PHONES_TMP.PARENT_ID as PARENT_ID, W_EMPLOYEE_D_PHONES_TMP.PHONE_TYPE as
PHONE_TYPE, W_EMPLOYEE_D_PHONES_TMP.DATASOURCE_NUM_ID as DATASOURCE_NUM_ID,
W_EMPLOYEE_D_PHONES_TMP.DATE_FROM as DATE_FROM
FROM W_EMPLOYEE_D_PHONES_TMP
WHERE PHONE_TYPE='WP'
```

4. Validate and save changes to the repository. If you are using the version controlling for the Informatica Repository, then check in your changes.

7.2.2.25 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 session SDE_ORA_EmployeeDimension uses the Informatica parameter $$QUALIFICATION_CATEGORY_LIST to get the list of categories for the defined Education Degrees in Oracle E-Business Suite. The mapping SDE_ORA_EmployeeDimension uses this parameter to identify and populate Highest Education Degree attribute for the Employee Dimension. Incorrect setup of the parameter could cause the issues with quality of the data for Highest Education Degree attribute in Employee Dimension.

To configure the categories that are used to evaluate the highest education degree code for an employee:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab, and select the SDE_ORA_EmployeeDimension task.
3. Display the Parameters subtab.
4. Enter an appropriate value for the parameter $$QUALIFICATION_CATEGORY_LIST.

   For example, $$QUALIFICATION_CATEGORY_LIST = 'DEGREE', 'DT'.
5. Save your changes.

To obtain the list of the valid values for the education degree codes, log in to the Oracle E-Business Suite Instance using SQL*Plus and execute the following SQL:

```
SELECT LOOKUP_CODE, MEANING
FROM HR_STANDARD_LOOKUPS
WHERE LOOKUP_TYPE = 'PER_CATEGORIES'
```

Select from the returned lookup_code column values and decide which ones are used to identify Education Degrees.

7.2.2.26 How to Configure Flags for the Pay Type Dimension

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, a Regular Compensation Earning type, or a Pension Compensation Earning type. These Flag columns are chosen to contribute towards slowly changing dimensions 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 must 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. Using a text editor, open the file_comp_pension_taxable_flg_ora*Version*.csv file in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
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 Informatica mapping values should be acceptable.

7.2.2.27 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 7–14 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 BI Applications is installed out of the box, but you can modify the values by performing the steps in the following procedure.

To configure the classification name parameters:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab, and select the task SDE_ORA_PayrollFact_Agg_Items.
3. Display the Parameters subtab, and specify the following parameters with an appropriate value.

   Include the following parameter values in ' ':

   - $$CLASSIFICATION_NAMES_FOR_GROSS_PAY='PTO Accruals,Earnings,Imputed Earnings'
   - $$CLASSIFICATION_NAMES_FOR_TOTAL_DEDUCTION='Involuntary Deductions,Pre-Tax Deductions,Voluntary Deductions,Tax Deductions'
   - $$CLASSIFICATION_NAMES_FOR_TOTAL_TAXES='Tax Deductions'
   - $$CLASSIFICATION_NAMES_FOR_NET_PAY_EARNINGS='PTO Accruals,Earnings'
   - $$CLASSIFICATION_NAMES_FOR_NET_PAY_DEDUCTIONS='PTO Accruals,Earnings'
   - $$HINT=/*+ USE_HASH(A T) */
4. Save your changes.

To find out the possible Classification Names existing in your system, 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 ()
```

7.2.3 Configuration Steps for Flexfields

This section explains how to configure the key flexfields in Oracle EBS applications.

For more information on flexfields, see *Oracle Applications Flexfields Guide Release 12*.

Flexfield configurations vary from implementation to implementation, in order to accommodate all kinds of flexfield configurations done at the Oracle EBS application source systems, the attributes sourced from the flexfield columns are parameterized and configurable in Oracle Business Intelligence Applications for HR.

To configure Oracle HR Analytics to support flexfields, do the following:

- Section 7.2.3.1, "How to Generate a Flexfield Configuration File for HCM hr_file_flex_kff_dff_user_config_map.csv"
- Section 7.2.3.2, "How to Configure the Key Flexfields for the Job Dimension"
- Section 7.2.3.3, "How to Configure the Key Flexfields for the HR Position Dimension"

- Section 7.2.3.4, "How to Configure the Key Flexfields for the Pay Grade Dimension"
- Section 7.2.3.5, "How to Configure Collect FTE, Collect Performance Ratings, and Load User Flex Data"

7.2.3.1 How to Generate a Flexfield Configuration File for HCM hr_file_flex_kff_dff_user_config_map.csv

To create a flexfield configuration file:

1. In DAC, display the Execute view.
2. On the Execution Plans tab, run the Execution Plan Human Resources – Oracle R12 – Flexfield.

   If the Execution Plan Human Resources – Oracle R12 – Flexfield runs successfully, then the hr_file_flex_kff_dff_user_config_map.csv file is generated in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).

The generated hr_file_flex_kff_dff_user_config_map.csv file contains information on the flexfield configuration data from the Oracle E-Business Suite source environment.

The hr_file_flex_kff_dff_user_config_map.csv file can be used to configure the following columns:

- Job Dimension:
  - JOB_CODE
  - JOB_NAME
  - JOB_FAMILY
  - JOB_FUNCTION
  - JOB_LEVEL
- HR Position Dimension:
  - POSITION_NUM
- Pay Grade Dimension:
  - PAY_LVL_NAME

Other source-specific flexfields that you want to load into the warehouse might require customizations, depending on your requirements.

The generated file hr_file_flex_kff_dff_user_config_map.csv includes the following columns:

- BI_COL_CODE
- LOGICAL_CODE
- SOURCE_TABLE
- CONTEXT_NAME
- FLEX_VALUE_SET_NAME
- WHERE_CLAUSE_PAR_REVIEW
- VALUE_SET_SQL
- VALUE_SET_WHERE_CALUSE

- SQL_LENGTH
- APP_ID

The columns LOGICAL_CODE, SOURCE_TABLE, CONTEXT_NAME , FLEX_VALUE_SET_NAME, VALUE_SET_SQL, VALUE_SET_WHERE_CALUSE, and APP_ID are populated with the data from the source system. You must configure the column BI_COL_CODE based on the Flex configuration in the source system.

The WHERE_CLAUSE_PAR_REVIEW and SQL_LENGTH columns in the CSV file might require user attention in some cases, as explained in the following paragraphs.

The WHERE_CLAUSE_PAR_REVIEW column in the generated hr_file_flex_kff_dff_user_config_map.csv file has possible values of Yes/No.

Some of the columns (VALUE_SET_SQL, VALUE_SET_WHERE_CLAUSE) might have values that require user attention, because they might have variables defined in the source system in the Flex tables.

If you plan to configure flexfields to any rows in the spreadsheet that have the WHERE_CLAUSE_PAR_REVIEW column set to Yes, then you must review the VALUE_SET_SQL and VALUE_SET_WHERE_CLAUSE for those columns and make sure the variables are replaced with the appropriate values.

The SQL_LENGTH column in the generated file hr_file_flex_kff_dff_user_config_map.csv provides an estimate of the length of the SQL that will be generated for a particular Flex configuration. Currently, there is a limit of 2000 characters for the value set SQL generated for any Flex configuration. For example, if you configure 8 rows in the CSV file for JOB_CODE, and the sum of SQL_LENGTH for these 8 rows is greater than 2000 characters, then the value set SQL used to retrieve Flex Data for JOB_CODE will be truncated and will generate errors while running the mapping. Because of this limitation, you must find alternate solutions if the sum of SQL_LENGTH for any Flex configuration is greater than 2000 characters.

7.2.3.2 How to Configure the Key Flexfields for the Job Dimension

This section provides information on configuring the Job Code, Job Name, and Job Family Code columns.

This set up is optional as the flexfield segments may not be present in your source and if present, the flexfield segments may not map to the intended functional meaning of the warehouse target columns.

To configure the key flexfields for the Job dimension:

1. Open the hr_file_flex_kff_dff_user_config_map.csv file, generated using steps provided in Section 7.2.3.1, "How to Generate a Flexfield Configuration File for HCM hr_file_flex_kff_dff_user_config_map.csv" in a text editor. If you are using Microsoft Excel, then select Data, then select Filter, and then select the Auto Filter option.
2. Filter the SOURCE_TABLE column for the PER_JOBS and PER_JOB_DEFINITIONS data.
3. Identify appropriate rows and assign the following flex codes in the BI_COL_CODE column.

Job Flex Field	BI_COL_CODE
Job Code	JOB_CODE
Job Name	JOB_NAME

Job Flex Field	BI_COL_CODE
Job Family	JOB_FAMILY
Job Function	JOB_FUNCTION
Job Level	JOB_LEVEL

4. Save and close the file.

After making changes to the hr_file_flex_kff_dff_user_config_map.csv file, you must run a full load of your regular ETL.

7.2.3.3 How to Configure the Key Flexfields for the HR Position Dimension

This section provides information on configuring the HR Position dimension column: Position Number.

This set up is optional as the flexfield segments may not be present in your source and if present, the flexfield segments may not map to the intended functional meaning of the warehouse target columns.

To configure the key flexfields for the HR Position dimension:

1. Open the hr_file_flex_kff_dff_user_config_map.csv file, generated using steps provided in Section 7.2.3.1, "How to Generate a Flexfield Configuration File for HCM hr_file_flex_kff_dff_user_config_map.csv" in a text editor. If you are using Microsoft Excel, then select Data, then select Filter, and then select the Auto Filter option.
2. Filter the SOURCE_TABLE column for the PER_POSITION_DEFINITIONS data.
3. Identify appropriate rows and assign the following flex codes in the BI_COL_CODE column.

HR Position Flex Field	BI_COL_CODE
Position Number	POSITION_NUM

4. Save and close the file.

After making changes to the hr_file_flex_kff_dff_user_config_map.csv file, you must run a full load of your regular ETL.

7.2.3.4 How to Configure the Key Flexfields for the Pay Grade Dimension

This section provides information on configuring the Pay Grade dimension column: Pay Level Name.

To configure the key flexfields for the Pay Grade dimension.

This set up is optional as the flexfield segments may not be present in your source and if present, the flexfield segments may not map to the intended functional meaning of the warehouse target columns.

To configure the key flexfields for the Pay Grade dimension.

1. Open the hr_file_flex_kff_dff_user_config_map.csv file, generated using steps provided in Section 7.2.3.1, "How to Generate a Flexfield Configuration File for HCM hr_file_flex_kff_dff_user_config_map.csv" in a text editor. If you are using

Microsoft Excel, then select Data, then select Filter, and then select the Auto Filter option.

2. Filter the SOURCE_TABLE column for the PER_GRADE_DEFINITIONS data.
3. Identify appropriate rows and assign the following flex codes in the BI_COL_CODE column.

HR Pay Level Flex Field	BI_COL_CODE
Pay Level Name	PAY_LVL_NAME

4. Save and close the file.

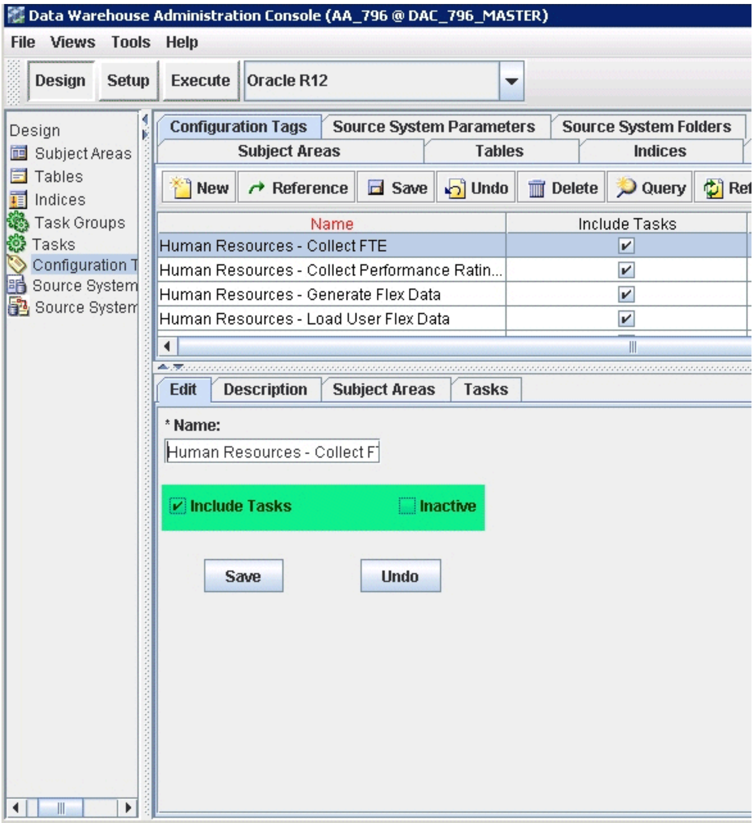
After making changes to the hr_file_flex_kff_dff_user_config_map.csv file, you must run a full load of your regular ETL.

7.2.3.5 How to Configure Collect FTE, Collect Performance Ratings, and Load User Flex Data

To configure the Collect FTE, Collect Performance Ratings, and Load User Flex Data configuration tags:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Configuration Tags tab.
3. In the Edit subtab, use the **Include Tasks** check box to include or exclude the following tags:
   - Human Resources - Collect FTE
   - Human Resources - Collect Performance Ratings
   - Human Resources - Generate Flex Data
   - Human Resources - Load User Flex Data

   Note: Display the Description sub-tab to see a detailed description of a configuration tag.



Name	Include Tasks
Human Resources - Collect FTE	☑
Human Resources - Collect Performance Ratin...	☑
Human Resources - Generate Flex Data	☑
Human Resources - Load User Flex Data	☑

* Name: Human Resources - Collect F

☑ Include Tasks □ Inactive

4. Save the details.

7.2.4 Configuration Steps for Oracle HR Analytics for PeopleSoft

This section contains configuration steps required before you do a full data load that apply to PeopleSoft. It contains the following topics:

- Section 7.2.4.1, "Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft HCM"
- Section 7.2.4.2, "How to Configure the domainValues_Employee_Ethnic_Group_Code_psft.csv"
- Section 7.2.4.3, "How to Configure the domainValues_Employee_Sex_MF_psft.csv"
- Section 7.2.4.4, "How to Configure the domainValues_Employee_Veteran_Status_Code_psft.csv"
- Section 7.2.4.5, "How to Configure the domainValues_Employment_Category_psft.csv"
- Section 7.2.4.6, "How to Configure the domainValues_Flsa_psft.csv"
- Section 7.2.4.7, "How to Configure the domainValues_Employment_Full_Time_Flg_psft.csv"
- Section 7.2.4.8, "How to Configure the domainValues_Employment_Status_psft.csv"
- Section 7.2.4.9, "How to Configure the domainValues_Wrkfc_EventType_psft.csv"
- Section 7.2.4.10, "How to Configure the domainValues_HRPosition_Active_Pos_Flg_psft.csv"

- Section 7.2.4.11, "How to Configure the domainValues_Pay_Type_Grp_Code_psft.csv"
- Section 7.2.4.12, "How to Configure the domainValues_Emp_EventType_EventReason_Code_psft.csv"
- Section 7.2.4.13, "How to Configure the domainValues_Recruitment_Source_Type_psftVersion.csv"
- Section 7.2.4.14, "How to Configure the domainValues_Requisition_Category_psftVersion.csv (WIP)"
- Section 7.2.4.15, "How to Configure the domainValues_Status_Vacancy_Recruitment_psftVersion.csv"
- Section 7.2.4.16, "How to Configure the domainValues_Recruitment_Event_Reason_psftVersion.csv"
- Section 7.2.4.17, "How to Configure the file_pension_comp_flg.csv"
- Section 7.2.4.18, "How to Configure Compensation Flag and Pension Compensation Flag for Pay Type Dimension"
- Section 7.2.4.19, "How to Configure the domainValues_perf_nrml_rating_psft.csv"
- Section 7.2.4.20, "How to Configure the domainValues_Wrkfc_IntlAssign_psft.csv"
- Section 7.2.4.21, "How to Configure the domainValues_AbsenceEvent_Status_psft.csv"
- Section 7.2.4.22, "How to Configure the domainValues_Learning_Enrollment_Status_psftVersion.csv"
- Section 7.2.4.23, "How to Build an Execution Plan for Running HR and GL Subject Areas Together"

7.2.4.1 Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft HCM

The following table lists the CSV worksheet files and the domain values for PeopleSoft HR Analytics in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

> **Note:** Some sessions may fail if these procedures are not compiled in the database before running the workflows.

Table 7–15 Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft HCM

Domain Value	Table	Column Description	Session
domainValues_Employee_Ethnic_Group_Code_psft.csv	W_EMPLOYEE_D.W_ETHNIC_GRP_CODE	Lists the Ethnic codes and corresponding domain values of 'Ethnic Group Code' for PeopleSoft.	SDE_PSFT_EmployeeDimension
domainValues_Employee_Sex_MF_psft.csv	W_EMPLOYEE_D.W_SEX_MF_CODE	Lists the Sex codes and corresponding domain values of 'Sex Code' for PeopleSoft.	SDE_PSFT_EmployeeDimension
domainValues_Employee_Veteran_Status_Code_psft.csv	W_EMPLOYEE_D.W_VETERAN_STATUS_CODE	Lists the Veteran codes and corresponding domain values of 'Veteran Status Code'.	SDE_PSFT_EmployeeDimension

Table 7–15 (Cont.) Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft HCM

Domain Value	Table	Column Description	Session
domainValues_Employment_Category_psft.csv	W_EMPLOYMENT_D.W_EMPLOYEE_CAT_CODE	Lists the User Person Types and corresponding domain values of 'Employment Category Code' for PeopleSoft.	SDE_PSFT_EmploymentDimension SDE_PSFT_EmploymentDimension_Workforce
domainValues_Flsa_psft.csv	W_EMPLOYMENT_D.W_EXEMPT_FLG, W_JOB_D.W_FLSA_STAT_CODE, W_HR_POSITION_D.W_EXEMPT_FLG	Lists the FLSA Statuses and their corresponding domain values of 'Exempt Flag' for the PeopleSoft Application.	SDE_PSFT_EmploymentDimension_Workforce, SDE_PSFT_JobDimension, SDE_PSFT_JobDimension_Full, SDE_ PSFT_HRPositionDimension
domainValues_Employment_Full_Time_Flg_psft.csv	W_EMPLOYMENT_D.W_FULL_TIME_FLG	Lists the Employment Categories and corresponding domain values of 'Full Time Flag'.	SDE_PSFT_EmploymentDimension
domainValues_Employment_Status_psft.csv	W_EMPLOYMENT_D.W_EMPLOYMENT_STAT_CODE	Lists the Per System Statuses and corresponding domain values of 'Employment Status'.	SDE_PSFT_EmploymentDimension
domainValues_HRPosition_Active_Pos_Flg_psft.csv	W_HR_POSITION_D.W_ACTIVE_POSITION_FLG	Lists the Position Statuses and corresponding domain values of 'Active Position Flag'	SDE_PSFT_HRPositionDimension
domainValues_Pay_Type_Grp_code_psft.csv	W_PAY_TYPE_D.W_PAY_TYPE_GRP_CODE	Lists the Classification Names, Element Names and corresponding domain values of 'Pay Type Group Code'	SDE_PSFT_PayTypeDimension_Total, SDE_PSFT_PayTypeDimension_Earnings, SDE_PSFT_PayTypeDimension_FederalTaxes, SDE_PSFT_PayTypeDimension_Deductions, SDE_PSFT_PayTypeDimension_LocalTaxes1, SDE_PSFT_PayTypeDimension_LocalTaxes2, SDE_PSFT_PayTypeDimension_StateTaxes1, SDE_PSFT_PayTypeDimension_StateTaxes2
domainValues_Emp_EventType_EventReason_Code_psft.csv	NA	Lists the event type reason codes.	SDE_PSFT_EventTypeDimension_Stage2

Table 7–15 (Cont.) Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft HCM

Domain Value	Table	Column Description	Session
domainValues_Wrkfc_EventType_psft.csv	W_WRKFC_EVENT_TYPE_D.W_EVENT_GRP_CODE, W_WRKFC_EVENT_TYPE_D.W_EVENT_SUBG_CODE, W_WRKFC_EVENT_TYPE_D.EVENT_CODE, W_WRKFC_EVENT_TYPE_D.PROMOTION_FLG, W_WRKFC_EVENT_TYPE_D.TRANSFER_FLG	Lists the Event Types, Event Codes and Meanings and their corresponding domain values of 'Event Group', 'Event Sub-Group' and 'Event' for the PeopleSoft Application.	SDE_PSFT_PersistedStage_WorkforceEventDimension_Domain
domainValues_Recruitment_Source_Type_psft*Version*.csv	W_RCRTMNT_SOURCE_D.W_SOURCE_TYPE_CODE	Lists the Recruitment sources (details) and corresponding domain values of 'Recruitment Source Type Code' for PeopleSoft.	SDE_PSFT_RecruitmentSourceDimension
domainValues_Requisition_Category_psft*Version*.csv	W_JOB_RQSTN_D.W_CATEGORY_CODE	Lists the Job Requisition Categories and corresponding domain values of 'Job Requisition Category Code' for PeopleSoft.	SDE_PSFT_JobRequisitionDimension, SDE_PSFT_JobRequisitionDimension_Full
domainValues_Status_Vacancy_Recruitment_psft*Version*.csv	W_STATUS_D.W_STATUS_CLASS, W_STATUS_D.W_STATUS_CODE	Lists the Job Requisition and Recruitment Statuses and corresponding domain values of 'Status Class' and 'Status Code' for PeopleSoft.	SDE_PSFT_StatusDimension_Vacancy_Recruitment_Status, SDE_PSFT_StatusDimension_Vacancy_Recruitment_Status_Full
domainValues_Recruitment_Event_Reason_psft*Version*.csv	W_RCRTMNT_EVENT_TYPE_D.W_EVENT_CODE, W_RCRTMNT_EVENT_TYPE_D.W_SUB_STAGE_CODE, W_RCRTMNT_EVENT_TYPE_D.W_STAGE_CODE, W_RCRTMNT_EVENT_TYPE_D.W_REASON_CODE, W_RCRTMNT_EVENT_TYPE_D.W_REASON_TYPE_CODE	Lists the Job Requisition Statuses and various Recruitment Events and corresponding domain values of 'Recruitment Event Code', 'Recruitment Sub Stage Code', 'Recruitment Stage Code', 'Recruitment Event Reason Code' and 'Recruitment Event Reason Type Code' for PeopleSoft.	SDE_PSFT_RecruitmentEventTypeDimension_Stage1, SDE_PSFT_RecruitmentEventTypeDimension_Stage1_Full, SDE_PSFT_RecruitmentEventTypeDimension_Stage2, SDE_PSFT_RecruitmentEventTypeDimension_Stage2_Full
domainValues_perf_nrml_rating_psft.csv	W_PERFORMANCE_BAND_D.PERF_BAND_CODE, PERF_BAND_NAME	Lists the performance ratings and review types and their corresponding domain values of Performance Band Code/Name.	SIL_PerformanceBandDimension

Table 7–15 (Cont.) Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft HCM

Domain Value	Table	Column Description	Session
domainValues_IntlAssign_psft.csv	W_INTL_ASSGN_D.W_COMP_OWNER_FLG	Lists the International Assignment Types and Home/Host Classes and their corresponding domain values of W_COMP_OWNER_FLG (the compensation owner).	SIL_IntlAssignmentDimension
domainValues_AbsenceEvent_Status_psft.csv	W_ABSENCE_EVENT_D.APPROVAL_STATUS	Lists the absence approval statuses ('Y'/'N') and their corresponding domain value of Approval Status.	SDE_PSFT_AbsenceEventDimension
domainValues_Learning_Enrollment_Status_psft*Version*.csv	W_STATUS_D.W_STATUS_CODE	Maps source statuses onto a warehouse status. Measures/dates/FKs are set accordingly.	SDE_PSFT_StatusDimension_LearningEnrollmentStatus SDE_PSFT_StatusDimension_LearningEnrollmentStatus_Full

7.2.4.2 How to Configure the domainValues_Employee_Ethnic_Group_Code_psft.csv

1. Identify the Ethnic Group Codes in your PeopleSoft source system by using the following SQL:

```
SELECT DISTINCT REG_REGION, ETHNIC_GRP_CD
 FROM PS_DIVERS_ETHNIC
 WHERE REG_REGION = 'USA'
 ORDER BY 1
```

2. Using a text editor, open the domainValues_Employee_Ethnic_Group_Code_psft.csv file in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the ETHNIC_GRP_CD to the ETHNIC_CODE column in the file. The data must be copied starting from the sixth line.
4. Map each Ethnic Code to one domain value.
5. Save and close the file.

7.2.4.3 How to Configure the domainValues_Employee_Sex_MF_psft.csv

1. Identify the Sex Codes in your PeopleSoft source system by using the following SQL:

```
SELECT A.FIELDVALUE, A.XLATLONGNAME
 FROM PSXLATITEM A
 WHERE A.EFFDT = (SELECT MAX (C.EFFDT) FROM PSXLATITEM C WHERE C.FIELDNAME =
A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE)
 AND A.FIELDNAME = 'SEX'
 ORDER BY 1
```

2. Using a text editor, open the domainValues_Employee_Sex_MF_psft.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the FIELDVALUE column to the SEX column in the file. The data must be copied starting from the sixth line.
4. Map each Sex Code to one domain value.

5. Save and close the file.

7.2.4.4 How to Configure the domainValues_Employee_Veteran_Status_Code_psft.csv

1. Identify the Veteran Status Codes in your PeopleSoft source system by using the following SQL:

```
SELECT A.FIELDVALUE, A.XLATLONGNAME
 FROM PSXLATITEM A
 WHERE A.EFFDT = (SELECT MAX (C.EFFDT) FROM PSXLATITEM C WHERE C.FIELDNAME =
A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE)
 AND A.FIELDNAME = 'MILITARY_STATUS'
 ORDER BY 1
```

2. Using a text editor, open the domainValues_Employee_Veteran_Status_Code_psft.csv file located in $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the FIELDVALUE column to the VETERAN_STATUS_CODE column in the file. The data must be copied starting from the sixth line.
4. Map each Veteran Status Code to one domain value.
5. Save and close the file.

7.2.4.5 How to Configure the domainValues_Employment_Category_psft.csv

A sub category is added to enhance the employment dimension and to map the EMPL_CLASS in the case of multiple SETID.

To configure the domainValues_Employment_Category_psft.csv file:

1. Identify the User Person Types, EMPL_CLASS and PER_ORG, in the PeopleSoft source system using the following SQL:

```
SELECT EMPL.EMPL_CLASS AS EMP_CAT_CODE,
    EMPL.SETID AS SETID_EMPL_CLASS,
    JOB.PER_ORG AS PER_ORG,
    empl.descrshort,
    COUNT(*) COUNTER
FROM PS_EMPL_CLASS_TBL EMPL,
    PS_JOB JOB
WHERE EMPL.EMPL_CLASS = JOB.EMPL_CLASS
GROUP BY EMPL.EMPL_CLASS,
    EMPL.SETID,
    JOB.PER_ORG,
    empl.descrshort
ORDER BY SETID_EMPL_CLASS,COUNTER DESC
```

Note: You must define PS_JOB.EMPL_CLASS in order to work with metrics derived from employment types. The preceding SQL does not return any data if PS_JOB_EMPL_CLASS is not defined.

2. On the Informatica Server host computer, from the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the domainValues_Employment_Category_psft.csv file in a text editor.
3. From the SQL query result, copy the EMP_CAT_CODE, SETID_EMPL_CLASS, and PER_ORG data to the domainValues_Employment_Category_psft.csv file in the EMP_CAT_CODE, SETID_EMPL_CLASS, and PER_ORG columns. Copy data starting from the sixth line.

4. In the domainValues_Employment_Category_psft.csv file, map each combination of EMP_CAT_CODE, SETID_EMPL_CLASS, and PER_ORG data (results of the SQL query) to one of the delivered combinations of W_EMP_CAT_CODE and W_EMPLOYEE_SUB_CAT_CODE domain value.

   You can map more than one combination of the EMP_CAT_CODE, SETID_EMPL_CLASS, and PER_ORG values to the same combination of the W_EMP_CAT_CODE and W_EMPLOYEE_SUB_CAT_CODE domain value by adding new rows. For examples on mapping data, see the domainValues_Employment_Category_psft.csv file.

   > **Note:** If you have the same EMP_CAT_CODE in multiple SETID_EMPL_CLASS and you are certain that they resolve to the same functional meaning, you can avoid duplication of rows into the domainValues_Employment_Category_psft.csv file by adding the word 'Any' to the SETID_EMPL_CLASS value. For example:
   >
   > ```
   > C, Any, EMP
   > R, STD, EMP
   > ```

5. Save and close the file.

7.2.4.6 How to Configure the domainValues_Flsa_psft.csv

To configure the domainValues_Flsa_psft.csv file.

1. Identify the FLSA Statuses in the PeopleSoft source system using the following SQL:

   ```
   SELECT A.FIELDVALUE, A.XLATLONGNAME
   FROM PSXLATITEM A
   WHERE A.EFFDT = (SELECT MAX (C.EFFDT)
                    FROM PSXLATITEM C
                    WHERE C.FIELDNAME = A.FIELDNAME
                    AND C.FIELDVALUE = A.FIELDVALUE
                    )
   AND A.FIELDNAME = 'FLSA_STATUS'
   ORDER BY 1
   ```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the domainValues_Flsa_psft.csv file in a text editor.
3. From the SQL query result, copy the FIELDVALUE data to the domainValues_Flsa_psft.csv file in the FLSA_STATUS_CODE column. Copy data starting from the sixth line.
4. Map each FLSA_STATUS_CODE value to one domain value.
5. Save and close the file.

7.2.4.7 How to Configure the domainValues_Employment_Full_Time_Flg_psft.csv

This section explains how to configure the domainValues_Employment_Full_Time_Flg_psft.csv file. The employment dimension is enhanced to include a Regular or Temporary worker configuration.

1. Identify the Employment Categories in your PeopleSoft source system by using the following SQL:

   ```
   SELECT CASE WHEN A.FIELDVALUE IN ('F','P') THEN A.FIELDVALUE ELSE NULL END AS
   FULL_PART_TIME,
   ```

```
 A.XLATLONGNAME FULL_PART_DESC,
 CASE WHEN A.FIELDVALUE = 'F' THEN 'R'
 WHEN A.FIELDVALUE = 'P' THEN 'T'
 ELSE NULL END AS REG_TEMP,
 CASE WHEN A.FIELDVALUE = 'F' THEN (SELECT B.XLATLONGNAME
 FROM PSXLATITEM B
 WHERE B.FIELDNAME = 'REG_TEMP'
  AND B.FIELDVALUE = 'R'
  AND B.EFF_STATUS = 'A' )
 WHEN A.FIELDVALUE = 'P' THEN (SELECT C.XLATLONGNAME
 FROM PSXLATITEM C
 WHERE C.FIELDNAME = 'REG_TEMP'
 AND C.FIELDVALUE = 'T'
 AND C.EFF_STATUS = 'A' )
 ELSE NULL END AS REG_TEMP_DESC,
 CASE WHEN A.FIELDVALUE = 'F' THEN 'Y'
 WHEN A.FIELDVALUE = 'P' THEN 'N'
 ELSE NULL END AS FULL_TIME_FLAG
 FROM PSXLATITEM A
 WHERE A.FIELDNAME = 'FULL_PART_TIME'
 AND A.EFF_STATUS = 'A'
```

2. Using a text editor, open the domainValues_Employment_Full_Time_Flg_psft.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

3. From the SQL query result, copy the FIELDVALUE data to the domainValues_Employment_Full_Time_Flg_psft.csv file in the FULL_PART_TIME column. Copy data starting from the sixth line.

4. The Informatica lookup against the domainValues_Employment_Full_Time_Flg_psft.csv is based on the FULL_PART_TIME and REG_TEMP values, they are used to determine if the record is for a Full Time or Part-Time employment and if they are a Regular or Temporary employment record.

   In the domainValues_Employment_Full_Time_Flg_psft.csv file, map each FULL_PART_TIME value to one domain value. Use the columns to the right of the FULL_PART_TIME to define whether the REG_TEMP value corresponds to a Regular or Temporary worker (REG_TEMP), and the worker is a Regular or Temporary worker (W_REG_TEMP_CODE, W_REG_TEMP_DESC).

5. Save and close the file.

7.2.4.8 How to Configure the domainValues_Employment_Status_psft.csv

1. Identify the Per System Statuses in your PeopleSoft source system by using the following SQL:

```
SELECT A.FIELDVALUE, A.XLATLONGNAME
FROM PSXLATITEM A
WHERE A.EFFDT = (SELECT MAX (C.EFFDT) FROM PSXLATITEM C WHERE C.FIELDNAME =
A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE) AND A.FIELDNAME = 'EMPL_STATUS'
ORDER BY 1
```

2. Using a text editor, open the domainValues_Employment_Full_Time_Flg_psft.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

3. Copy the FIELDVALUE to the EMPL_STATUS column in the file. The data must be copied starting from the sixth line.

4. Map each EMPL_STATUS to one domain value.
5. Save and close the file.

7.2.4.9 How to Configure the domainValues_Wrkfc_EventType_psft.csv

This section explains how to configure the domainValues_Wrkfc_EventType_psft.csv file. The Workforce Event dimension is available. During data loading, data is loaded into a new persisted staging table, W_PSFT_DMN_WEVT_TYP_PS, which is used to input data to the dimension table, W_WRKFC_EVENT_TYPE_D.

The default domainValues_Wrkfc_EventType_psft.csv file contains event groups, sub groups, and events, which are supported by the workforce event dimension. Using the columns EVENT_TYPE, EVENT_REASON, ORG_CHANGE_FLG, JOB_CHANGE_FLG, POS_CHANGE_FLG, GRD_CHANGE_FLG, LOC_CHANGE_FLG, and SUP_CHANGE_FLG, map a combination of source system attribute changes to a conformed warehouse event.

By default, the following events are defined in the domainValues_Wrkfc_EventType_psft.csv file under the column heading, EVENT_NAME:

- International Transfer End
- Additional Assignment End
- Layoff
- Assignment Change
- Promotion
- Renew Contract
- Assignment Start
- International Transfer
- Transfer
- New Hire
- Rehire
- Involuntary Termination
- Retirement
- Resignation

To configure the domainValues_Wrkfc_EventType_psft.csv file:

1. Identify the *potential* list of Workforce Event Reasons in the PeopleSoft source system using the following SQL:

   ```
   SELECT ACTION, ACTION_REASON, DESCR FROM PS_ACTN_REASON_TBL ORDER BY 1
   ```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the domainValues_Wrkfc_EventType_psft.csv file in a text editor.
3. Map the PeopleSoft Action, Reason from the SQL to a single row combination of:

   EVENT_TYPE

   EVENT_REASON

   W_EVENT_GRP_CODE

W_EVENT_GRP_NAME

W_EVENT_SUBG_CODE

W_EVENT_SUBG_NAME

EVENT_CODE

EVENT_NAME

Note that the value in EVENT_CODE should be unique, in combination with the EVENT_TYPE and EVENT_REASON columns. The target table is W_WRKFC_EVENT_TYPE_D, unlike in previous releases, which used W_CODE_D.

When setting W_EVENT_GRP_CODE, W_EVENT_SUBG_CODE, and EVENT_CODE values, it is recommended that you prepend SUBG_CODE values with the GRP_CODE value followed by a tilda (~). This naming convention is not strictly required, but as with all W_ columns, the values in the .csv might have dependencies in the RPD.

4. Configure the domainValues_Wrkfc_EventType_psft.csv file based on your requirements.

Table 7–16 Configuration Options for the domainValues_Wrkfc_EventType_psft.csv file

Configuration Options or Flags	Values	Description
EVENT_REASON	Any	If you want to configure the *_FLG columns instead of the EVENT_REASON column as you do not consider the event reason given against the source transaction as important to the actual event, then for the EVENT_REASON column, provide the value "Any." The *_FLG configuration options are provided in this table.
ORG_CHANGE_FLG	Any Y	A change in the DEPTID value between the current and previous records, in the source table PS_JOB.
JOB_CHANGE_FLG	Any Y	A change in the JOBCODE value between the current and previous records, in the source table PS_JOB.
POS_CHANGE_FLG	Any Y	A change in the POSITION_NBR value between the current and previous records, in the source table PS_JOB.
GRD_CHANGE_FLG	Any Y	A change in the GRADE, STEP, or SAL_ADMIN_PLAN value between the current and previous records, in the source table PS_JOB.
LOC_CHANGE_FLG	Any Y	A change in the LOCATION value between the current and previous records, in the source table PS_JOB.
SUP_CHANGE_FLG	Any Y	A change in the SUPERVISOR_ID or REPORTS_TO value between the current and previous date records, in the source table PS_JOB. If you do not consider the change in job attribute (for example, a Job change) for a source transaction as important or influential to the actual event, then for the *_FLG columns, provide value Any. If you do consider the change in assignment attribute important, then provide the value Y.
Unassigned	-	In cases where (on the source system) it is optional to provide an action or action reason, HR analytics provide a mechanism to capture this reason as unassigned whilst still allowing the conformed mapping.

Table 7–16 (Cont.) Configuration Options for the domainValues_Wrkfc_EventType_psft.csv file

Configuration Options or Flags	Values	Description
Not Applicable	-	In cases where (on the source system) there is no enforced action or action reason, HR analytics provides a mechanism to capture the event. Do not alter or remove these settings from the default domainValues_Wrkfc_EventType_psft.csv file.
Mandatory Events	-	To correctly identify hires and terminations: ■ All actions that set the ps_job.hire_dt should be mapped to event group "HIRE" ■ All actions that set the ps_job.termination_dt should be mapped to event group "TERM" For the hire and termination events, set all the change flag values to "N". To verify actions that set the hire or termination dates, check the ps_action_stat_tbl values.
PROMOTION_EVENT_FLG	Y or N	To identify a promotion on the PeopleSoft source system, sometimes only the action or action reason data is not enough. To support this search, formally map a promotion action if you have one (for example, PRO), or use a combination of one or more *_FLG columns. Additionally, set the PROMOTION_EVENT_FLG column value to 'Y' to explicitly indicate that this selection is a promotion event. For examples on this configuration option, see the domainValues_Wrkfc_EventType_psft.csv file.
TRANSFER_EVENT_FLG	Y or N	To identify a transfer on the PeopleSoft source system, sometimes only the Action or Action Reason data is not enough. To support this search, formally map a transfer action or action reason event reason if you have one, or use a combination of one or more * _FLG columns. Additionally, set the TRANSFER_EVENT_FLG column value to 'Y' to explicitly indicate that this selection is a transfer event. For examples on this configuration option, see the domainValues_Wrkfc_EventType_psft.csv file.
Identifying a Termination Event (Voluntary or Involuntary)	-	To define employees' voluntary and involuntary leaving reasons, you must provide rows in the domainValues_Wrkfc_EventType_psft.csv file as employees' voluntary and involuntary leaving reasons is a sub-concept and this information is not captured in the source system. For examples on this configuration option, see W_EVENT_GRP_CODE = TERM in the domainValues_Wrkfc_EventType_psft.csv file.
EXCLUDE_FLG	-	Exclude Flag can be used to prevent certain PeopleSoft Actions from being picked up as events in the Workforce Event Fact. Excluding events that are not important for analysis helps limit the size of the Workforce Event Fact.

5. Save and close the file.

7.2.4.10 How to Configure the domainValues_HRPosition_Active_Pos_Flg_psft.csv

1. Identify the Position Statuses in your PeopleSoft source system by using the following SQL:

```
SELECT A.FIELDVALUE, A.XLATLONGNAME
 FROM PSXLATITEM A
 WHERE A.EFFDT = (SELECT MAX (C.EFFDT) FROM PSXLATITEM C WHERE C.FIELDNAME =
 A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE)
 AND A.FIELDNAME = 'EFF_STATUS'
 ORDER BY 1
```

2. Using a text editor, open the domainValues_HRPosition_Active_Pos_Flg_psft.csv file in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the FIELDVALUE to the EFF_STATUS column in the file. The data must be copied starting from the sixth line.
4. Map each position EFF_STATUS to one domain value.
5. Save and close the file.

7.2.4.11 How to Configure the domainValues_Pay_Type_Grp_Code_psft.csv

Table 7–17 shows pay type sources and their associated group codes.

Table 7–17 Pay Elements for ETL

PAYTYPE_SOURCE	W_PAY_TYPE_GRP_CODE
DEDUCTIONS	DEDUCTIONS
FEDERAL_TAXES	TAXES
LOCAL_TAXES	TAXES
STATE_TAXES	TAXES
EARNINGS	REGULAR
TOTAL	TOTAL

To configure the domainValues_Pay_Type_Grp_Code_psft.csv:

1. Using a text editor, open the domainValues_Pay_Type_Grp_Code_psft.csv file in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
2. Copy the Classification Name and Element Name to the PAYTYPE_SOURCE. The data must be edited starting from the sixth line. Use commas to separate the entries.
3. 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'.
4. Save and close the file.

7.2.4.12 How to Configure the domainValues_Emp_EventType_EventReason_Code_psft.csv

1. Identify the Costing (Debit or Credit) in your PeopleSoft source system by using the following SQL:

   ```
   SELECT ACTION + ':'+ ACTION_REASON EVENT_REASON, DESCR FROM PS_ACTN_REASON_TBL
   ORDER BY 1
   ```

   Oracle Syntax:

   ```
   SELECT ACTION ||':'|| ACTION_REASON EVENT_REASON, DESCR FROM PS_ACTN_REASON_TBL
   ORDER BY 1
   ```

2. Using a text editor, open the domainValues_Emp_EventType_EventReason_Code_psft.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

3. Copy EVENT_REASON to the EVENT_REASON column in the file. The data must be copied starting from the sixth line.
4. Map each EVENT_REASON to one domain value.
5. Save and close the file.

7.2.4.13 How to Configure the domainValues_Recruitment_Source_Type_ psftVersion.csv

This section explains how to configure the domainValues_Recruitment_Source_Type_ psft*Version*.csv file.

1. Identify the Recruitment Source Type in your PeopleSoft source system by using the following SQL:

```
SELECT DISTINCT HRS_SOURCE_ID SOURCE_TYPE_CODE, HRS_SOURCE_NAME SOURCE_TYPE_
NAME
FROM PS_HRS_SOURCE
ORDER BY HRS_SOURCE_ID
```

2. Using a text editor, open the domainValues_Recruitment_Source_Type_psft *Version*.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_ HOME*\server\infa_shared\LkpFiles).
3. Copy the Source Type Code and Source Type Name to the SOURCE_TYPE_CODE and SOURCE_TYPE_NAME columns in the file, respectively. The data must be copied starting from the sixth line. Use commas to separate the entries.
4. Map each Source Type Code (SOURCE_TYPE_CODE) to one domain value for each of the domain columns--W_SOURCE_TYPE_CODE and W_SOURCE_TYPE_ DESC. Source Type Name (SOURCE_TYPE_NAME) is extracted to help you map the domain values.
5. Save and close the file.

7.2.4.14 How to Configure the domainValues_Requisition_Category_ psftVersion.csv (WIP)

This section explains how to configure the domainValues_Requisition_Category_ psft*Version*.csv file.

1. Identify the Requisition Category in your PeopleSoft source system by using the following SQL:

```
SELECT A.FIELDVALUE as CATEGORY_CODE, A.XLATLONGNAME AS CATEGORY_NAME
 FROM PSXLATITEM A
 WHERE A.EFFDT = (SELECT MAX(C.EFFDT) FROM PSXLATITEM C
 WHERE C.FIELDNAME = A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE )
 AND A.FIELDNAME IN ('HRS_JO_TYPE')
```

2. Using a text editor, open the domainValues_Requisition_Category_psft *Version*.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_ HOME*\server\infa_shared\LkpFiles).
3. Copy the category code and category name to the CATEGORY_CODE, CATEGORY_NAME columns in the file respectively. The data must be copied starting from the sixth line. Use commas to separate the entries.
4. Map each category code (CATEGORY_CODE) to one domain value for each of the domain columns--W_CATEGORY_CODE and W_CATEGORY_NAME. Category name (CATEGORY_NAME) is extracted to help you map the domain values.

5. Save and close the file.

7.2.4.15 How to Configure the domainValues_Status_Vacancy_Recruitment_psftVersion.csv

This section explains how to configure the domainValues_Status_Vacancy_Recruitment_psft*Version*.csv file.

1. Identify the Vacancy/Requisition and Recruitment Status in your PeopleSoft source system by using the following SQL:

   ```
   SELECT STATUS_CODE, DESCRSHORT AS STATUS_NAME,
    CASE WHEN STATUS_AREA = 1 THEN 'VACANCY_STATUS' ELSE 'RECRUITMENT_STATUS' END
   AS W_STATUS_CLASS
    FROM PS_HRS_STS_TBL
    WHERE PS_HRS_STS_TBL.STATUS_AREA IN ('1', '3')
   ```

2. Using a text editor, open the domainValues_Status_Vacancy_Recruitment_psft*Version*.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the Status Code, Status Name, and Status Class to the STATUS_CODE, STATUS_NAME, and W_STATUS_CLASS columns in the file respectively. The data must be copied starting from the sixth line. Use commas to separate the entries.
4. Map each Status Code (STATUS_CODE) to one domain value for each of the domain columns--W_STATUS_CLASS, W_STATUS_CODE, and W_STATUS_NAME. Status Name (STATUS_NAME) is extracted to help you map the domain values.
5. Save and close the file.

7.2.4.16 How to Configure the domainValues_Recruitment_Event_Reason_psftVersion.csv

This section explains how to configure the domainValues_Recruitment_Event_Reason_psft*Version*.csv file.

1. Identify the Recruitment Event Reasons in your PeopleSoft source system by using the following SQL:

   ```
   /* Select all status codes. All of these rows do not have a reason code. */
   SELECT
    STATUS_AREA,
    STATUS_AREA || '~' || STATUS_CODE STATUS_CODE,
    DESCR STATUS_DESC,
    NULL REASON_CODE,
    NULL REASON_DESC
    FROM
    PS_HRS_STS_TBL
    WHERE
    (STATUS_AREA = '1' OR STATUS_AREA = '3')
    UNION ALL
   /* Select all status reasons. */
   SELECT
    A.STATUS_AREA,
    A.STATUS_AREA || '~' || A.STATUS_CODE STATUS_CODE,
    A.DESCR STATUS_DESC,
    A.STATUS_AREA || '~' || A.STATUS_CODE || '~' || B.STATUS_REASON REASON_CODE,
    B.DESCR REASON_DESC
    FROM
   ```

```
PS_HRS_STS_TBL A, PS_HRS_STSRSN_TBL B
WHERE
B.STATUS_AREA = A.STATUS_AREA AND
B.STATUS_CODE = A.STATUS_CODE AND
(A.STATUS_AREA = '1' OR A.STATUS_AREA = '3')
```

2. Using a text editor, open the domainValues_Recruitment_Event_Reason_psft*Version*.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the Status Area, Status Code, Status Description, Reason Code, and Reason Description to the STATUS_AREA, STATUS_CODE, STATUS_DESC, REASON_CODE, REASON_DESC columns in the file respectively. The data must be copied starting from the sixth line. Use commas to separate the entries.
4. Map each Status Code (STATUS_CODE) and Reason Code (REASON_CODE) to one domain value for each of these domain columns:
   - W_EVENT_CODE
   - W_EVENT_DESC
   - W_SUB_STAGE_CODE
   - W_SUB_STAGE_DESC
   - W_STAGE_CODE
   - W_STAGE_DESC
   - W_REASON_CODE
   - W_REASON_DESC
   - W_REASON_TYPE_CODE
   - W_REASON_TYPE_DESC
   - EVENT_SEQ_NUM

   Status Description (STATUS_DESC) and Reason Description (REASON_DESC) are extracted to help you map the domain values.
5. Save and close the file.

7.2.4.17 How to Configure the file_pension_comp_flg.csv

1. Identify the Earnings Codes in your PeopleSoft source system by using the following SQL:

   ```
   SELECT DISTINCT ERNCD, DESCR FROM PS_EARNINGS_TBL ORDER BY 1
   ```

2. Using a text editor, open the file file_pension_comp_flg_psft.csv, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy over the ERNCD and DESCR column into the file. The data must be copied starting from the sixth line.
4. Relate a Pension Compensation Flag and a Compensation Flag based on the nature of the Earnings Code.
5. Save and close the file.

The SQL can be further tuned if all the Earnings Code of your Organization is of a particular type. For example, if all the Earnings Code that you are using starts with 'B' then the SQL can be modified as:

```
SELECT DISTINCT ERNCD, DESCR FROM PS_EARNINGS_TBL WHERE ERNCD LIKE 'B%' ORDER BY 1
```

7.2.4.18 How to Configure Compensation Flag and Pension Compensation Flag for Pay Type Dimension

The file 'file_pension_comp_flg_psft.csv' is used to populate the Compensation Flag and the Pension Compensation Flag. The value of the Compensation Flag and the Pension Compensation Flag is chosen based on the Input Earnings Code.

- Compensation Flag (COMP_FLG)
- Pension Compensation Flag (PENSION_COMP_FLG)

To modify Compensation Flag and Pension Compensation Flag:

1. Edit the file 'file_pension_comp_flg_psft.csv' in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles) using in a text editor.
2. Enter the COMP_FLG and PENSION_COMP_FLG value against the correct Earnings Code.
3. Save the file.

7.2.4.19 How to Configure the domainValues_perf_nrml_rating_psft.csv

This section explains how to configure the domainValues_perf_nrml_rating_psft.csv file. Using the PeopleSoft source system's performance functionality determines the configuration options for the domainValues_perf_nrml_rating_psft.csv file.

To configure the domainValues_perf_nrml_rating_psft.csv file:

1. Identify the Performance Ratings and Review Types in the PeopleSoft source system using the following SQL:

```
SELECT
 mdl.RATING_MODEL,
 mdl.DESCR RATING_MODEL_DESC,
 rtng.REVIEW_RATING,
 rtng.DESCR REVIEW_RATING_DESC,
 CASE WHEN pct.max_ep_rating > 0
      THEN round(100 * rtng.ep_rating / pct.max_ep_rating, 0)
      WHEN pct.max_review_points > 0
      THEN round(100 *rtng.review_points / pct.max_review_points, 0)
      WHEN pct.max_band_points > 0
      THEN round(100 * ((rtng.from_points + rtng.to_points) / 2) / pct.max_
band_points, 0)
      WHEN pct.max_eligibility_points > 0
      THEN round(100 * rtng.eligibility_points / pct.max_eligibility_points, 0)
      WHEN pct.max_rnk > 0
      THEN round(100 * SUM(1) OVER
            (PARTITION BY mdl.rating_model
            ORDER BY rtng.review_rating RANGE UNBOUNDED PRECEDING) /
            pct.max_rnk)
 END deflt_rating_pct
,null perf_band_code
FROM
 PS_RATING_MDL_TBL mdl
,PS_REVW_RATING_TBL rtng
,(SELECT
   rating_model
  ,effdt
  ,max(ep_rating) max_ep_rating
  ,max(review_points) max_review_points
  ,max(eligibility_points) max_eligibility_points
  ,max((from_points + to_points) / 2) max_band_points
  ,count(*) max_rnk
  FROM ps_revw_rating_tbl
  GROUP BY rating_model,effdt
 ) pct
WHERE mdl.RATING_MODEL = rtng.RATING_MODEL
AND rtng.EFFDT = mdl.EFFDT
AND mdl.rating_model = pct.rating_model
AND mdl.effdt = pct.effdt
AND mdl.EFFDT =
 (SELECT MAX(tmp.EFFDT)
 FROM PS_RATING_MDL_TBL tmp
 WHERE mdl.RATING_MODEL=tmp.RATING_MODEL)
ORDER BY 1,2,3
```

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the domainValues_perf_nrml_rating_psft.csv file in a text editor.
3. From the SQL query result, copy the RATING_MODEL, RATING_MODEL_DESC, REVIEW_RATING, and REVIEW_RATING_DESC data into the domainValues_perf_nrml_rating_psft.csv file. Copy data starting from the sixth line.
4. Map each RATING_MODEL, RATING_MODEL_DESC, REVIEW_RATING and REVIEW_RATING_DESC data to one NORMALIZED_RATING or to one PERF_BAND_CODE domain value.
   - The NORMALIZED_RATING values are from 0 to 100 (percent).
   - The PERF_BAND_CODE values should correspond to the PERF_BAND_CODE column in the $PMServer\srcfiles\file_perf_band.csv file (for example, in *INFA_HOME*\server\infa_shared\SrcFiles).
5. Save and close the file.

Note: You must perform this additional DAC parameter configuration.

To perform DAC task parameter configurations:

1. In DAC, query back the task name SDE_PSFT_PersistedStage_WorkforceEvent_Performance.
2. Set the $$APP_REVIEW_EXCL_STATUS parameter to exclude Review statuses (that is, any transaction that is not completed), the default value, for example, 'CA','TA', which is handled as an IN statement in the mapping SQL.
3. Set the $$APP_REVIEW_TYPE parameter to the Review type to include (only one type, for example, Annual Review is supported) the default value, for example, 'K0ANNUAL'.

Note: To prevent Informatica mapping failure, include the DAC variable values in single-quotes.

7.2.4.20 How to Configure the domainValues_Wrkfc_IntlAssign_psft.csv

To configure the domainValues_Wrkfc_IntlAssign_psft.csv file.

1. Identify the International Assignment Types and Home/Host Classes in your PeopleSoft source system using the following SQL (using these SQL statements depends on the configuration that you require, see in this section the note under step 4 before running these SQL statements):

   ```
   SELECT
    NULL PAYGROUP,
    ASSGN_TYPE INTL_ASSGN_TYPE_CODE,
    FIELDVALUE HOME_HOST_CLASS_CODE,
    CASE WHEN FIELDVALUE = 'H'
     THEN 'N'
     ELSE 'Y'
    END W_COMP_OWNER_FLG
   FROM
    PS_ASSGN_TYPE_TBL ASG,
    PSXLATITEM  FLG
   WHERE FLG.FIELDNAME LIKE 'HOME_HOST_CLASS'
   ORDER BY 1,2,3;
   SELECT
    PAYGROUP  PAYGROUP,
    NULL INTL_ASSGN_TYPE_CODE,
    FIELDVALUE HOME_HOST_CLASS_CODE,
    CASE WHEN FIELDVALUE = 'H'
     THEN 'N'
     ELSE 'Y'
    END W_COMP_OWNER_FLG
   FROM
    PS_PAYGROUP_TBL PYG,
    PSXLATITEM FLG
   WHERE FLG.FIELDNAME LIKE 'HOME_HOST_CLASS'
   ORDER BY 1,2,3;
   ```

2. Open the domainValues_Wrkfc_IntlAssign_psft.csv file using a text editor in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

3. From the SQL query result, copy the INTL_ASSGN_TYPE_CODE and HOME_HOST_CLASS_CODE data into the domainValues_Wrkfc_IntlAssign_psft.csv file. Copy data starting from the sixth line.

4. Map each INTL_ASSGN_TYPE_CODE and HOME_HOST_CLASS value to one W_COMP_OWNER_FLG domain value (Y/N).

   > **Note:** Optionally, if you use a dummy Payroll in order to identify Employees on International Assignment, then define the PAYGROUP value to one W_COMP_OWNER_FLG domain value (Y/N).

7.2.4.21 How to Configure the domainValues_AbsenceEvent_Status_psft.csv

To configure the domainValues_AbsenceEvent_Status_psft.csv file.

1. Approval Status in the PeopleSoft source system is 'Y/N', with the edited field's default value set to 'N'. You can also identify Approval Status values using the following SQL:

   ```
   SELECT DISTINCT EMPLOYER_APPROVED FROM PS_ABSENCE_HIST
   ```

You will also find the (Y/N) values.

> **Note:** For later use, a null value is added for absence status to include Requested or in progress.

2. From the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles), open the domainValues_AbsenceEvent_Status_psft.csv file in Microsoft Excel; in the file, verify if the information provided in the table is available, starting from the fifth line.

APPROVAL_STATUS	ABSENCE_STATUS_CODE	ABSENCE_STATUS_NAME
Y	APPROVED	Approved
N	UNAPPROVED	Not Approved
	REQUESTED	Requested or In Progress

7.2.4.22 How to Configure the domainValues_Learning_Enrollment_Status_psftVersion.csv

This section explains how to configure the domainValues_Learning_Enrollment_Status_psft*Version*.csv.

1. Identify the Learning Enrollment Statuses in your PeopleSoft source system by using the following SQL:

```
SELECT
A.FIELDVALUE STATUS_CODE,
A.XLATLONGNAME STATUS_NAME
FROM
PSXLATITEM A LEFT OUTER JOIN PSXLATITEMLANG B ON
A.FIELDNAME=B.FIELDNAME AND A.FIELDVALUE=B.FIELDVALUE AND B.LANGUAGE_CD='ENG'
WHERE
A.FIELDNAME IN ('LM_STTS') AND
A.EFFDT = (SELECT MAX(C.EFFDT) FROM PSXLATITEM C WHERE C.FIELDNAME =
A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE )
```

2. Using a text editor, open the domainValues_Learning_Enrollment_Status_psft*Version*.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the Status Code and Status Name to the STATUS_CODE and STATUS_NAME columns in the file respectively. The data must be copied starting from the sixth line. Use commas to separate the entries.
4. Map each Status Code to one domain value. Status Names are also extracted with Status Codes to help you map the domain values.
5. Save and close the file.

7.2.4.23 How to Build an Execution Plan for Running HR and GL Subject Areas Together

Human Resources Analytics requires that some financial fact tables from Oracle BI Applications Financial Analytics be populated in order to display default dashboards for "HR - Performance." Although DAC execution plans are shipped by default for Oracle EBS and Universal adaptors, PeopleSoft HR Analytics customers need to build an execution plan to deploy the financial metrics in HR Analytics, because PeopleSoft FSCM transaction tables do not exist in Human Resources Analytics.

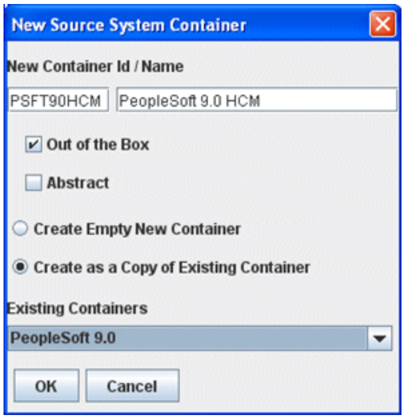
The following procedure shows an example of how to build a composite execution plan to deploy financial-related metrics in HR Analytics using PeopleSoft 9.0 as a source. Be sure to replace "9.0" with "8.9" or "9.1," as appropriate for your PeopleSoft OLTP version.

To build an execution plan to deploy financial-related metrics in HR Analytics:

1. In DAC, select **File**, and then select **New Source System Container**. In the New Source System Container dialog, create a custom container called "PeopleSoft 9.0 HCM" from the existing container "PeopleSoft 9.0," and then click **OK**.

   Figure 7–1 shows the New Source System Container dialog for the PeopleSoft 9.0 HCM container.

Figure 7–1 New Source System Container Dialog for PeopleSoft 9.0 HCM Container



New Container Id / Name

PSFT90HCM | PeopleSoft 9.0 HCM

☑ Out of the Box

☐ Abstract

○ Create Empty New Container

● Create as a Copy of Existing Container

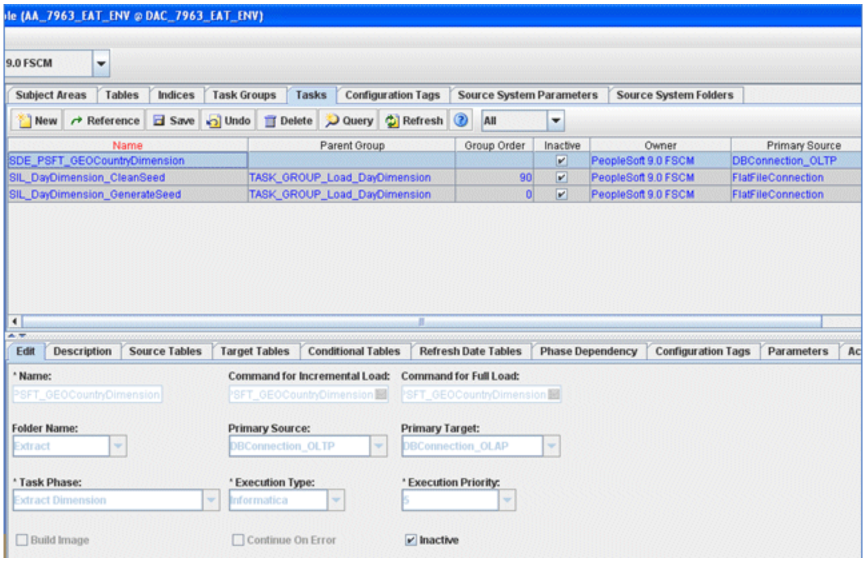
Existing Containers

PeopleSoft 9.0

2. Similar to step 1, create another new custom container called "PeopleSoft 9.0 FSCM" from the existing container "PeopleSoft 9.0," and then click **OK**.
3. In Design view, select the new container "PeopleSoft 9.0 FSCM" and then select the Tasks tab. Query for the following three tasks and deactivate them:
   - SDE_PSFT_GEOCountryDimension
   - SIL_DayDimension_GenerateSeed
   - SIL_DayDimension_CleanSeed

   Figure 7–2 shows these three tasks.

Figure 7–2 FSCM Tasks in DAC



Name	Parent Group	Group Order	Inactive	Owner	Primary Source
SDE_PSFT_GEOCountryDimension			☑	PeopleSoft 9.0 FSCM	DBConnection_OLTP
SIL_DayDimension_CleanSeed	TASK_GROUP_Load_DayDimension	90	☑	PeopleSoft 9.0 FSCM	FlatFileConnection
SIL_DayDimension_GenerateSeed	TASK_GROUP_Load_DayDimension	0	☑	PeopleSoft 9.0 FSCM	FlatFileConnection

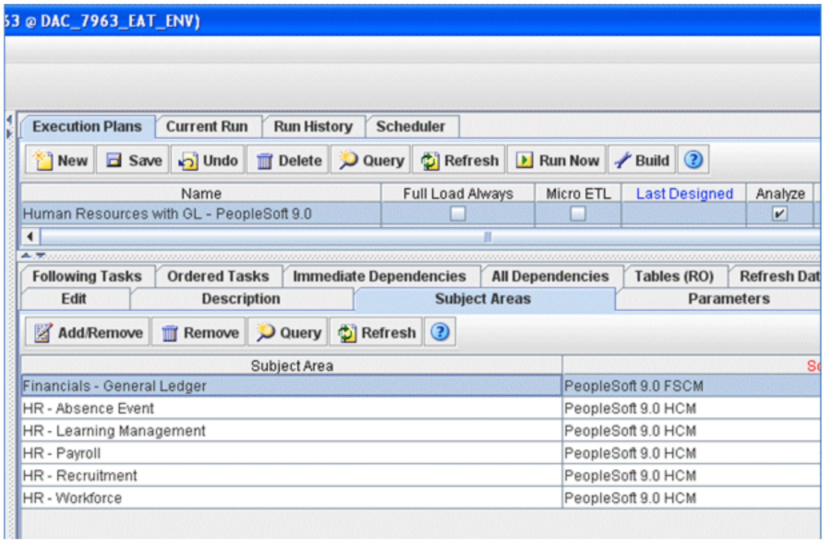
* Name: PSFT_GEOCountryDimension	Command for Incremental Load: PSFT_GEOCountryDimension	Command for Full Load: PSFT_GEOCountryDimension
Folder Name: Extract	Primary Source: DBConnection_OLTP	Primary Target: DBConnection_OLAP
* Task Phase: Extract Dimension	* Execution Type: Informatica	* Execution Priority: 5
☐ Build Image	☐ Continue On Error	☑ Inactive

4. Create a new Execution plan called "Human Resources with GL - PeopleSoft 9.0." Choose subject areas as follows:

   - Financials - General Ledger
   - HR - Absence Event
   - HR - Learning Management
   - HR - Payroll
   - HR - Recruitment
   - HR - Workforce

   Figure 7–3 shows the Subject Areas subtab in DAC for the "Human Resources with GL - PeopleSoft 9.0" execution plan.

Figure 7–3 Subject Areas for the "Human Resources with GL - PeopleSoft 9.0" Execution Plan



Name	Full Load Always	Micro ETL	Last Designed	Analyze
Human Resources with GL - PeopleSoft 9.0	☐	☐		☑

Subject Area	Sc
Financials - General Ledger	PeopleSoft 9.0 FSCM
HR - Absence Event	PeopleSoft 9.0 HCM
HR - Learning Management	PeopleSoft 9.0 HCM
HR - Payroll	PeopleSoft 9.0 HCM
HR - Recruitment	PeopleSoft 9.0 HCM
HR - Workforce	PeopleSoft 9.0 HCM

Note the following:

- The "Financials - General Ledger" subject area is the only subject area from the container "PeopleSoft 9.0 FSCM." All the other subject areas are from the container "PeopleSoft 9.0 HCM."
- If "HR - Learning Management" is not a subject area of interest, do not select it. Because e-Learning comes in a different physical schema for PeopleSoft OLTP, if you do not have e-Learning implemented at the OLTP level, you should not select the Learning subject area. If you selected Learning, it is expected that you have e-Learning implemented, and DAC will need the physical connection for your e-Learning schema (commonly known as ELM).

5. Select the Parameters subtab in the bottom pane. Then, generate one copy of the parameters for each container. Set the values of the parameters as shown in Table 7–8.

Table 7–18 Parameter Values for Human Resources with GL - PeopleSoft 9.0 Execution Plan

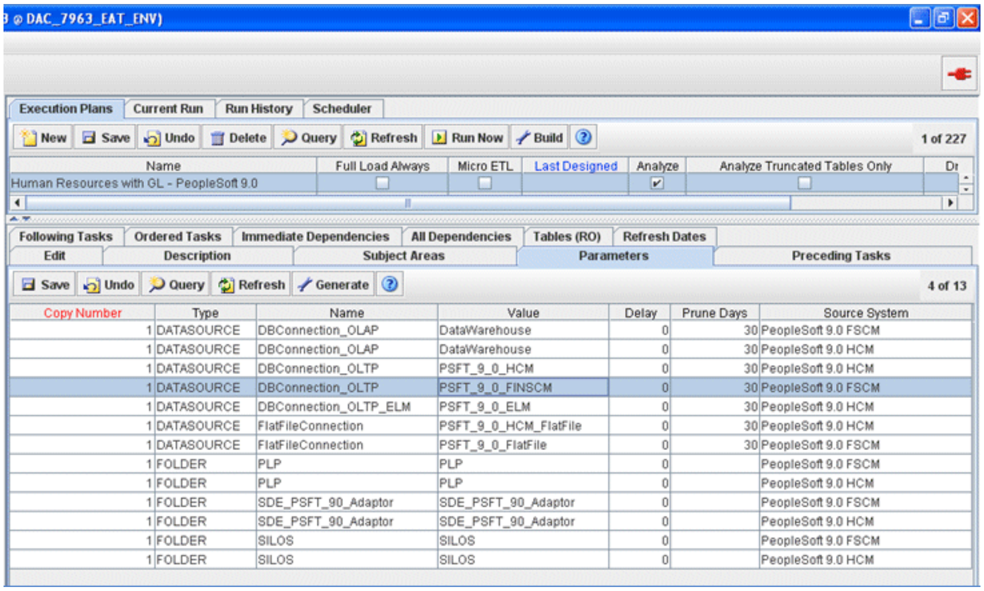
Type	Name	Value	Delay	Prune Days	Source System
DATASOURCE	DBConnection_OLAP	DataWarehouse	0	30	PeopleSoft 9.0 FSCM
DATASOURCE	DBConnection_OLAP	DataWarehouse	0	30	PeopleSoft 9.0 HCM
DATASOURCE	DBConnection_OLTP	PSFT_9_0_HCM	0	30	PeopleSoft 9.0 HCM
DATASOURCE	DBConnection_OLTP	PSFT_9_0_FINSCM	0	30	PeopleSoft 9.0 FSCM
DATASOURCE	DBConnnection_OLTP_ELM	PSFT_9_0_ELM	0	30	PeopleSoft 9.0 HCM
DATASOURCE	FlatFileConnection	PSFT_9_0_HCM_FlatFile	0	30	PeopleSoft 9.0 HCM
DATASOURCE	FlatFileConnection	PSFT_9_0_FlatFile	0	30	PeopleSoft 9.0 FSCM
FOLDER	PLP	PLP	0	-	PeopleSoft 9.0 FSCM
FOLDER	PLP	PLP	0	-	PeopleSoft 9.0 HCM
FOLDER	SDE_PSFT_90_Adaptor	SDE_PSFT_90_Adaptor	0	-	PeopleSoft 9.0 FSCM
FOLDER	SDE_PSFT_90_Adaptor	SDE_PSFT_90_Adaptor	0	-	PeopleSoft 9.0 HCM
FOLDER	SILOS	SILOS	0	-	PeopleSoft 9.0 FSCM
FOLDER	SILOS	SILOS	0	-	PeopleSoft 9.0 HCM

Note the following:

- Be sure to choose the physical data source PSFT_9_0_FINSCM for the logical data source DBConnection_OLTP when the container is "PeopleSoft 9.0 FSCM," and choose PSFT_9_0_HCM for the same logical data source when the container is "PeopleSoft 9.0 HCM."
- The value of "Prune Days" is set by default and can be changed if needed.
- The physical Informatica folders are set to the appropriate corresponding logical folder, for both containers.
- You should only provide the physical data source PSDT_9_0_ELM against the logical data source DBConnection_OLTP_ELM if you selected the Learning subject area in the previous step.

Figure 7–4 shows the Parameters subtab in DAC.

Figure 7–4 Parameters Subtab for the "Human Resources with GL - PeopleSoft 9.0" Execution Plan



Name	Full Load Always	Micro ETL	Last Designed	Analyze	Analyze Truncated Tables Only	Dr
Human Resources with GL - PeopleSoft 9.0	☐	☐		☑	☐	

Copy Number	Type	Name	Value	Delay	Prune Days	Source System
1	DATASOURCE	DBConnection_OLAP	DataWarehouse	0	30	PeopleSoft 9.0 FSCM
1	DATASOURCE	DBConnection_OLAP	DataWarehouse	0	30	PeopleSoft 9.0 HCM
1	DATASOURCE	DBConnection_OLTP	PSFT_9_0_HCM	0	30	PeopleSoft 9.0 HCM
1	DATASOURCE	DBConnection_OLTP	PSFT_9_0_FINSCM	0	30	PeopleSoft 9.0 FSCM
1	DATASOURCE	DBConnection_OLTP_ELM	PSFT_9_0_ELM	0	30	PeopleSoft 9.0 HCM
1	DATASOURCE	FlatFileConnection	PSFT_9_0_HCM_FlatFile	0	30	PeopleSoft 9.0 HCM
1	DATASOURCE	FlatFileConnection	PSFT_9_0_FlatFile	0	30	PeopleSoft 9.0 FSCM
1	FOLDER	PLP	PLP	0		PeopleSoft 9.0 FSCM
1	FOLDER	PLP	PLP	0		PeopleSoft 9.0 HCM
1	FOLDER	SDE_PSFT_90_Adaptor	SDE_PSFT_90_Adaptor	0		PeopleSoft 9.0 FSCM
1	FOLDER	SDE_PSFT_90_Adaptor	SDE_PSFT_90_Adaptor	0		PeopleSoft 9.0 HCM
1	FOLDER	SILOS	SILOS	0		PeopleSoft 9.0 FSCM
1	FOLDER	SILOS	SILOS	0		PeopleSoft 9.0 HCM

6. Build this execution plan.

7. Select Set up, then select the Physical Data Sources tab. Configure the connections for the following physical data sources:

   - DataWarehouse
   - PSFT_9_0_HCM
   - PSFT_9_0_ELM (if you use e-Learning)
   - PSFT_9_0_FINSCM

8. Execute the "Human Resources with GL - PeopleSoft 9.0" execution plan.

7.2.5 Configuration Steps for Oracle HR Analytics for Universal

See technical note 1269240.1 on My Oracle Support for detailed information about using Oracle HR Analytics with the Universal adapter.

7.3 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle HR Analytics.

- Section 7.3.1, "Configuration Steps for Oracle HR Analytics for All Source Systems"
- Section 7.3.2, "Configuration Steps for Oracle HR Analytics for Oracle EBS"
- Section 7.3.3, "Configuration Steps for Oracle HR Analytics for PeopleSoft"

7.3.1 Configuration Steps for Oracle HR Analytics for All Source Systems

This section contains configuration steps that apply to all source systems. It contains the following topics:

- Section 7.3.1.1, "How to Aggregate the Payroll Table for Oracle HR Analytics"
- Section 7.3.1.2, "Resolving Dimensions for Every Snapshot Date"

- Section 7.3.1.3, "About Oracle HR Analytics DAC Parameters"
- Section 7.3.1.4, "About Future-Dated Security for Oracle HR Analytics"

7.3.1.1 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 ETL 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 ETL run. Oracle HR Analytics minimizes the incremental aggregation effort, by modifying the aggregate table incrementally as the base table is updated. Oracle BI Applications looks for new records in the base table during the incremental ETL. This process is done in two steps:

1. There are new records in the W_PAYROLL_A table, which are inserted after the last ETL 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 DAC.

To load the Payroll aggregate table (W_PAYROLL_A):

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab, and select the Task PLP_PayrollAggregate_Load.
3. Display the Parameters subtab, and add a parameter with the name $$GRAIN.
4. Depending upon the aggregation requirement, define the value as one of the following: 'DAY', 'WEEK', 'MONTH', 'QUARTER' or 'YEAR'.
5. Select Static as the parameter type.
6. Save the task.

7.3.1.2 Resolving Dimensions for Every Snapshot Date

For dimensions that keep history, it is important to look up the dimension values for each snapshot date in the fact. To achieve this, set the Dimension Type2 Flag parameters in DAC for the task PLP_WorkforceEventFact_Month. Setting these parameters enables the workforce snapshot fact table to correctly resolve the type 2 slowly-changing dimensional foreign key.

By default, the Type2 Flag parameters are set to N, which means that dimensions with history may not be perfectly in sync with the fact on every snapshot date. Change this setting to Y for each dimension where the WIDs need to be resolved as of the snapshot date. There might be a small performance impact, particularly with a full load, if all the flags are set to Y, because the ETL will be performing many additional lookups for every snapshot.

The following list shows the parameters defined in the task PLP_WorkforceEventFact_Month in DAC. These parameters have a default value of N. Set these parameters to Y only for dimensions that require WIDs to be resolved as of the snapshot date.

- $$EMPLOYEE_TYPE2_FLG
- $$ORGANIZATION_TYPE2_FLG
- $$JOB_TYPE2_FLG
- $$HR_POSITION_TYPE2_FLG
- $$LOCATION_TYPE2_FLG

7.3.1.3 About Oracle HR Analytics DAC Parameters

The DAC parameters described in this section are specific to Oracle HR Analytics. The values you set for these parameters largely affect the data in your Data Warehouse and will ultimately drive your reporting experiences.

- $$HR_WRKFC_EXTRACT_DATE

  Set this date parameter to a date from which you need to extract the workforce analytics data.

- $$HR_ABSNC_EXTRACT_DATE

  Set this date parameter to a date from which you need to extract the absence data for the workforce. This date parameter value should be greater than or equal to the value of $$HR_WRKFC_EXTRACT_DATE.

- $$HR_WRKFC_SNAPSHOT_DT

  Set this date parameter to the date from which monthly snapshots need to be aggregated from the workforce event fact. Setting a very early date causes a large number of snapshots. Use discretion when choosing a very early date, because doing so is likely to affect ETL performance. Because the snapshots are typically taken on a monthly basis, it is required that you enter a date against this parameter that is at least the "first day" of a month, or even better, the "first day" of a year (Gregorian). Note that this date parameter value should be greater than or equal to the value of $$HR_WRKFC_EXTRACT_DATE.

7.3.1.4 About Future-Dated Security for Oracle HR Analytics

HR Analytics loads future-dated transactions into the data warehouse. Future-dated transactions can be excluded from user querying. The following list describes where and how future-dated security is implemented in the HR subject areas:

- Fact - HR - Event Information (Workforce) and Fact - HR - Operation (Workforce) are future-data secured by the session level variable, HR_MAX_EFFECTIVE_DT, which defaults to the value CURRENT_DATE in the default configuration.

- Fact - HR - Recruitment Event Information has future-dated security by CURRENT_DATE in the logical table sources.
- The following fact tables are not restricted by future-dated security, and users can query future-dated data:
  - Fact - HR - Absence Event
  - Fact - HR - Payroll
  - Fact - HR - Learning Enrollment Events
  - Fact - HR - Learning Enrollment and Completion

7.3.2 Configuration Steps for Oracle HR Analytics for Oracle EBS

This section contains configuration steps that apply to Oracle EBS. It contains the following topics:

- Section 7.3.2.1, "About Incremental Processing for Oracle EBS"
- Section 7.3.2.2, "Configuring Original Job Requisition Status"
- Section 7.3.2.3, "About Job Requisition Metrics in Oracle EBS"
- Section 7.3.2.4, "Configuring Employee Assignment and Applicant Assignment Correlation"
- Section 7.3.2.5, "About Oracle HR Analytics DAC Parameters and Configuration Tags for Oracle EBS"
- Section 7.3.2.6, "About Duration Unit Conversion to Hours"

7.3.2.1 About Incremental Processing for Oracle EBS

Incremental processing for Oracle EBS adaptors supports refreshing of all new or changed data, except where the data has been purged. This includes new records, corrections, date-tracked updates, and date-tracked deletes.

There are separate mappings to detect and process purges.

Note: This section is provided for informational purposes only. No configuration steps are required for this topic.

7.3.2.2 Configuring Original Job Requisition Status

In Oracle EBS, the Job Requisition status is not preserved as historical information in the OLTP. Therefore, as a Job Requisition status changes, for example from Drafted to Approved to Open to Closed, the OLTP saves only the last status.

The Job Requisition Open event is a significant event because several metrics depend on it. Therefore, you must track this event by configuring the original Job Requisition status event, which occurs on the Job Requisition start date. For example, if the latest status is "Closed," then you can deduce that at one point it had a status of "Open." Therefore, you should map the original status to "Open." However, if the latest status is "Approval Denied," then the requisition was never opened. Therefore, you should map the original status of the Job Requisition to another value, such as "Requested."

Use the file_job_requisition_status_original_ora12.csv file to map the original status of the Job Requisition to the event. An example of how to map this event is provided in the following configuration steps.

To configure the original Job Requisition Status:

1. Identify the Job Requisition statuses in your Oracle EBS source system by using the following SQL:

   ```
   SELECT
     HR_STANDARD_LOOKUPS.LOOKUP_CODE,
     HR_STANDARD_LOOKUPS.MEANING
   FROM
     HR_STANDARD_LOOKUPS
   WHERE
     HR_STANDARD_LOOKUPS.LOOKUP_TYPE = 'VACANCY_STATUS'
   ```

2. Using a text editor, open the file_job_requisition_status_original_ora12.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the Lookup Code and Meaning to the REQUISITION_CURRENT_STATUS_CODE and REQUISITION_CURRENT_STATUS_NAME columns in the file respectively. The data must be copied starting from the sixth line. Use commas to separate the entries.
4. Map each REQUISITION_CURRENT_STATUS_CODE to one REQUISITION_ORIGINAL_STATUS_CODE.

   REQUISITION_ORIGINAL_STATUS_CODE is used primarily to determine if a job requisition had a status of OPEN at an earlier date. For example, if the REQUISITION_CURRENT_STATUS_CODE of a job requisition is CLOSED, it may mean that the job requisition was OPEN on an earlier date. In this case, the original requisition status can be classified as APPROVED. The APPROVED status can be mapped to RQSTN_OPEN as W_EVENT_CODE, W_SUB_STAGE_CODE, and W_STAGE_CODE in the domainValues_Recruitment_Event_Reason_ora*Version*.csv.

   If the current Job Requisition status is REJECTED, it may mean that this job requisition previously had a status of PENDING on an earlier date and was never in OPEN status. In this case, the original requisition status can be classified as PENDING instead of OPEN. The PENDING status can be mapped to RQSTN_APPROVAL_PENDING as W_EVENT_CODE, W_SUB_STAGE_CODE and RQSTN_PENDING as stage code.

   All values of REQUISITION_ORIGINAL_STATUS_CODE should exist as a value in REQUISITION_CURRENT_STATUS_CODE.
5. Save and close the file.

7.3.2.3 About Job Requisition Metrics in Oracle EBS

Job Requisition in EBS includes the following metrics:

- Number of Openings
- Budgeted Headcount
- Budgeted FTE

In iRecruitment, users enter Job Requisition details, which include Number of Openings. In the ETL run, Number of Openings is mapped to the BUDGET_MEASUREMENT_VALUE field. By default, the NUMBER_OF_OPENINGS field is populated by what is configured at the organization level. The Number of Openings field is not visible in the application by default; however, the application can be customized so that the field is visible.

In iRecruitment, the field with label Number of Openings is mapped to the BUDGET_MEASUREMENT_VALUE column of the PER_ALL_VACANCIES table. By default, the NUMBER_OF_OPENINGS column of the PER_ALL_VACANCIES table is populated by what is configured at the organization level. The Number of Openings field is not visible in the application by default; however, the application can be customized so that the field is visible.

Three parameters have been provided to allow flexibility when setting up these metrics. The following table describes the parameters and their default values, which you can modify to suit your implementation requirements.

Task Name	Parameter Name	Default Values
SDE_ORA_JobRequisitionDimension	$$BUDGETED_FTE_SOURCE_COL	CASE WHEN PER_ALL_VACANCIES.BUDGET_MEASUREMENT_TYPE = 'FTE' THEN PER_ALL_VACANCIES.BUDGET_MEASUREMENT_VALUE ELSE NULL END
SDE_ORA_JobRequisitionDimension	$$BUDGETED_HEADCOUNT_SOURCE_COL	CASE WHEN PER_ALL_VACANCIES.BUDGET_MEASUREMENT_TYPE = 'HEAD' THEN PER_ALL_VACANCIES.BUDGET_MEASUREMENT_VALUE ELSE PER_ALL_VACANCIES.NUMBER_OF_OPENINGS END
SDE_ORA_JobRequisitionDimension	$$NUMBER_OF_OPENINGS_SOURCE_COL	PER_ALL_VACANCIES.NUMBER_OF_OPENINGS
SDE_ORA_JobRequisitionEventFact	$$BUDGETED_FTE_SOURCE_COL	CASE WHEN PER_ALL_VACANCIES.BUDGET_MEASUREMENT_TYPE = 'FTE' THEN PER_ALL_VACANCIES.BUDGET_MEASUREMENT_VALUE ELSE NULL END
SDE_ORA_JobRequisitionEventFact	$$BUDGETED_HEADCOUNT_SOURCE_COL	CASE WHEN PER_ALL_VACANCIES.BUDGET_MEASUREMENT_TYPE = 'HEAD' THEN PER_ALL_VACANCIES.BUDGET_MEASUREMENT_VALUE ELSE PER_ALL_VACANCIES.NUMBER_OF_OPENINGS END
SDE_ORA_JobRequisitionEventFact	$$NUMBER_OF_OPENINGS_SOURCE_COL	PER_ALL_VACANCIES.NUMBER_OF_OPENINGS

7.3.2.4 Configuring Employee Assignment and Applicant Assignment Correlation

Recruitment analytics is designed to correlate the applicant with that of the employee getting hired in the job. This is done in the mapping 'SDE_ORA_ApplicantEventFact_EmpAplAssignments.' By default, the correlation is based only on the PERSON_ID column. However, the logic for this correlation can vary based on customer implementations and might require checks on additional columns to make it more reliable.

The following two parameters have been provided to allow flexibility when setting up the correlation between the employee assignment and the applicant assignment:

- $$EMP_APL_CORR_VACANCY_ID_MATCH_REQD

  If the customer implementation is such that the VACANCY_ID forms the basis of the correlation between employee assignment and the applicant assignment, then this parameter must be set to 'Y.' By default, the parameter value is set to 'N.'

- $$EMP_APL_CORR_APPLICATION_ID_MATCH_REQD

  If the customer implementation is such that the APPLICATION_ID forms the basis of the correlation between employee assignment and the applicant assignment, then this parameter must be set to 'Y.' By default, the parameter value is set to 'N.'

If required, customers can set both parameter values to 'Y,' so that both conditions are considered for the correlation.

Note that both of these conditions for correlating employee and applicant assignments are in addition to the existing condition on PERSON_ID.

To configure the Employee and Applicant Assignment Correlation:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab, and query for task 'SDE_ORA_ApplicantEventFact_EmpAplAssignments.'
3. Go to the Parameters subtab and set the appropriate values for the following parameters:
   - $$EMP_APL_CORR_VACANCY_ID_MATCH_REQD
   - $$EMP_APL_CORR_APPLICATION_ID_MATCH_REQD
4. Choose the task and push the changes to all the referenced containers.

7.3.2.5 About Oracle HR Analytics DAC Parameters and Configuration Tags for Oracle EBS

Be aware of the following DAC parameter for Oracle HR Analytics with Oracle EBS source systems:

- $$HR_WRKFC_ADJ_SERVICE_DATE

  This parameter is used to determine the period of service for workers. Set this to 'Y' to start calculating period of service/placement from rehire date, and 'N' to start calculating period of service/placement from original hire date.

In addition, be aware of the following Oracle EBS-specific configuration tags in the DAC repository. Include these tags as needed, depending on the Execution plan you are running.

- Human Resources - Collect FTE

  This configuration tag enables you to collect the Full Time Equivalent (FTE) values for workers in HR-Workforce analytics. If the tag is inactivated (disabled), FTE Values are not collected.

- Human Resources - Collect Performance Ratings

  This configuration tag enables to collect the Performance Ratings for workers in HR-Workforce analytics. If the tag is inactivated (disabled), Performance Ratings are not collected.

- Human Resources - Load User Flex Data

  This configuration tag enables loading of user customized Oracle E-Business Suite Flexfield data to the warehouse dimension tables (such as JOB_CODE in W_JOB_D). For more information, see Section 7.2.3, "Configuration Steps for Flexflelds." You must include this tag to see Flexfield columns.

7.3.2.6 About Duration Unit Conversion to Hours

The file_duration_units_conversion_to_hours.csv file affects the calculation and conversion of assignment working frequency/amount from hourly, daily, weekly, monthly, quarterly or yearly amounts to hourly, daily, weekly, monthly, quarterly or yearly amounts. The conversation affects the table column W_WRKFC_EVT_MONTH_F.STD_HOURS_MONTH.

This file sets the conversion factor from minutes to hours as 1/60 = 0.0167, hours to hours as 1, days to hours as 8, and so on. In most cases, you can leave the values unchanged. However, if your system uses different conversions (for example, days to hours as 8.5 or 9), you must change the values.

7.3.3 Configuration Steps for Oracle HR Analytics for PeopleSoft

This section contains configuration steps that apply to PeopleSoft. It contains the following topic:

- Section 7.3.3.1, "About Incremental Processing for PeopleSoft"
- Section 7.3.3.2, "How to Tune Performance for the Payroll Table for PeopleSoft HCM HR Analytics"
- Section 7.3.3.3, "About Oracle HR Analytics DAC Parameters for PeopleSoft"

7.3.3.1 About Incremental Processing for PeopleSoft

This section describes the default behavior for People Soft incremental processing. Although most data is refreshed, there are some restrictions due to no last update timestamps on some base tables. These restrictions are as follows:

- Job data - no restrictions
- Appraisals - only new records or changes to the most recent record are processed
- International Assignments - only changes to the most recent record are processed

There are separate mappings to detect and process purges.

7.3.3.2 How to Tune Performance for the Payroll Table for PeopleSoft HCM HR Analytics

The Payroll mappings are designed to take care of the Payroll Adjustment and the Payroll Reversal process. These are complicated processes and normally take a reasonable amount of time to complete. Depending on your requirements, there are two ways you can achieve better overall performance in the Payroll Analytics. These techniques are as follows:

- Deactivate Adjustments and Reversals, if not required, for a given pay period.
- Defer calculating Adjustments and Reversals for some time, until needed.

To deactivate Adjustment and Reversal calculations for Payroll:

If you decide that there will be no Payroll Adjustment and Reversal for a pay period, then you can choose to deactivate those tasks in DAC.

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Select the Tasks tab.
3. Select the Inactive check the box for the following tasks:
   - SDE_PSFT_PayrollFact_Deductions_Adjustments_Reversals
   - SDE_PSFT_PayrollFact_Earnings_Adjustments_Reversals
   - SDE_PSFT_PayrollFact_EarningsOther_Adjustments_Reversals
   - SDE_PSFT_PayrollFact_Total_Adjustments_Reversals

- SDE_PSFT_PayrollFact_Taxes_Adjustments_Reversals

4. Save the tasks.
5. Assemble the Subject Area HR- Payroll and build the Execution Plans that contain this Subject Area.

To defer Adjustment and Reversal calculations for Payroll:

1. Deactivate Adjustment and Reversal calculations for Payroll and reassemble the Subject Area Payroll and rebuild the Execution Plans that contain this Subject Area.
2. When you choose to run these again, reactivate the tasks, and reassemble Subject Area (Payroll) and Execution Plan.
3. Adjust the parameter $$LAST_EXTRACT_DATE so that it goes back to the date (preferably one day earlier) since when you decided not to run these tasks.

7.3.3.3 About Oracle HR Analytics DAC Parameters for PeopleSoft

You can use the Oracle HR Analytics DAC parameters for PeopleSoft to customize International Assignment and Appraisal extractions. To force changes to any appraisal or international assignment to be picked up, you can customize the incremental mappings. The filter restricting records to the most recent one can be removed. Doing so increases the processing time for incremental load.

The Oracle HR Analytics DAC parameters for PeopleSoft are as follows:

- $$APP_REVIEW_EXCL_STATUS

  Set this parameter to exclude statuses (such as, anything that is not completed). The default example is 'CA','TA', which is handled as an IN statement in the mapping SQL.

- $$APP_REVIEW_TYPE

  Set this parameter to the Review type to include. Only one type is supported, such as Annual Review. The default example is 'K0ANNUAL.'

8 Configuring Oracle Sales Analytics

This section describes how to configure Oracle Sales Analytics used with a Siebel CRM source system. It contains the following topics:

- Section 8.1, "Overview of Oracle Sales Analytics"
- Section 8.2, "Configuration Required Before A Full Load"
- Section 8.3, "Configuration Steps for Controlling Your Data Set"
- Section 8.4, "Configuring Usage Accelerator"

8.1 Overview of Oracle Sales Analytics

Oracle Sales Analytics includes a wide range of tasks, analytics and engagement tactics that maximize the value of the customer relationship and contribute to sustainable revenue growth. Oracle Sales Analytics also includes Usage Accelerator for Siebel Sales Applications.

8.2 Configuration Required Before A Full Load

This section contains configuration steps that you need to perform on Oracle Sales Analytics before you do a full data load. It contains the following topics:

- Section 8.2.1, "How to Deploy Stored Procedures"

8.2.1 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 improve the performance of the database. You deploy stored procedures by copying the stored procedure files from your Oracle BI Analytics 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. Navigate to the *MW_HOME*\biapps\dwrep\Informatica\Stored_Procedure_Scripts folder.
2. Open one of the folders based your database platform, and copy the source code from the file FIND_AUDIT_VALUES.sql into the target data warehouse schema.
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.

8.3 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle Sales Analytics. It contains the following topic:

- Section 8.3.1, "How to Configure the Snapshot Frequency in the Pipeline Fact"

8.3.1 How to Configure the Snapshot Frequency in the Pipeline Fact

Even though the snapshot of Pipeline is captured every time ETL is run, you can choose to persist daily snapshots, or weekly snapshots or monthly snapshots or even yearly snapshots. There is one parameter 'GRAIN' you need to modify to configure for this.

The GRAIN parameter has a preconfigured value of 'DAY'. The possible values for the GRAIN parameter are:

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

To configure the snapshot frequency in the Pipeline fact:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab, and query for the SIL_PipelineFact task.
3. Display the Parameters subtab, and create a parameter called $$GRAIN, and enter an appropriate parameter value.

8.4 Configuring Usage Accelerator

This section explains how to configure Usage Accelerator. It contains the following topics:

- Section 8.4.1, "What is Usage Accelerator?"
- Section 8.4.2, "Configuring Usage Accelerator"
- Section 8.4.3, "About the Usage Accelerator Execution Plan"

8.4.1 What is Usage Accelerator?

Usage Accelerator enables sales organizations to measure and manage user adoption and effectiveness of Oracle's Siebel Sales. Every level of the sales organization benefits from the richer understanding of how each member of the team is contributing to the overall objective. Usage Accelerator provides a comprehensive set of management dashboards and prebuilt metrics that deliver rich information to help accelerate organizational usage, and drive alignment and increase the overall ROI of CRM implementations.

8.4.2 Configuring Usage Accelerator

This section provides an overview of configuring Usage Accelerator, and contains the following topics:

- Section 8.4.2.1, "Tracking of Targeted Users"
- Section 8.4.2.2, "Other Siebel Required Responsibilities"
- Section 8.4.2.3, "About Metric Definitions and Variables"
- Section 8.4.2.4, "Configuring Variables"

8.4.2.1 Tracking of Targeted Users

Usage Accelerator determines the users that you want to measure through the use of a new responsibility called 'Usage Accelerator – Tracking'. You must associate this new responsibility with each user in the operational Siebel application. Choosing which users you want to measure enables you to optimize the storage space required for the application data.

8.4.2.2 Other Siebel Required Responsibilities

Usage Accelerator introduces six new responsibilities. Three of the responsibilities will be used by all organizations except for Siebel Financial Services customers. Siebel Financial Services customers use three responsibilities that are targeted specifically for their user groups. The responsibilities determine which dashboards a user will see when they use Usage Accelerator. These responsibilities need to be assigned to users in the operational Siebel CRM application. The responsibilities are:

Table 8–1 User Accelerator Responsibilities

Usage Accelerator Responsibilities	Usage Accelerator Responsibilities (for Siebel Financial Services)
Usage Accelerator – Sales Rep	Usage Accelerator - FINS Sales Rep
Usage Accelerator - Sales Manager	Usage Accelerator – FINS Sales Manager
Usage Accelerator - Sales Executive	Usage Accelerator – FINS Sales Executive

Note: You must not remove a Responsibility for a User and later add the same Responsibility for that user in Siebel CRM Administration View 'Responsibilities'. If you do so, the related ETL will fail. For more information, see the troubleshooting section in *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

The following table illustrates the dashboards and data-level security that are assigned to each Usage Accelerator responsibility. For more information about security in Oracle BI Applications, see *Oracle Business Intelligence Applications Security Guide*.

Table 8–2 Dashboards and Data-Level Security That Are Assigned to Each Usage Accelerator Responsibility

User Responsibility	Data Level Security	Dashboard Name (View)	Dashboard Page
Usage Accelerator - Sales Rep	Primary Position Data Level Security	Score Card	Individual Scorecard
Usage Accelerator - FINS Sales Rep	Primary Position Data Level Security	Action Plan	Account Coverage (Team)
Usage Accelerator - FINS Sales Rep	Primary Position Data Level Security	Action Plan	Contact Coverage (Team)
Usage Accelerator - FINS Sales Rep	Primary Position Data Level Security	Action Plan	Opportunity Coverage (Team)
Usage Accelerator - FINS Sales Rep	Primary Position Data Level Security	Action Plan	Financial Account Coverage (Team) - FINS Only
Usage Accelerator - FINS Sales Rep	Primary Position Data Level Security	Action Plan	Account Completeness (Team)
Usage Accelerator - FINS Sales Rep	Primary Position Data Level Security	Action Plan	Contact Completeness (Team)
Usage Accelerator - FINS Sales Rep	Primary Position Data Level Security	Action Plan	Opportunity Updates (Team)
Usage Accelerator - Sales Manager	No Position Based Security	Master Data Management - Customer Hub	Master Record Completeness
Usage Accelerator - Sales Manager	No Position Based Security	Master Data Management - Customer Hub	Master Record Completeness Detail
Usage Accelerator - Sales Manager	No Position Based Security	Master Data Management - Customer Hub	Accuracy
Usage Accelerator - Sales Manager	No Position Based Security	Score Card	Team Scorecard
Usage Accelerator - Sales Manager	No Position Based Security	Score Card	Individual Scorecard
Usage Accelerator – FINS Sales Manager	No Position Based Security	Action Plan	Account Coverage (Team)
Usage Accelerator – FINS Sales Manager	No Position Based Security	Action Plan	Contact Coverage (Team)
Usage Accelerator – FINS Sales Manager	No Position Based Security	Action Plan	Opportunity Coverage (Team)
Usage Accelerator – FINS Sales Manager	No Position Based Security	Action Plan	Financial Account Coverage (Team) – FINS Only
Usage Accelerator – FINS Sales Manager	No Position Based Security	Action Plan	Account Completeness (Team)
Usage Accelerator – FINS Sales Manager	No Position Based Security	Action Plan	Contact Completeness (Team)
Usage Accelerator – FINS Sales Manager	No Position Based Security	Action Plan	Opportunity Updates (Team)
Usage Accelerator – FINS Sales Manager	No Position Based Security	Coverage	Account Coverage
Usage Accelerator – FINS Sales Manager	No Position Based Security	Coverage	Contact Coverage

Table 8–2 (Cont.) Dashboards and Data-Level Security That Are Assigned to Each Usage Accelerator Responsibility

User Responsibility	Data Level Security	Dashboard Name (View)	Dashboard Page
Usage Accelerator – FINS Sales Manager	No Position Based Security	Coverage	Opportunity Coverage
Usage Accelerator – FINS Sales Manager	No Position Based Security	Coverage	Financial Account Coverage - FINS Only
Usage Accelerator – FINS Sales Manager	No Position Based Security	Completeness	Account Completeness
Usage Accelerator – FINS Sales Manager	No Position Based Security	Completeness	Contact Completeness
Usage Accelerator – FINS Sales Manager	No Position Based Security	Master Data Management - Customer Hub	Master Record Completeness
Usage Accelerator – FINS Sales Manager	No Position Based Security	Master Data Management - Customer Hub	Master Record Completeness Detail
Usage Accelerator – FINS Sales Manager	No Position Based Security	Master Data Management - Customer Hub	Accuracy
Usage Accelerator – FINS Sales Manager	No Position Based Security	Opportunity Updates	Opportunity Updates
Usage Accelerator – FINS Sales Manager	No Position Based Security	User Adoption	Active Users
Usage Accelerator – FINS Sales Manager	No Position Based Security	User Adoption	Application Usage - Excluded for FINS
Usage Accelerator – FINS Sales Manager	No Position Based Security	User Adoption	Application Usage - FINS Only
Usage Accelerator - Sales Executive	No Position Based Security	Scorecard	Team Scorecard
Usage Accelerator - Sales Executive	No Position Based Security	Scorecard	Individual Scorecard
Usage Accelerator - Sales Executive	No Position Based Security	Action Plan	Account Coverage (Team)
Usage Accelerator - Sales Executive	No Position Based Security	Action Plan	Contact Coverage (Team)
Usage Accelerator - Sales Executive	No Position Based Security	Action Plan	Opportunity Coverage (Team)
Usage Accelerator - Sales Executive	No Position Based Security	Action Plan	Financial Account Coverage (Team) – Fins Only
Usage Accelerator - Sales Executive	No Position Based Security	Action Plan	Account Completeness (Team)
Usage Accelerator - Sales Executive	No Position Based Security	Action Plan	Contact Completeness (Team)
Usage Accelerator - Sales Executive	No Position Based Security	Action Plan	Opportunity Updates (Team)
Usage Accelerator - Sales Executive	No Position Based Security	Coverage	Account Coverage
Usage Accelerator - Sales Executive	No Position Based Security	Coverage	Contact Coverage

Table 8–2 (Cont.) Dashboards and Data-Level Security That Are Assigned to Each Usage Accelerator Responsibility

User Responsibility	Data Level Security	Dashboard Name (View)	Dashboard Page
Usage Accelerator - Sales Executive	No Position Based Security	Coverage	Opportunity Coverage
Usage Accelerator - Sales Executive	No Position Based Security	Coverage	Financial Account Coverage – FINS Only
Usage Accelerator - Sales Executive	No Position Based Security	Completeness	Account Completeness
Usage Accelerator - Sales Executive	No Position Based Security	Completeness	Contact Completeness
Usage Accelerator - Sales Executive	No Position Based Security	Master Data Management - Customer Hub	Master Record Completeness
Usage Accelerator - Sales Executive	No Position Based Security	Master Data Management - Customer Hub	Master Record Completeness Detail
Usage Accelerator - Sales Executive	No Position Based Security	Master Data Management - Customer Hub	Accuracy
Usage Accelerator - Sales Executive	No Position Based Security	Opportunity Updates	Opportunity Updates
Usage Accelerator - Sales Executive	No Position Based Security	User Adoption	Active Users
Usage Accelerator - Sales Executive	No Position Based Security	User Adoption	Application Usage - Excluded for FINS
Usage Accelerator - Sales Executive	No Position Based Security	User Adoption	Application Usage - FINS Only

8.4.2.3 About Metric Definitions and Variables

This section explains Usage Accelerator metric definitions and variables, and contains the following topics:

- Section 8.4.2.3.1, "About the Deployment Date Variable"
- Section 8.4.2.3.2, "User Adoption Metrics"
- Section 8.4.2.3.3, "Opportunity Updates Metrics"
- Section 8.4.2.3.4, "Completeness Metrics"
- Section 8.4.2.3.5, "Coverage Metrics"

8.4.2.3.1 About the Deployment Date Variable Usage Accelerator module must be calibrated with a Deployment Date variable that determines the time that you want the Analysis to commence. This date will default to the date you create in the data warehouse, but you can change the date by changing the UA_DEPLOY_DT variable. Note that if you set the Deployment Date earlier than the current calendar date, metrics that rely on the Last Updated Date (such as # of Updated Records, # of Logins, # of Sync's, and so on) of a record may be inaccurate. For this reason we recommend setting the Deployment Date to the current Calendar Date.

8.4.2.3.2 User Adoption Metrics User Adoption Metrics provide visibility into the behavioral attributes and usage patterns of those using the system. These metrics enable you to quickly identify areas in need of improvement and individuals who require additional help meet the company's CRM standards.

Table 8–3 User Adoption Metrics

Metric Category	Description	Variables
Active User	Identifies users that have not logged in or synced in the last two rolling period.	Number of logins needed is currently set to 2. Variable Name is: ACTIVE_NUMLOGINS. Number of Syncs needed is currently set to 0. Variable Name is: ACTIVE_NUMSYNCS.
Login	Measures the number of days the user's login. Note – Only one login a day will be captured for each user.	Not Applicable
Sync	Measures the number of days the user's sync. Note – Only one sync a day will be captured for each user.	Not Applicable
Records Created	Tracks the number of Accounts, Contacts, Opportunities, Activities, Quotes created by each user and group. FINS version also tracks number of Financial Accounts and Policies Created.	Not Applicable
Records Updated	Tracks the number of Accounts, Contacts, Opportunities, Activities, Quotes updated by each user and group. FINS version also tracks number of Financial Accounts and Policies Created.	Not Applicable

8.4.2.3.3 Opportunity Updates Metrics Opportunity Updates Metrics deliver reliable information to organizations that use Opportunity Management for tracking their pipeline and sales forecasts. The metrics provide a focus on opportunities by ensuring that current and accurate information is in place.

Table 8–4 Opportunity Updates Metrics

Metric Category	Description	Variables
Opportunity Updates	Identifies Opportunities that have not been modified in the time period specified or have a close date that has passed.	Opportunity Updated Needed is currently set to 90 Days. Variable Name is: OPT_OUTDATED_NUMDAYS

8.4.2.3.4 Completeness Metrics Completeness Metrics assist in the completion of contact and account information by highlighting what is missing in an individual entry or in a set of entries.

Table 8–5 Information Completeness Metrics

Metric Category	Description	Variables
Account Completeness	Looks for the presence of information in each of the primary Account Address fields.	Address Line 1 City State Zipcode Phone

Table 8–5 (Cont.) Information Completeness Metrics

Metric Category	Description	Variables
Contact Completeness	Looks for the presence of information in each of the primary Contact Address fields.	Address Line 1 City State Zipcode Work Phone Email Address Title

8.4.2.3.5 Coverage Metrics Coverage Metrics monitor sales activity against Accounts, Contacts, Opportunities and Financial Accounts. It is not envisioned that you will need to use all of these metrics, just the ones that are in alignment with your sales process.

Table 8–6 Coverage Metrics

Metric Category	Description	Variables
Account Coverage	Determines whether an account has had a contact and activity associated to it in the time frame specified.	Activity Needed is currently set to 90 Days. Variable name is: ACC_NEED_ACT_NUMDAYS. Contact Needed is currently set to 180 Days. Variable name is: ACC_NEED_CON_NUMDAYS.
Contact Coverage	Determines whether a Contact has had an activity associated in the time frame specified.	Activity Needed is currently set to 90 Days. Variable Name is: CON_NEED_ACT_NUMDAYS.
Opportunity Coverage	Determines whether an Opportunity has had a contact and activity associated to it in the time frame specified.	Activity Needed is currently set to 90 Days. Variable name is: OPT_NEED_ACT_NUMDAYS. Contact Needed is currently set to 180 Days. Variable name is: OPT_NEED_CON_NUMDAYS.
Financial Account Coverage & Insurance Policy Coverage (FINS)	Determines whether a Financial Account or Insurance Policy has had an Activity Associated to it in the time frame specified	Activity Needed is currently set to 90 Days. Variable name is: AST_NEED_ACT_NUMDAYS.

8.4.2.4 Configuring Variables

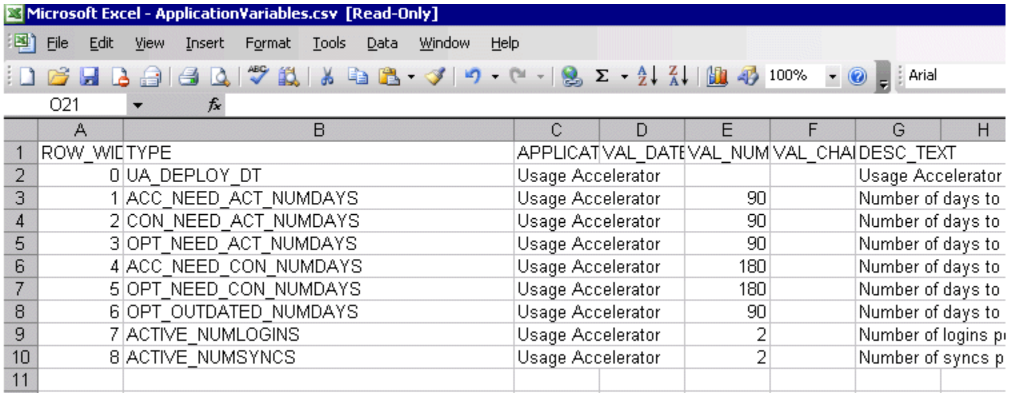
This section explains how to set values for the variables described in Section 8.4.2.3 for Siebel Business and Industry Applications.

1. Go to SiebelAnalytics\dwrep\Informatica\ParameterFiles.
2. Copy ApplicationVariables.csv and Day_FiscalCalendar.csv to the $PMServer\SrcFiles directory on the Informatica Server computer (for example, *INFA_HOME*\server\infa_shared\SrcFiles).

3. Edit the value of the parameters in the ApplicationVariables.csv file. Note that the variable names are listed in the TYPE column, and the variable values are listed in the VAL_NUM column.

   Figure 8–1 shows the ApplicationVariables.csv file.

Figure 8–1 ApplicationVariables.csv File


Microsoft Excel - ApplicationVariables.csv [Read-Only]
File Edit View Insert Format Tools Data Window Help
100%
Arial
O21
fx

	A	B	C	D	E	F	G	H
1	ROW_WID	TYPE	APPLICAT	VAL_DATE	VAL_NUM	VAL_CHAI	DESC_TEXT	
2	0	UA_DEPLOY_DT	Usage Accelerator				Usage Accelerator	
3	1	ACC_NEED_ACT_NUMDAYS	Usage Accelerator		90		Number of days to	
4	2	CON_NEED_ACT_NUMDAYS	Usage Accelerator		90		Number of days to	
5	3	OPT_NEED_ACT_NUMDAYS	Usage Accelerator		90		Number of days to	
6	4	ACC_NEED_CON_NUMDAYS	Usage Accelerator		180		Number of days to	
7	5	OPT_NEED_CON_NUMDAYS	Usage Accelerator		180		Number of days to	
8	6	OPT_OUTDATED_NUMDAYS	Usage Accelerator		90		Number of days to	
9	7	ACTIVE_NUMLOGINS	Usage Accelerator		2		Number of logins p	
10	8	ACTIVE_NUMSYNCS	Usage Accelerator		2		Number of syncs p	
11								

4. Save and close the file

8.4.3 About the Usage Accelerator Execution Plan

If you want to create a new Execution Plan or modify an existing Execution Plan to include a Usage Accelerator Subject area, then it should include a preceding task 'Create View For Usage Accelerator'. This is already done for the Subject Areas delivered out of the box. This step is also required if you are using any customized Subject Area containing Usage accelerator warehouse tables for your Execution Plans. These tables are prefixed with 'W_UA*'.

9 Configuring Oracle Contact Center Telephony Analytics

This section describes how to configure the Oracle Contact Center Telephony Analytics with the Universal adapter. It contains the following main topics:

- Section 9.1, "Overview of Oracle Contact Center Telephony Analytics"
- Section 9.2, "Configuration Required Before A Full Load for Contact Center Telephony Analytics"

> **Note:** For configuration steps that apply to all analytics modules, for example, Oracle Financial Analytics, Oracle HR Analytics, Oracle Sales Analytics, see Chapter 3, "Configuring Common Areas and Dimensions."

9.1 Overview of Oracle Contact Center Telephony Analytics

Oracle Contact Center Telephony Analytics tracks incoming and outgoing contacts and their resulting activities, and the agents performing contact center tasks. The Oracle Contact Center Telephony Analytics application is made up of these functional areas:

- Representative Activities
- Contact Center and Agent Performance
- Contact Center Benchmark and Targets
- Interactive Voice Response History

9.2 Configuration Required Before A Full Load for Contact Center Telephony Analytics

This chapter describes how to configure Contact Center Telephony Analytics. It contains the following topics:

- Section 9.2.1, "About the Dimension Key Resolution Process for Universal Source"
- Section 9.2.2, "How to Load the Dimension Table"
- Section 9.2.3, "How to Load the Fact Table"
- Section 9.2.4, "Setting Up The DATASOURCE_NUM_ID Parameter for the Universal Adapter"

- Section 9.2.5, "About Configuring the Event Type Columns"
- Section 9.2.6, "About Configuring The Contact Reason Column"
- Section 9.2.7, "About Configuring Contact Status Type Column"
- Section 9.2.8, "Setting Up the Representative Activities Table"
- Section 9.2.9, "Setting Up the Contact Center Performance Table"
- Section 9.2.10, "Setting Up the Benchmarks and Targets Table"
- Section 9.2.11, "Logical Delete of Records From Fact Tables"
- Section 9.2.12, "How to Configure Dates and Times"
- Section 9.2.13, "Configuring Flags for Oracle Contact Center Telephony Analytics"
- Section 9.2.14, "How to Exclude Representative Data for Post-Load Processing"

9.2.1 About the Dimension Key Resolution Process for Universal Source

Dimension keys uniquely identify each record in a dimension table. The purpose of dimension keys is to relate a record in the dimension table to a record in the fact table. Therefore, the dimension key must be stored in both the dimension table and the fact table and resolved using the dimension table's INTEGRATION_ID and DATASOURCE_NUM_ID column values.

For universal business adapters, users supply the dimension INTEGRATION_ID and DATASOURCE_NUM_ID column values through a flat file interface. The same values for INTEGRATION_ID and DATASOURCE_NUM_ID are expected in both the dimension and fact business adapters so that the correct dimension key is resolved and loaded into the fact table.

The dimension key resolution process entails two steps:

1. Run the dimension table workflows to extract and load dimension records.

   The dimension load mapping automatically creates a surrogate key for each record in the dimension table. This surrogate key value populates the dimension table's primary key column, which is referred to as the *dimension WID*. Similar to the INTEGRATION_ID column, which uniquely identifies the record within the source system, the dimension key uniquely identifies the record in the data warehouse dimension table.

2. Run the fact table workflows to extract and load fact records.

   Records must contain the dimension ID column values for each fact record; these values must be the same values as the INTEGRATION_ID in the corresponding dimension tables.

The following sections describe these two steps in more detail by taking the example of one fact table (W_REP_ACTIVITY_F) and one dimension table (W_EVENT_TYPE_D). However, this process applies to all fact and dimension tables joined by a dimension key.

9.2.2 How to Load the Dimension Table

This task is explained by taking the example of the W_EVENT_TYPE_D table.

Loading the W_EVENT_TYPE_D table requires the following ETL processes:

1. The SDE_Universal_EventTypeDimension mapping extracts the data from file_event_type.csv and populates the W_EVENT_TYPE_DS staging table.

2. The SIL_EventTypeDimension mapping sources data from the staging table and passes it over to the Source Independent Load (SIL) mapping. The SIL mapping generates the surrogate key for each record in the staging table, then inserts it into W_EVENT_TYPE_D target table.

9.2.3 How to Load the Fact Table

This task is explained by taking the example of the W_REP_ACTIVITY_F fact table.

Loading the W_REP_ACTIVITY_F fact table requires the following ETL processes:

1. The SDE_Universal_Rep_Activity mapping extracts the data from file_rep_activity.csv and populates the W_REP_ACTIVITY_FS staging table.
2. The SIL_Rep_ActivityFact mapping sources the data from the staging table, and the fact SIL mapplet resolves the dimension key by doing a lookup on W_EVENT_TYPE_D using the values supplied in the ACTIVITY_TYPE_ID column and the DATASOURCE_NUM_ID column. Then, the SIL populates the W_REP_ACTIVITY_F fact table.

Since the dimension *_ID values are supplied through the Universal Interface flat file, it is critical that you supply the same value for the INTEGRATION_ID in the dimension table and the corresponding *_ID field in the joined fact table. In addition, you must verify that the DATASOURCE_NUM_ID column values match (for Universal Sources, the value that is defined out-of-the-box for the DATASOURCE_NUM_ID column is 3). If you supply different values for the two tables, the fact table load mapping is not able to resolve the dimension key. As a result, you cannot perform queries on the fact table using that dimension.

9.2.4 Setting Up The DATASOURCE_NUM_ID Parameter for the Universal Adapter

All the Source Dependent Extract mappings (SDE) for the Universal Adapter are designed to accept the value for the DATASOURCE_NUM_ID column, either from the input data file or from a predefined parameter. If the value for a record is not supplied in a file, then the value assigned to the parameter is picked up.

The DATASOURCE_NUM_ID parameter is defined in the DAC Universal container and is set to the internal DAC system variable '@DAC_DATASOURCE_NUM_ID'. The value for this variable is dynamically set by DAC depending on the physical data source that is being used.

9.2.5 About Configuring the Event Type Columns

The W_EVENT_TYPE_D table is a dimension class table which tracks various types of events. This dimension table plays multiple roles in the Oracle Contact Center Telephony Analytics application. The various roles are described in the following section.

9.2.5.1 The Activity Type Column

Contact center representatives engage in various activities, such as logging into the Automated Call Distributor (ACD) system to handle customer calls, taking a scheduled break, taking an unscheduled break, and so on. All such activities that you want to load into the warehouse should be provided in the file_event_type.csv source file to be stored in the W_EVENT_TYPE_D table with the W_EVENT_CLASS column set to the REPACTIVITY domain value.

The ACTIVITY_TYPE_WID dimension key in the W_REP_ACTIVITY_F fact table identifies the nature of the activity. This key is resolved using the W_EVENT_TYPE_D table. To resolve the ACTIVITY_TYPE_WID dimension key in the W_REP_ACTIVITY_F table, the W_REP_ACTIVITY_F and W_EVENT_TYPE_D tables are joined through the ACTIVITY_TYPE_ID column and the DATASOURCE_NUM_ID column. For the ACTIVITY_TYPE_WID dimension key to resolve properly in the W_REP_ACTIVITY_F fact table, you must verify that the ACTIVITY_TYPE_ID column and the DATASOURCE_NUM_ID column values in file_rep_activity.csv file match with the INTEGRATION_ID column and the DATASOURCE_NUM_ID column values in the file_event_type.csv file. If the two columns do not match for a particular record, the fact load mapping cannot resolve the dimension key for that fact record.

9.2.5.2 Call Type (or Contact Type) Column

In a contact center there are various reasons for which customers contact an organization and organizations contact a customer. The contact can happen through various channels, such as phone, online chat, email, fax, and so on.

Some of the reasons for a customer to contact your organization include:

- Placing an order for a product or service.
- Lodging a complaint.
- Inquiring about a product offering.

Some of the reasons for your organization to contact a customer include:

- Performing a customer satisfaction survey.
- Following up on an inquiry.

The call types that you want to load into the Oracle Business Analytics Warehouse are provided in the file_event_type.csv source file to be stored in the W_EVENT_TYPE_D table with the W_EVENT_CLASS column set to the `CONTACT` domain value.

The CALL_TYPE_WID dimension key in W_ACD_EVENT_F fact table identifies the type of call. This key is resolved using the W_EVENT_TYPE_D table. To resolve the CALL_TYPE_WID dimension key in W_ACD_EVENT_F fact table, the W_ACD_EVENT_F and W_EVENT_TYPE_D tables are joined through the CALL_TYPE_WID column and the DATASOURCE_NUM_ID column. For the CALL_TYPE_WID dimension key to resolve properly in the W_ACD_EVENT_F fact table, you must verify that the CALL_TYPE_WID column and the DATASOURCE_NUM_ID column values in the file_acd_event.csv file match with the INTEGRATION_ID column and the DATASOURCE_NUM_ID column values in the file_event_type.csv file. If the two columns do not match for a particular record, the fact load mapping cannot resolve the dimension key for that fact record.

The CNTCT_TYPE_WID dimension key in the W_CNTCT_CNTR_PERF_F fact table identifies the same information and it is resolved in a similar process. It requires the CNTCT_TYPE_ID column and the DATASOURCE_NUM_ID column values in the file_cntct_cntr_perf.csv file to match with the INTEGRATION_ID column and the DATASOURCE_NUM_ID column values in the file_event_type.csv file.

9.2.5.3 Call Event Type Column

In a call center, the Computer Telephony Integration (CTI) system tracks each activity associated with a call, including:

- Call placed in queue to be answered
- Call answered by a contact representative
- Call placed on hold by a contact representative

- Call transferred from one contact representative to another
- Call hung up by the customer

The call events that you want to load into the Oracle Business Analytics Warehouse are provided in the file_event_type.csv source file and stored in the W_EVENT_TYPE_D table with the W_EVENT_CLASS column set to INTRA_CALL.

To resolve the CALL_EVENT_TYPE_WID dimension key in W_ACD_EVENT_F fact table, the W_ACD_EVENT_F and W_EVENT_TYPE_D tables are joined through the CALL_EVNT_TYPE_ID column and the DATASOURCE_NUM_ID column. For the CALL_EVENT_TYPE_WID dimension key to resolve properly in the W_ACD_EVENT_F fact table, you must verify that the CALL_EVNT_TYPE_ID column and the DATASOURCE_NUM_ID column values in file_acd_events.csv file match with the INTEGRATION_ID column and the DATASOURCE_NUM_ID column values in the file_event_type.csv file. If the two columns do not match for a particular record, the fact load mapping cannot resolve the dimension key for that fact record.

9.2.5.4 After-Call Work Activity (ACW) Type Column

On many occasions, contact representative are required to perform certain tasks after the initial contact. These tasks might include creating a follow-up action item list, dispatching the case from the contact to a particular group, and so on. These activities are known as after-call work (ACW) activities. The call events that you want to load into the Oracle Business Analytics Warehouse are provided in the file_event_type.csv source file and stored in the W_EVENT_TYPE_D table with the W_EVENT_CLASS column set to the ACWACTIVITY domain value.

To resolve the ACW_ACT_TYPE_WID dimension key in W_CNTCT_CNTR_PERF_F fact table, the W_CNTCT_CNTR_PERF_F table is joined with W_EVENT_TYPE_D table through the ACW_ACT_TYPE_ID column and the DATASOURCE_NUM_ID column. For the ACW_ACT_TYPE_WID dimension key to resolve properly in the W_CNTCT_CNTR_PERF_F fact table, you must verify that the ACW_ACT_TYPE_ID column and the DATASOURCE_NUM_ID column values in the file_cntct_cntr_perf.csv file match with the INTEGRATION_ID column and the DATASOURCE_NUM_ID column values in the file_event_type.csv file. If the two columns do not match for a particular record, the fact load mapping cannot resolve the dimension key for that fact record.

9.2.6 About Configuring The Contact Reason Column

The W_REASON_D table is a class dimension table that tracks various classes of reason such as those associated with the opportunity, opportunity status, quota, contact, contact transfer and so on. These classes are differentiated by the W_REASON_CLASS column.

For Oracle Contact Center Telephony Analytics, Contact Reason is an important reason class. All contacts made either by the customer to your organization, or by your organization to a customer, need to have a reason (for example, an inquiry or complaint).

The Contact Reasons that you want to load into the Oracle Business Analytics Warehouse are provided in the file_reason.csv source file to be stored in the W_REASON_D table with the W_REASON_CLASS column set to the domain value CONTACT. The Contact Transfer Reasons are provided with the W_REASON_CLASS set to CONTACT_TRANSFER.

To resolve the CNTCT_REASON_WID dimension key in W_CNTCT_CNTR_PERF_F fact table, the W_CNTCT_CNTR_PERF_F table is joined with W_REASON_D through the CNTCT_REASON_ID column and the DATASOURCE_NUM_ID column. For the CNTCT_REASON_WID dimension key to resolve properly in the W_CNTCT_CNTR_PERF_F fact table, you must verify that the CNTCT_REASON_ID column and the DATASOURCE_NUM_ID column values in file_cntct_cntr_perf.csv match with the INTEGRATION_ID column and the DATASOURCE_NUM_ID column values in the file_reason.csv file. If the two columns do not match for a particular record, the fact load mapping cannot resolve the dimension key for that fact record.

9.2.7 About Configuring Contact Status Type Column

The W_STATUS_D is a dimension class table that tracks various classes of status such as those associated with the purchase orders, requisitions, shipments, inventory, customer cases, contact statuses, and so on. These classes are differentiated by the W_STATUS_CLASS column.

For the Oracle Contact Center Telephony Analytics application, Contact Status is an important status class. All contacts made either by the customer to your organization, or by your organization to a customer, are assigned a status. Examples include:

- Customer abandoned call before contact
- Customer abandoned call during contact
- Contact completed

The contact statuses that you want to load into the Oracle Business Analytics Warehouse are provided in the file_status.csv source file to be stored in the W_STATUS_D table with the W_STATUS_CLASS column set to the `CONTACT` domain value.

The W_STATUS_CODE column in the W_STATUS_D table also contains domain values. The four domain values `ABANDONED`, `RELEASE`, `DISCONNECTED`, and `HANGUP` are used in the computation of Contact Center Performance metrics. Therefore, it is critical that while you load all your Contact Statuses through the source file, the records are mapped into the appropriate W_STATUS_CODE domain value. See Section 17.14, "Configuring the Domain Value Set with CSV Worksheet Files" for more information.

To resolve the CNTCT_STATUS_WID dimension key in W_CNTCT_CNTR_PERF_F fact table, the W_CNTCT_CNTR_PERF_F table is joined with W_STATUS_D through the CNTCT_STATUS_ID column and the DATASOURCE_NUM_ID column. For the CNTCT_STATUS_WID dimension key to resolve properly in the W_CNTCT_CNTR_PERF_F fact table, you must verify that the CNTCT_STATUS_ID column and the DATASOURCE_NUM_ID column values in file_cntct_cntr_perf.csv match with the INTEGRATION_ID column and the DATASOURCE_NUM_ID column values in the file_status.csv file. If the two columns do not match for a particular record, the fact load mapping cannot resolve the dimension key for that fact record.

9.2.8 Setting Up the Representative Activities Table

When setting up the Representative Activities table you must consider the following:

- All events in the Representative Activities table are time span events. The events are not point in time events.
- The calculation of the Actual, Scheduled, Login, and Break durations are based on the event durations in the source-system data. To avoid duplication in a representative's time, the representative activity records must not overlap in time. For example, if the Login and Break activities overlap in time in the source-system data, then the time durations are counted towards both categories.

- The hourly aggregate is the lowest level of aggregation provided. Representatives are counted as present for an hourly bucket if they are present for any part of that hour. For example, if a representative activity starts at 9.45 A.M. and ends at 10.45 A.M., the representative is counted as present for 9-10 A.M. and 10-11 A.M. time buckets. No weight factor is used to indicate the representative is available for part of the hour. However, the duration of activities are apportioned into the two hourly buckets. If the Local Time Zone is different from the warehouse time zone, the duration of activities will be apportioned in all the local and warehouse hours that had part of that activity.
- The number of breaks a representative takes is calculated by counting the number of break records. There is one break record for each representative for each break (scheduled or actual). If a break is split into multiple records in the source system, then it is counted as multiple breaks in the Oracle Business Analytics Warehouse.
- If a representative's activity spans across the date boundary, then you must provide two different records for that representative for the same activity, with different activity start dates and times. The same applies to the local date boundary as well. Hence, if an activity crosses the day boundary in warehouse time and local time at different points in time (due to time-zone difference between local and warehouse), there should be three records for that particular activity, each having an Activity start day that is the same as the Activity end day and the Activity start local day the same as the Activity end local day. For example, if a representative logs on to the Automatic Call Distributor (ACD) system at 23:30 on January 4, 2006 (23:00 on January 4, 2006 in local time) and logs off from the ACD system at 01:30 on January 5, 2006 (01:00 on January 5, 2006 in local time) then create three records in the file_rep_activity.csv flat file interface, as shown in the following table.

Table 9–1 Setting up activities and date boundaries

Rep	Activity Type	Activity Start Date	Activity Start Time	Activity End Date	Activity End Time	Activity Start Local Date	Activity Start Local Time	Activity End Local Date	Activity End Local Time
Rep1	LOGIN	01/04/2006	23:30:00	01/04/2006	23:59:59	01/04/2006	23:00:00	01/04/2006	23:29:59
Rep1	LOGIN	01/05/2006	00:00:00	01/05/2006	00:29:59	01/04/2006	23:29:59	01/04/2006	23:59:59
Rep1	LOGIN	01/05/2006	00:30:00	01/05/2006	01:30:00	01/05/2006	00:00:00	01/05/2006	01:00:00

To set up the Representative Activities table:

1. Using a text editor, open the file_rep_activity.csv file, located in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).
2. Type in your data to load the Representative Activities table.
3. Save and close the file.

9.2.9 Setting Up the Contact Center Performance Table

The Contact Center Performance table stores information at a contact and representative grain. For example, if a call is handled by two representatives, there are two records for that contact in the Contact Center Performance table.

When setting up the Contact Center Performance table you must consider the following:

- The Abandoned Contact Count, Answered Contact Count, Hangup Contact Count, and Released Contact Count metrics are counts of contacts based on the Contact Status. The Contact Center Performance table is preconfigured to expect the Contact Status in the file_cntct_cntr_perf.csv file is to be at a Contact level. If you configure the Contact Status at the contact and representative level, you need to ensure that these aggregate metrics are defined at the contact and representative level in the appropriate workflows. You need to make any changes in the `SELECT` clause of the Source Qualifier SQL statement in the PLP_ ContactCenterPerformanceRepHourAggregate_Load mapping. You also need to configure the metadata in the repository file. You need to change the definitions of these metrics in the Logical Table Source that maps to the W_CNTCT_CNTR_ PERF_F fact table.
- *Answered contacts* are defined as the contacts whose status is not marked as ABANDONED. The Answered Contact Count metric is calculated as follows:

  ```
  COUNT(DISTINCT (CASE WHEN W_STATUS_D.W_STATUS_CODE != 'ABANDONED' THEN
  W_CNTCT_CNTR_PERF_F.CNTCT_NUM END)).
  ```

  You can choose not to count calls which are completed in the Interactive Voice Response (IVR) as an answered call. You can exclude these contacts from the Answered Contact Count by assigning them a different or new Contact Status.
- As contacts are associated with a representative, a Contact Representative Identifier must be provided for each fact record. The IVR can be modeled as a virtual representative and assigned a Contact Representative Identifier. Calls waiting in a general queue, and not assigned to a representative, also need to be given their own Contact Representative Identifier.
- The majority of the data for the Contact Center Performance table is sourced from the data in the file_acd_event.csv file. You must ensure that the source data is consistent across the file_acd_event.csv and file_cntct_cntr_perf.csv files.

To set up the Contact Center Performance table:

1. Using a text editor, open the file_cntct_cntr_perf.csv file, located in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_ shared\SrcFiles).
2. Type in your data to load the Contact Center Performance table.
3. Save and close the file.

9.2.10 Setting Up the Benchmarks and Targets Table

The benchmarks and targets Subject Area is used to establish targets for various metrics for your Contact Center and also capture industry benchmarks for the same metrics. The values of these metrics are calculated using the Contact Center Performance Subject Area and are compared to the Benchmarks and Targets table to evaluate the performance of your Contact Center.

When setting up the Benchmarks and Targets table you must consider the following:

- The file_cntct_cntr_bnchmrk_tgt.csv file must supply the effective date range for each benchmark record. The date range is used to identify the appropriate benchmark to compare with the actuals and the determination of other metrics such as the Service Level. Actuals refers to the actual value of the metric (during the period) as opposed to the planned or targeted value of the metric.

- You need to supply an appropriate date range for the benchmark records. For example, if the benchmark records do not vary over time, a large date range can be used. An example is shown as follows:
  - PERIOD_START_DT 01/01/1899
  - PERIOD_END_DT 01/01/3714
- The Benchmarks and Targets table is preconfigured at the contact level. You can define other benchmarks and targets, for example, an Hourly-Total-Hold-Duration benchmark, and these can be added using the extension columns in the data warehouse. For more information on the methodology for storing additional data in the data warehouse, see Chapter 17, "Customizing the Oracle Business Analytics Warehouse."
- For each dimension in the Benchmark and Targets fact table, you can decide if a benchmark or target varies by that dimension or not. If you choose to keep a benchmark or target constant over a dimension, you need to supply a question mark (?) as the value for the dimension ID. In addition, the metric must be leveled in the repository (RPD) at the grand-total level of that dimension. This dimension ID also must be removed from the join in the SQL statement in the PLP_ContactCenterPerformanceServiceLevelAggregate mapping. If you choose to vary a benchmark or target by a dimension, you need to provide benchmark or target for each value of the dimension.
- The FORECAST_CNTCT_CNT table in the source file is preconfigured to forecast the number of calls for a day for a combination of dimensions.

The Benchmarks and Targets table is preconfigured with the smallest effective date range of a day. To changing the grain to be hourly, perform the following procedure.

To configure the grain of the Benchmarks and Targets table as an hourly level:

1. Add the Hour WID to the W_CNTCT_CNTR_BNCHMRK_TGT_F fact table.
2. Modify the SDE_Universal_ContactCenterBenchmarkTargetFact and SIL_ContactCenterBenchmarkTargetFact mappings to populate the Hour WID based on the Period Start Date and Period End Date.

   These dates need to fall on the hour boundaries and not in the middle of an hourly interval.
3. Modify the PLP_ContactCenterPerformanceRepHourAggregate_Load mapping SQL statement to now join also on Hour WID to the Benchmarks and Targets table.
4. Modify the metadata in the repository to include the new physical and logical joins to the W_HOUR_OF_DAY_D dimension.
5. Set the content pane settings on the fact table to the newly added Hour (Time) dimension in the RPD.

To set up the Benchmarks and Targets table:

1. Using a text editor, open the file_cntct_cntr_bnchmrk_tgt.csv file, located in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).
2. Type in your data to load the Benchmarks and Targets.
3. Save and close the file.

9.2.11 Logical Delete of Records From Fact Tables

If a fact record is deleted physically in the telephony data source, you can logically delete that record from the data warehouse. To do this, a target staging table called <FACT table>_PE must be created for each fact table, and a mapping must be created that can load this table.

For example, to delete records from W_ACD_EVENT_F, a new staging table must be created as W_ACD_EVENT_F_PE, where PE stands for primary extract. The structure of this table will be same for any fact table as it will have only two columns: INTEGRATION_ID and DATASOURCE_NUM_ID. A new mapping, SDE_Universal_<Fact>_Primary_Extract, must be created to load this the primary extract staging table with all the INTEGRATION_ID and DATASOURCE_NUM_ID records that have not been deleted and are currently present in OLTP system.

The following example illustrates this process. In this example, there are five records in the telephony data source and in the data warehouse table W_ACD_EVENT_F, as follows:

Table 9–2 Example records in the W_ACD_EVENT_F table

INTEGRATION_ID	DATASOURCE_NUM_ID	CALLID_NUM	DELETE_FLG
1	0	20060101_C1	N
2	0	20060101_C2	N
3	0	20060101_C3	N
4	0	20060101_C4	N
5	0	20060101_C5	N

For any reason, records with an Integration_Id = 3 and a Datasource_num_Id = 0 are deleted from the source system. Then Our primary extract mapping should load the remaining four records in the W_ACD_EVENT_PE table, as follows:

Table 9–3 Example records in the W_ACD_EVENT_PE table

INTEGRATION_ID	DATASOURCE_NUM_ID
1	0
2	0
4	0
5	0

This will be followed by the SIL_IdentifyDelete and the SIL_*_SoftDelete mappings, which are already available in the data warehouse repository, as installed out-of-the-box. The SIL_ACDEventFact_IdentifyDelete mapping will do a left outer join on W_ACD_EVENT_F and W_ACD_EVENT_F_PE to find out the missing records in W_ACD_EVENT_F_PE, and to check whether the missing record is not already deleted (DELETE_FLG='Y') in the W_ACD_EVENT_F. If not, it will load those records in the staging table W_ACD_EVENT_F_DEL.

Table 9–4 Example records in the W_ACD_EVENT_F_DEL table

INTEGRATION_ID	DATASOURCE_NUM_ID
3	0

The SIL_ACDEventFact_SoftDelete mapping will read records from the W_ACD_EVENT_F_DEL and update the DELETE_FLG of those records in W_ACD_EVENT_F to 'Y', as follows:

Table 9–5 Example records in the W_ACD_EVENT_F table

INTEGRATION_ID	DATASOURCE_NUM_ID	CALLID_NUM	DELETE_FLG
1	0	20060101_C1	N
2	0	20060101_C2	N
3	0	20060101_C3	Y
4	0	20060101_C4	N
5	0	20060101_C5	N

9.2.12 How to Configure Dates and Times

The Contact Center Telephony Analytics application supports analysis by both the local date/time and the Warehouse date/time. On the fact tables, the _LDT columns refer to the local date and time while the _DT fields refer to the warehouse date and time. If you want to capture the local date and time on a transaction, you will need to provide that data in the _LDT columns through the flat file interface. You will also have to apply the appropriate transformation to compute the warehouse date/time and supply this in the _DT columns through the flat file interface. Even if you choose not to support analysis based on local time, you will need to supply data for the _DT (warehouse date/time) columns. Table 9–6 provides a list of the applicable date and local date columns for each fact table.

Table 9–6 Date Columns and Flat Files

Flat file	Applicable Date Column	Applicable Local Date Column	Table Using the Local Date
file_rep_activity.csv	ACTIVITY_START_DT, ACTIVITY_END_DT	ACTIVITY_START_LDT, ACTIVITY_END_LDT	W_REP_ACTIVITY_F
file_acd_event.csv	EVENT_START_DT, EVENT_END_DT	EVENT_START_LDT, EVENT_END_LDT	W_ACD_EVENT_F
file_cntct_cntr_perf.csv	CNTCT_START_DT, CNTCT_END_DT	CNTCT_START_LDT, CNTCT_END_LDT	W_CNTCT_CNTR_PERF_F
file_rep_activity_hour.csv (Alternate Load)	ACTIVITY_DT	ACTIVITY_LDT	W_REP_ACTIVITY_HOUR_A
file_cntct_cntr_perf_rep_hour.csv (Alternate Load)	CNTCT_DT	CNTCT_LDT	W_CNTCT_CNTR_PERF_REP_HOUR_A

To provide local dates in the flat file interface:

1. Open the applicable flat file interface.
2. In the flat file interface, input the new dates in the *_DT fields.
3. Input the corresponding local dates in the *LDT fields.
4. Save your changes to the flat file interface.

5. Run a test load for 10 records to verify that your new dates are loaded into the applicable table.

9.2.13 Configuring Flags for Oracle Contact Center Telephony Analytics

Many of the fact and dimension tables within the Oracle Contact Center Telephony Analytics application use flag fields to provide value-added information pertaining to a contact or contact representative. These flag fields are configurable and include the following:

- CONSULT_ FLG
- CONFERENCE_FLG
- PERTINENT_ INFO_FLG
- CNTCT_MTCH_FLG
- IVR_FLG

The possible values for these flag fields in the data warehouse tables are Y or N. Table 9–7 shows Configurable Flag Values and Descriptions.

Table 9–7 Configurable Flag Values and Descriptions

Flag	Flag Value	Description
CONSULT_FLG	Y	Indicates that the contact representative consulted with other contact representative during the course of the call or contact.
-	N	Indicates that the contact representative did not consult with other contact representative during the course of the call or contact.
CONFERENCE_FLG	Y	Indicates that the contact representative conferenced with other contact representatives during the course of the call or contact.
-	N	Indicates that the contact representative did not conference with other contact representative during the course of the call or contact.
PERTINENT_INFO_FLG	Y	Indicates that the pertinent information was available for the contact.
-	N	Indicates that the pertinent information was not available for the contact.
CNTCT_MTCH_FLG	Y	Indicates that the contact was matched with the existing customer data using Customer Entered Digits (CED) such as PIN Numbers, Account Numbers, or Social Security Number.
-	N	Indicates that the contact could not be matched with the existing customer data using Customer Entered Digits (CED) such as PIN Numbers, Account Numbers, or Social Security Number.
IVR_FLG	Y	Indicates that the call associated with the call was recorded in the Interactive Voice Response (IVR) system
-	N	Indicates that the call associated with the call was not recorded in the Interactive Voice Response (IVR) system

9.2.14 How to Exclude Representative Data for Post-Load Processing

This section contains information about excluding representative data from contact representative and organization performance aggregate tables for post-load processing.

You may want to exclude certain contact representatives from the aggregation process for various reasons. For example, you may want to exclude representatives who are on vacation, or who are working on special projects.

The default configuration calculates contact-related information for all contact representatives in the enterprise. There are five aggregate tables supplied with the Oracle Contact Center Telephony Analytics application for improving the performance of the dashboards and reports:

- W_CNTCT_CNTRC_PERF_REP_HOUR_A
- W_CNTCT_CNTR_PERF_REP_DAY_A
- W_CNTCT_CNTR_PERF_REP_MONTH_A
- W_CNTCT_CNTR_PERF_ORG_HOUR_A
- W_CNTCT_CNTR_PERF_ORG_DAY_A

To exclude data about specific representatives from the aggregation calculation:

1. In Informatica PowerCenter Designer, open the PLP folder.
2. Open the PLP_ContactCenterPerformanceRepAggregate_Extract mapping.
3. Double-click the EXP_CNTCT_CNTR_PERF transformation.
4. Locate the EXCLUSION_IND_FLG port.
5. This port is preconfigured with a value of N indicating that all rows are included in the aggregates. Change this logic to include your logic to determine which groups of records you want to exclude.
6. Validate and save the mapping to the repository.

> **Note:** If you exclude data from an aggregate table, you also need to apply the same filter to the Logical Table Source corresponding to the W_CNTCT_CNTR_PERF_F base fact table in the repository metadata (Fact Service Contact Center Performance logical table). The metrics computed from the base fact tables are now consistent with those computed from the aggregate tables.

9.2.15 How to Configure Alternate Load Plan to Enable Direct Load to the Entry-Level Rep Activity Aggregates

The Representative Activity fact table stores information at an individual representative activity grain. In case the data is not available at that grain, you can alternately choose to provide summarized data at the hourly level and load the Rep Activity hour level aggregate table W_REP_ACTIVITY_HOUR_A directly.

This data will be used by the post-load process to load the following higher level aggregates:

- W_REP_ACTIVITY_DAY_A
- W_REP_ACTIVITY_MONTH_A

To configure the alternate load plan to directly load aggregate table:

1. In DAC, display the Subject Areas tab and query for the 'Service – Rep Activity' Subject Area.
2. With this Subject Area selected, display the Configuration Tags subtab.
3. Deselect the configuration tag 'ECC- Load into base fact and then aggregates'.
4. Check the configuration tag 'ECC – Load directly into aggregate facts'.
5. Reassemble the Subject Area 'Service – Rep Activity' and save.
6. Rebuild the Execution Plan 'Universal Contact Center Telephony Analytics'.

To provide hour level data:

1. Using a text editor, open the file rep_activity_hour.csv, located in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).
2. Enter the data for Representative activities at the hourly level.
3. Save and close the file.

> **Note:** If you directly load the entry-level aggregate table, the metrics in the Fact - Service - Rep Activities logical fact table that are mapped only to the base W_REP_ACTIVITY_F LTS will not be populated.

9.2.16 How to Configure Alternate Load Plan to Enable Direct Load to the Entry-Level Contact Performance Aggregates

The Contact Center Performance table stores information at a contact and representative grain. In case the data is not available at a contact grain, you can alternately choose to provide summarized data at the hourly level and load the Contact Center Performance hourly aggregate table, W_CNTCT_CNTR_PERF_REP_HOUR_A, directly.

This data will be used by the post-load process to load the following higher level aggregates.

W_CNTCT_CNTR_PERF_REP_DAY_A

W_CNTCT_CNTR_PERF_REP_MONTH_A

W_CNTCT_CNTR_PERF_ORG_HOUR_A

W_CNTCT_CNTR_PERF_ORG_DAY_A

To configure the alternate load plan to directly load aggregate:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Subject Areas tab, and query for the 'Service - Contact Center Performance' Subject Area.
3. With this Subject Area selected, display the Configuration Tags tab.
4. Deselect the Inactive check box for the configuration tag 'ECC- Load into base fact and then aggregates'.
5. Select the Inactive check box for the configuration tag 'ECC – Load directly into aggregate facts'.
6. Reassemble the Subject Area 'Service - Contact Center Performance and save.

7. Rebuild the Execution Plan 'Universal Contact Center Telephony Analytics'.

To provide hour level data:

1. Using a text editor, open the file_rep_activity_hour.csv, located in the folder $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).
2. Enter the data for Contact Center Performance at the hourly level.
3. Save and close the file.

> **Note:** The Configuration Tags for the Direct Load to both Rep Activity and Contact Center Performance Hourly Aggregates is the same and hence either both can be loaded at hourly level or both can be loaded at point-in-time grain. If you directly load the entry-level aggregate table, the metrics in the Fact - Service - Contact Center Performance logical fact table that are mapped only to the base W_CNTCT_CNTR_PERF_F LTS will not be populated.

9.2.17 How to Set Up the Integration_Id of the Contact Center Telephony Facts in the Warehouse

The INTEGRATION_ID by default is provided in the SrcFiles and they are unique for a DATASOURCE_NUM_ID in a fact. Alternately, they can also be formed in the warehouse in the SDE adapter mappings using a set of Dimension_Ids and other columns of each record. The combination of Dimension Ids and other columns used for forming Integration_id is unique within a DATASOURCE_NUM_ID and identifies the grain of the fact. For example, in the W_ACD_EVENT_F fact table, INTEGRATION_ID is formed as CALLID_NUM||'~'||EVENT_SEQUENCE_NUM.

To set up formation of Integration_Id in SDE adapter:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab, and query for the SDE_Universal task.
3. In the lower pane, display the Parameters subtab.
4. Change the value of the parameter $$READ_INTEGRATION_ID_FROM_FILE to FALSE.

Table 9–8 lists the expression of INTEGRATION_ID for each fact and the SDE task that creates it.

Table 9–8 Expressions and Task Names

Table Name	INTEGRATION_ID Expression	Task Name
W_ACD_EVENT_F	CALLID_NUM||'~'||EVENT_SEQUENCE_NUM	SDE_Universal_ACDEventFact
W_CNTCT_CNTR_PERF_F	CNTCT_NUM||'~'||CNTCT_REP_ID||'~'||CNTCT_START_DT	SDE_Universal_ContactCenterPerformanceFact

Table 9–8 (Cont.) Expressions and Task Names

<table>
<tr><th>Table Name</th><th>INTEGRATION_ID Expression</th><th>Task Name</th></tr>
<tr><td>W_CNTCT_CNTR_BNCHMRK_TGT_F</td><td>PERIOD_START_DT||'~'||COMPANY_ORG_ID||'~'||CNTCT_REP_ID||'~'||REP_HR_ORG_ID||'~'||CNTCT_REP_ORG_ID||'~'||CNTCT_CNTR_LOC_ID||'~'||CHNL_TYPE_ID||'~'||CUSTOMER_ID||'~'||PRODUCT_ID</td><td>SDE_Universal_ContactCenterBenchmarkTargetFact</td></tr>
<tr><td>W_CNTCT_CNTR_PERF_REP_HOUR_A</td><td>CNTCT_CNTR_LOC_ID||'~'||CNTCT_REP_ID||'~'||CNTCT_DT</td><td>SDE_Universal_ContactCenterPerformanceRepHourAggregate</td></tr>
<tr><td>W_REP_ACTIVITY_F</td><td>CNTCT_REP_ID||'~'||ACTIVITY_TYPE_ID||'~'||ACTIVITY_START_DT||'~'||ACTIVITY_START_LDT</td><td>SDE_Universal_RepActivityFact</td></tr>
<tr><td>W_REP_ACTIVITY_HOUR_A</td><td>CNTCT_REP_ID||'~'||ACTIVITY_TYPE_ID||'~'||ACTIVITY_DT</td><td>SDE_Universal_RepActivityHourAggregate</td></tr>
</table>

9.2.18 Setting Up the IVR Menu and IVR Navigation Profile Tables

The IVR Menu Dimension table W_IVR_MENU_D stores information about each menu point in the Interactive Voice Response System. The Integration_id in this table uniquely identifies a IVR MENU and is used as the dimension identifier IVR_MENU_ID in the ACD Event Fact fact table, W_ACD_EVENT_F.

If a call has n number of consecutive IVR events without any transfer to a contact representative, the NAV_PATH_ID for that call is identified as MENU1-MENU2-MENU3-……..-MENUn where MENUm is the IVR_MENU_ID of the mth event in that call. The NAV_PATH_ID is used to determine the navigation profile dimension key for a call.

It is recommended that the Integration_Id of the IVR MENU dimension W_IVR_MENU_D does not exceed 10 characters in order to keep the NAV_PATH_ID within the defined size limit of 255 characters.

9.2.18.1 Determining Navigation Profile and Transfer Flag for a Call

If a call has multiple transfer from IVR to Rep or REP to IVR, a Navigation Profile is determined for each segment of the call that was completely within the IVR system. For example, suppose a call had eight events as shown in Table 9–9.

Table 9–9 Example Call Events in a Navigation Profile

CallId	Event Type	Event Sequence Number	Menu Id
20060104-C1	IVR	1	MENU1
20060104-C1	IVR	2	MENU2
20060104-C1	IVR	3	MENU3
20060104-C1	REP	4	-
20060104-C1	IVR	5	MENU4

Table 9–9 (Cont.) Example Call Events in a Navigation Profile

CallId	Event Type	Event Sequence Number	Menu Id
20060104-C1	IVR	6	MENU5
20060104-C1	IVR	7	MENU6
20060104-C1	IVR	8	MENU7

There will be two NAV_PATH_ID for both IVR segment of the call. Transfer Flag will be 'Y' for the first segment because the call was transferred to REP after this segment and it will be 'N' for the next segment where it completed in IVR itself.

Table 9–10 Example Navigation Path IDs

CallId	NAV PATH ID	Transfer Flag
20060104-C1	MENU1-MENU2-MENU3	Y
20060104-C1	MENU4-MENU5-MENU6	N

The mapping PLP_IVRNavigation Profile Dimension will load two records in the IVR Navigation Profile Dimension table W_IVR_NAV_PROFL_D, one for NAV_PATH_ID 'MENU1-MENU2-MENU3' and 'MENU4-MENU5-MENU6' and create unique surrogate key for each of them. If any NAV_PATH_ID already exists in the table W_IVR_NAV_PROFL_D and is Effective on the day when the call was made, that record will not be inserted.

9.2.18.2 Slowly Changing Dimension Behavior of IVR Menu and IVR Navigation Profile Dimensions

IVR Menu dimension is configured as a slowly changing dimension. The history of any change in the following attributes will be maintained in the IVR Menu Dimension.

- PROMPT_MSG
- PROMPT_MESG_DURN
- RSPNS_ERROR_MSG
- MENU_BROADCAST_MSG
- TIMEOUT_DURN
- TIMEOUT_MSG
- MENU_ACTIVE_FLG

When one or more of the above listed attributes change, the new record is inserted in W_IVR_MENU_D and a new dimension key (ROW_WID) is assigned to it. This change, in turn, result into a change in all the Navigation Profiles which has this MENU_WID at any position from 1 to 15th. The history of this change is maintained in the IVR Navigation profile dimension table, W_IVR_NAV_PROFL_D.

9.2.19 How to View Duration Metrics in hh:mm:ss Format in Oracle BI Answers

The Oracle BI Applications metadata repository (.rpd) has several duration metric in the 'Fact – Service - Contact Center Performance' and 'Fact – Service – Rep Activities' subject areas. Most of these duration metrics also have a version that is a pre-formatted number to be displayed in the hh:mm:ss format. In order to view these duration metrics in hh:mi:ss format in an Oracle BI Answers report, do the following:

1. Select the duration in hh:mi:ss metric using the column selectors in the 'Criteria' tab.
2. Click Format Column to open the Column Properties dialog.
3. Go to Data format.
4. Select 'Override Default data format'.
5. Select Treat Number as 'Custom' and give Custom Numeric Format as "0#:##:##".

10

Configuring Oracle Price Analytics

This section describes how to configure Oracle Price Analytics. It contains the following topics:

- Section 10.1, "Overview of Oracle Price Analytics"
- Section 10.2, "Configuration Required Before a Full Load for Oracle Price Analytics"
- Section 10.3, "Price Waterfall Element Sample Data"

10.1 Overview of Oracle Price Analytics

Oracle Price Analytics is aimed at pricing analysis, sales operations, product marketing and management, and finance. It provides pricing analytics across the full price waterfall of contracts, quotes, orders and competitor pricing, allowing business users to do the following:

- Identify the true realized price, as well as understand how price is diluted at appropriate summary and transactional levels.
- Monitor price and discounting trends across segment, strategy, channel, geography and time.
- Understand price and margin variation within the same products or customers through price band analysis.
- Look for "outliers" in customer and product profitability to determine the root cause as to why some groups perform better than others and to identify surgical pricing policy and deal opportunities.
- Combine insights from historical data with Oracle data mining and predictive technologies to improve forward-looking decisions
- Break down price elements at the transaction level and identify the true realized price and margin through detailed waterfall analysis
- Highlight discounts that are disproportionate to volume and determine which regions, products or customers are responsible

The following sources can populate pricing data:

- Siebel CRM 8.1 and 8.1.1
- Universal source

10.2 Configuration Required Before a Full Load for Oracle Price Analytics

This section contains configuration steps that you need to perform on Oracle Price Analytics before you do a full data load. It contains the following topics:

- Section 10.2.1, "Configuration Steps for Oracle Price Analytics for All Source Systems"
- Section 10.2.2, "About Configuring Domain Value and CSV Worksheet Files for Oracle Price Analytics"
- Section 10.2.3, "How to Configure the domainValues_PriceWaterfallElement_Name.csv"
- Section 10.2.4, "Configuration Steps for Universal Sources"

10.2.1 Configuration Steps for Oracle Price Analytics for All Source Systems

For configuration steps that apply to all Oracle BI Applications modules, see Chapter 3, "Configuring Common Areas and Dimensions."

10.2.2 About Configuring Domain Value and CSV Worksheet Files for Oracle Price Analytics

Table 10–1 lists the CSV worksheet files and the domain values for Oracle Price Analytics that are located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

Table 10–1 Domain Values and CSV Worksheet Files for Oracle Price Analytics

Worksheet File Name	Description	Session
domainValues_PriceWaterfallElement_Name_SBL.csv	Lists the Price Waterfall Element Name column and the corresponding domain values for the Siebel 8.1.1 application. For information about how to edit this file, see Section 10.2.3, "How to Configure the domainValues_PriceWaterfallElement_Name.csv."	SDE_PriceWaterfallElementDimension

10.2.3 How to Configure the domainValues_PriceWaterfallElement_Name.csv

This section provides instructions for how to configure the domainValues_PriceWaterfallElement_Name.csv file.

The waterfall element names you will use are taken from the Siebel CRM flat file FILE_PWF_ELEMENT.csv. The different element types used in the sample data in this file are the following:

- **Segment**

  The revenues that are part of a waterfall, such as ceiling revenue, list revenue, and so on.

- **Revenue Adjustment**

  The adjustments made to the segment elements, for example, ceiling adjustment, invoice adjustment, and so on.

- **Cost Adjustment**

  All other adjustments that are not part of any segment.

To configure the domainValues_PriceWaterfallElement_Name.csv file

1. Identify the different waterfall elements and the names that are currently used by looking at the file FILE_PWF_ELEMENT.csv located in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).
2. Copy the ELEMENT_NAME and ELEMENT_TYPE columns.

> **Note:** Copy the data starting after the fifth line. The first five lines contain descriptions and header information.

3. Open the domainValues_PriceWaterfallElement_Name_SBL.csv file using a text editor. This file is located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
4. Paste the ELEMENT_NAME and ELEMENT_TYPE columns you copied from the FILE_PWF_ELEMENT.csv file into the domainValues_PriceWaterfallElement_Name_SBL.csv file.
5. Map each waterfall element name and type to one domain value.
6. Save and close the file.

10.2.4 Configuration Steps for Universal Sources

Oracle Price Analytics relies on data from universal sources, such as flat files, for waterfall related data.

Table 10–2 lists the flat file source tables and the corresponding data warehouse tables for waterfall related data.

Table 10–2 Flat File Source Tables and Corresponding Warehouse Tables

Flat File	Description	Loads Target
FILE_PRI_STRATEGY_DS	This file holds information about the different pricing strategies being used.	W_PRI_STRATEGY_D
FILE_PRI_SEGMENT_DS	This file holds the different pricing segment details.	W_PRI_SEGMENT_D
FILE_PWF_ELEMENT	This file contains information about the different waterfall elements.	W_PWF_ELEMENT_D
FILE_ORDIT_WTR_LOG_FS	This file holds the waterfall information for all the transaction data for Order Item.	W_ORDIT_WTR_LOG_F
FILE_QTEIT_WTR_LOG_FS	This file holds the waterfall information for all the transaction data for Quote Item.	W_QTEIT_WTR_LOG_F

10.2.4.1 Populating Flat File Data For Siebel Sources

This section provides guidelines for populating pricing data into flat files when the source is Siebel.

Oracle Price Analytics does not provide a way to load pricing strategy, pricing segment or price waterfall element information from a Siebel source. All such dimensions must be loaded with a universal source, such as flat files.

The source files for the pricing-related dimensions must conform to the following rules:

- The Pricing Segment and Pricing Strategy IDs provided in the flat file must be the same for all the order lines in any given order.
- The ROW_ID must be unique in all the flat files because they are used to form the Integration IDs.
- The information added must be consistent with the existing data in the Siebel system. For instance, the Competitor Name added in the file must exist in the source system for proper resolution.
- The Order Line IDs in the Order Item Waterfall fact source must exist in the source table S_ORDER_ITEM.
- The Quote Line IDs in Quote Item Waterfall fact source must be a part of source table S_QUOTE_ITEM.

The Oracle Price Analytics facts W_ORDIT_WTR_LOG_F and W_QTEIT_WTR_LOG_F are loaded using the Order Item and Quote Item facts as well as flat files.

The pricing columns in the Order Item and Quote Item facts are loaded as shown in Table 10–3.

Table 10–3 Pricing Column in Order Item and Quote Item Facts

Column Name	Expression
CEIL_PRI	IIF(ISNULL(FLAT_FILE_DATA),START_PRI,FLAT_FILE_DATA)
SEG_PRI	IIF(ISNULL(FLAT_FILE_DATA),START_PRI,FLAT_FILE_DATA)
INV_PRI	IIF(ISNULL(FLAT_FILE_DATA),NET_PRI,FLAT_FILE_DATA)
PKT_PRI	IIF(ISNULL(FLAT_FILE_DATA),NET_PRI,FLAT_FILE_DATA)
PKT_MARGIN	IIF(ISNULL(FLAT_FILE_DATA),START_PRI-NET_PRICE,FLAT_FILE_DATA)

If you need to load different values for the pricing columns other than the existing prices, you can use the flat files FILE_ORDERITEM_FS.csv and FILE_QUOTEITEM_FS.csv. Based on the Integration IDs, the pricing data is looked up from these flat files and loaded into the fact tables.

10.2.4.2 Populating Flat File Data for Non-Siebel Sources

This section provides guidelines for populating pricing data into flat files for non-Siebel sources.

For non-Siebel sources, the source files for the pricing-related dimensions must conform to the following rules:

- The Order Line IDs in the Order Item Waterfall fact source must exist in fact file source FILE_ORDERITEM_DS.
- The Quote Line IDs in Quote Item Waterfall fact source must be a part of the fact file source FILE_QUOTEITEM_DS.

- Ensure all the ROW_IDs are unique so as to avoid any duplication or index issues.
- All the fact IDs added must be consistent with the ROW_ID of dimension file sources for proper resolution.

10.2.4.3 Data Standards for Flat Files

The flat files being used for Oracle Price Analytics facts, such as FILE_ORDIT_WTR_LOG_FS and FILE_QTEIT_WTR_LOG_FS, must be consistent with the line item tables. The prices in the waterfall log table must be the aggregated price in the line item tables. And, in the case of assembled or packaged products, the item tables store the package or assembly and the individual items that make up the package or assembly as separate line items. The line items in the flat file must store the individual prices and not rolled up prices; that is, if a package does not have a price and only the items inside it have prices, either the price of the package should be 0 and the items should have the prices or the package should have the rolled up prices and the item prices should be 0 to prevent double counting. Also, the Waterfall log table should store only the package or assembly and not the items that comprise it, and the price should be the rolled up price for a unit package or assembly.

10.3 Price Waterfall Element Sample Data

This section provides price waterfall element sample data.

Table 10–4 Price Waterfall Element Sample Data

ELEMENT_NAME	ELEMENT_TYPE	GROUP_NAME	BASIS_SEGMENT	TOKEN	ORDER_INDEX	REVN_COST_IND	DISP_ON_ZERO
Ceiling Revenue	Segment	Revenue	Ceiling Revenue	CEILING	1	0	Y
Segment Revenue	Segment	Revenue	Segment Revenue	SEGMENT	4	0	Y
Invoice Revenue	Segment	Revenue	Invoice Revenue	INVOICE	7	0	Y
Pocket Revenue	Segment	Revenue	Pocket Revenue	POCKET	10	0	Y
Pocket Margin	Segment	Revenue	Pocket Margin	POCKET MARGIN	15	0	Y
Cost	Adjustment	Cost	Pocket Revenue	COST	13	1	Y
Cost Adjustment	Cost Adjustment	Cost Adjustment	Pocket Revenue	OFF_COST	14	1	Y
Ceiling Adjustment	Adjustment Revenue	Customer Adjustment	Ceiling Revenue	OFF_CEILING	2	0	Y
Volume Adjustment	Revenue Adjustment	Rebates	Segment Revenue	OFF_SEGMENT	5	0	Y
Invoice Adjustment	Revenue Adjustment	Memo	Invoice Revenue	OFF_INVOICE	8	0	Y
Pocket Adjustment	Revenue Adjustment	Services	Pocket Revenue	OFF_POCKET	11	0	Y
Product Adjustment	Revenue Adjustment	Rebates	Segment Revenue	OFF_SEGMENT	6	0	Y

10.3.1 Example of an Order for a Simple Product

In this scenario, a simple order is created for a company that manufactures and sells laptops. Table 10–5 provides an example of the order information in the Order Item fact table.

Table 10–5 Sample Data for a Simple Product

Sales Order Number	LINE ID	PRODUCT	QUANTITY	UNIT PRICE	TOP_LVL_LN_ FLG	INC_CALC_IND
100	1001	Laptop	10	$1,248	Y	1

Table 10–6 provides an example of the Order Item waterfall log fact data for the transaction represented in Table 10–5.

Table 10–6 Order Item Waterfall Log Fact Data for a Simple Product

Order Identifier	Line ID	Row Identifier	PWF Element Identifier	Extended Qty	Unit Price	Element Amount
100	1001	2001	Ceiling Revenue	10	$1,248	$12,480
100	1001	2002	Ceiling Revenue	10	$(200)	$(2000)
100	1001	2003	Segment Revenue	10	$1,048	$10,480
100	1001	2004	Volume Adjustment	10	$(100)	$(1000)
100	1001	2005	Product Adjustment	10	$(50)	$(500)
100	1001	2006	Invoice Revenue	10	$898	$8980
100	1001	2007	Invoice Adjustment	10	$(120)	$(1200)
100	1001	2008	Pocket Revenue	10	$778	$7780
100	1001	2009	Pocket Adjustment	10	$(88)	$(880)
100	1001	2010	Cost	10	$(400)	$(4000)
100	1001	2011	Pocket Margin	10	$290	$2900

As this example shows, each waterfall element is stored as an individual record and the Waterfall Element dimension identifies whether the element is a revenue or an adjustment.

10.3.2 Example of an Order for a Configured Product

Table 10–7 provides an example of an order for an assembled product that has multiple child products.

Table 10–7 Sample Data for an Assembled Product

Sales Order Number	LINE ID	PRODUCT	QUANTITY	UNIT PRICE	TOP_LVL_ LN_FLG	INCL_CALC_ IND
101	1002	Desktop	1	$1,200	Y	1
101	1003	Monitor	1	$800	Y	0
101	1004	Keyboard	1	$250	N	0
101	1005	Mouse	1	$150	N	0

The Price Waterfall is stored for the packaged product and not the individual child items. Table 10–8 provides an example of the Order Item waterfall log fact data for the transaction represented in Table 10–7.

Table 10–8 Order Item Waterfall Log Fact Data for an Assembled Product

Order Identifier	Line ID	Row Identifier	PWF Element Identifier	Extended Qty	Unit Price	Element Amount
101	1002	2012	Ceiling Revenue	1	$1,200	$1,200
101	1002	2013	Ceiling Adjustment	1	$(200)	$(200)
101	1002	2014	Segment Revenue	1	$1,000	$1,000
101	1002	2015	Volume Adjustment	1	$(100)	$(100)
101	1002	2016	Product Adjustment	1	$(50)	$(50)
101	1002	2017	Invoice Revenue	1	$850	$850
101	1002	2018	Invoice Adjustment	1	$(120)	$(120)
101	1002	2019	Pocket Revenue	1	$730	$730
101	1002	2020	Pocket Adjustment	1	$(80)	$(80)
101	1002	2021	Cost	1	$(400)	$(400)
101	1002	2022	Pocket Margin	1	$250	$250

11

Configuring Oracle Service Analytics

This section describes how to configure Oracle Service Analytics, and contains the following topics:

- Section 11.1, "Overview of Oracle Service Analytics"
- Section 11.2, "Configuration Required Before A Full Load With An Oracle EBS Source System"
- Section 11.3, "Configuration Required Before A Full Load With Siebel CRM Source Systems"

11.1 Overview of Oracle Service Analytics

Oracle Service Analytics provides organizations with a comprehensive, up-to-date overview of customer service effectiveness. It enables companies to take targeted actions to improve productivity of Service centers, reduce costs, and increase customer satisfaction.

Oracle Service Analytics provides analysis for the following objects:

- Service Request, which helps in tracking and analyzing important metrics related to Service Requests such as SR Resolution, SR Aging, Average SRs per Employees, and so on.
- Activities, which provides insight into the performance of field service representatives involved in schedules maintenance activities and other scheduled/non-scheduled field activities. It also allows the field service managers to analyze the cost and expenses incurred for any field service activities.
- Asset, which helps in tracking assets installed at customer base, analyzing service activities within install base and warranty statuses.
- Agreement/Contract, which provides the Service Executive and Service Manager with critical insight into service contract profitability and helps them determine the revenue impact of service program and their acceptance with customers.
- Survey, which consolidates the customer surveys to gather statistics on customer satisfaction that helps the organization to provide better services to the customer.

11.2 Configuration Required Before A Full Load With An Oracle EBS Source System

This section contains configuration steps that you need to perform on Oracle Service Analytics before you do a full data load using an Oracle EBS Source System. It contains the following topics:

- Section 11.2.1, "Configuring DAC Parameters for Oracle Service Analytics With An Oracle EBS Source System"
- Section 11.2.2, "Configuring Domain Values for Oracle Service Analytics With An Oracle EBS Source System"
- Section 11.2.3, "Configuring Your RPD Repository for Oracle Service Analytics With An Oracle EBS Source System"

11.2.1 Configuring DAC Parameters for Oracle Service Analytics With An Oracle EBS Source System

This section summarizes the DAC parameters that you can set for Oracle Service Analytics. To set these parameters in DAC, go to the Design view, and select the appropriate custom container from the drop-down list. Then, display the Source System Parameters tab.

Table 11–1 lists the Oracle Service Analytics DAC parameters.

Table 11–1 DAC Parameters for Oracle Service Analytics

Parameter Name	Session Name	Description	Default Value
$$ASSIGNMENT_STATUS	Mplt_BC_ORA_ActivityFact	Used to sort the assignees based on assignment status.	Decode(assignment_status_id, 10,5, 14,4,3,3,35,2, 5,1)
$$ASSIGNMENT_STATUS_INCLUDE_LIST	Mplt_BC_ORA_ActivityFact	Assignments Statuses to be included to fetch assignee records.	(10,14,3,5, 35)
$$COST_TYPE	Mplt_BC_ORA_AgreeItemFact	Use this to define cost type.	N/A
$$MASTER_ORG	Mplt_BC_ORA_AssetFact	Use this to define one or more Organizations. It acts like a list.	204
$$MASTER_ORG	Mplt_BC_ORA_AssetDimension	Use this to define one or more Organizations. It acts like a list.	204
$$TRAVEL_MIN	Mplt_BC_ORA_ActivityFact	Used to determine the multiplier to convert Travel Time in Minutes based on Travel Duration UOM.	DECODE (ACTUAL_TRAVEL_DURATION_UOM, 'YR', 365*24*60,'QRT', 122*24*60, 'MTH', 30*24*60, 'WK', 7*24*60, 'DAY',24*60,'HR', 60,'MIN', 1)

11.2.2 Configuring Domain Values for Oracle Service Analytics With An Oracle EBS Source System

This section contains configuration steps that you need to perform to configure Oracle Service Analytics with an Oracle EBS source system, and contains the following topics:

- Section 11.2.2.1, "About Domain Values and CSV Worksheet Files for Oracle Service Analytics"
- Section 11.2.2.2, "How to Configure Activity Status"

- Section 11.2.2.3, "How to Configure Asset Status"
- Section 11.2.2.4, "How to Configure Agreement Status"
- Section 11.2.2.5, "How to Configure Service Request Status"
- Section 11.2.2.6, "How to Configure Activity Priority"
- Section 11.2.2.7, "How to Configure Activity Category"
- Section 11.2.2.8, "How to Configure Activity Type"
- Section 11.2.2.9, "How to Configure Activity Resolution"
- Section 11.2.2.10, "How to Configure Asset Cost Category"
- Section 11.2.2.11, "How to Configure Request Severity Code"
- Section 11.2.2.12, "How to Configure Service Request Area Code"

11.2.2.1 About Domain Values and CSV Worksheet Files for Oracle Service Analytics

To configure Oracle Marketing Analytics, you specify the domain values that you want to use (for example, for Activity Status, or Asset Status).

You configure Oracle Service Analytics by mapping domain values to columns in CSV files located in the $PMServer\LkpFiles folder (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

For more information on domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

Note: When editing CSV files, ensure that you:

- Do not 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.

11.2.2.2 How to Configure Activity Status

This section explains how to configure the Activity Status codes using the domainValues_Status_Activity_ora11i.csv configuration file.

Activity Status domain values are: Acknowledged, Approved, Attended, Accepted, Booked, Cancelled, Confirmed, Declined, Failed, Dispatched, Done, In Progress, On Hold, Open, Planned, Queued, Requested, Rescheduled, Scheduled, Assigned.

To configure Activity Status:

1. Identify the Activity Status codes that are used in your source system by using the following SQL:

```
SELECT distinct JTF_TASK_STATUSES_B.TASK_STATUS_ID STATUS_CODE, JTF_TASK_
STATUSES_TL.NAME STATUS_NAME
FROM JTF_TASK_STATUSES_B, JTF_TASK_STATUSES_TL
WHERE JTF_TASK_STATUSES_B.TASK_STATUS_ID = JTF_TASK_STATUSES_TL.TASK_STATUS_ID
AND JTF_TASK_STATUSES_TL.LANGUAGE= 'US'
```

2. Using a text editor, open the domainValues_Status_Activity_ora11i.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   You must ensure that the STATUS_CODE and STATUS_NAME values specified in the domainValues_Status_Activity_ora11i.csv file match the Activity Status codes that are used in your source system.

3. Edit the STATUS_CODE and STATUS_NAME values in the CSV file to match the Status Codes and Status Names values that you returned using the query in step 1.

   Map each Activity Status code to a W_STATUS_CODE and W_STATUS_DESC pair.

   Note: Do not change the default values in the other columns in the file (for example, W_STATUS_CODE.

4. Save and close the file.

11.2.2.3 How to Configure Asset Status

This section explains how to configure the Asset Status codes using the domainValues_Status_Asset_ora11i.csv configuration file.

Asset Status domain values are: Renewal, In Force, In Inventory, In Repair, In Transit, In Use, Inactive, Installed, Lapsed, Leased Loaner, Purchased, Release, Removed, Rented, Sold, Transferred.

To configure Asset Status:

1. Identify the Asset Status codes that are used in your source system by using the following SQL:

```
SELECT DISTINCT CSI_INSTANCE_STATUSES.INSTANCE_STATUS_ID STATUS_CODE, CSI_
INSTANCE_STATUSES.NAME STATUS_NAME
FROM CSI_INSTANCE_STATUSES
```

2. Using a text editor, open the domainValues_Status_Asset_ora11i.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   You must ensure that the STATUS_CODE and STATUS_NAME values specified in the domainValues_Status_Asset_ora11i.csv file match the Asset Status codes that are used in your source system.

3. Edit the STATUS_CODE and STATUS_NAME values in the CSV file to match the Status Codes and Status Names values that you returned using the query in step 1.

   Map each Asset Status code to a W_STATUS_CODE and W_STATUS_DESC pair.

   Note: Do not change the default values in the other columns in the file (for example, W_STATUS_CODE.

4. Save and close the file.

11.2.2.4 How to Configure Agreement Status

This section explains how to configure the Agreement Status codes using the domainValues_Status_AgreeItem_ora11i.csv configuration file.

Agreement Status domain values are: Active, Approved, Cancelled, Closed, Current, Draft, Expired, In Process, Inactive, Open, Pending, Quoted, Rejected, Signed, Suspended, Under Negotiation, On Hold.

To configure Agreement Status:

1. Identify the Agreement Status codes that are used in your source system by using the following SQL:

   ```
   SELECT DISTINCT OKC_STATUSES_B.CODE,
   OKC_STATUSES_TL.MEANING
   FROM OKC_STATUSES_B,OKC_STATUSES_TL
   WHERE OKC_STATUSES_B.CODE=OKC_STATUSES_TL.CODE
   AND OKC_STATUSES_TL.LANGUAGE='US'
   ```

2. Using a text editor, open the domainValues_Status_AgreeItem_ora11i.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   You must ensure that the STATUS_CODE and STATUS_NAME values specified in the domainValues_Status_AgreeItem_ora11i.csv file match the Agreement Status codes that are used in your source system.

3. Edit the STATUS_CODE and STATUS_NAME values in the CSV file to match the Status Codes and Status Names values that you returned using the query in step 1.

   Map each Agreement Status code to a W_STATUS_CODE and W_STATUS_DESC pair.

   Note: Do not change the default values in the other columns in the file (for example, W_STATUS_CODE.

4. Save and close the file.

11.2.2.5 How to Configure Service Request Status

This section explains how to configure the Service Request Status codes using the domainValues_Status_ServiceRequest_ora11i.csv configuration file.

Service Request Status domain values are: Closed, Approved, Cancelled, Closed, Completed, Exception Handling, In Progress, Not Started, Open, Pending, Quoted, Rejected, Revised, Submitted.

To configure Service Request Status:

1. Identify the Service Request Status codes that are used in your source system by using the following SQL:

   ```
   SELECT CS_INCIDENT_STATUSES_B.INCIDENT_STATUS_ID, CS_INCIDENT_STATUSES_TL.NAME
   FROM CS_INCIDENT_STATUSES_B, CS_INCIDENT_STATUSES_TL
   WHERE CS_INCIDENT_STATUSES_B.INCIDENT_STATUS_ID = CS_INCIDENT_STATUSES_
   TL.INCIDENT_STATUS_ID and CS_INCIDENT_STATUSES_TL.LANGUAGE = 'US'
   ```

2. Using a text editor, open the domainValues_Status_ServiceRequest_ora11i.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   You must ensure that the STATUS_CODE and STATUS_NAME values specified in the domainValues_Status_ServiceRequest_ora11i.csv file match the Service Request Status codes that are used in your source system.

3. Edit the STATUS_CODE and STATUS_NAME values in the CSV file to match the Status Codes and Status Names values that you returned using the query in step 1.

   Map each Service Request Status code to a W_STATUS_CODE and W_STATUS_DESC pair.

Note: Do not change the default values in the other columns in the file (for example, W_STATUS_CODE.

4. Save and close the file.

11.2.2.6 How to Configure Activity Priority

This section explains how to configure the Activity Priority codes using the domainValues_Xact_Types_Activity_Priority_ora11i.csv configuration file.

Activity Priority domain values are: 1-ASAP, 2-High, 3-Medium, 4-Low, 5-Optional, 6-Unprioritized.

To configure Activity Priority:

1. Identify the Activity Priority codes that are used in your source system by using the following SQL:

```
SELECT DISTINCT JTF_TASK_PRIORITIES_B.TASK_PRIORITY_ID
FROM JTF_TASK_PRIORITIES_B, JTF_TASK_PRIORITIES_TL
WHERE JTF_TASK_PRIORITIES_B.TASK_PRIORITY_ID = JTF_TASK_PRIORITIES_TL.TASK_
PRIORITY_ID AND JTF_TASK_PRIORITIES_TL.LANGUAGE = 'US'
```

2. Using a text editor, open the domainValues_Xact_Types_Activity_Priority_ora11i.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   You must ensure that the XACT_TYPE_CODE values specified in the domainValues_Xact_Types_Activity_Priority_ora11i.csv file match the Activity Priority codes that are used in your source system.

3. Edit the XACT_TYPE_CODE values in the CSV file to match the Activity Priority codes that you returned using the query in step 1.

   Map each Activity Priority code to a W_XACT_TYPE_CODE and W_XACT_TYPE_DESC pair.

   Note: Do not change the default values in the other columns in the file (for example, W_XACT_TYPE_CODE.

4. Save and close the file.

11.2.2.7 How to Configure Activity Category

This section explains how to configure the Activity Category codes using the domainValues_Xact_Types_Activity_Category_ora11i.csv configuration file.

Activity Category domain values are: Research, Follow-Up, Urgent, Outsourced, Customer Retention Activity, Sales Activity, Service Activity, Customer Satisfaction, Diagnostic, Field Engineer Activity, Other, Preventive Maintenance, Repair Activity.

To configure Activity Category:

1. Identify the Activity Category codes that are used in your source system by using the following SQL:

```
SELECT JTF_TASK_TYPES_B.TASK_TYPE_ID
FROM JTF_TASK_TYPES_B, JTF_TASK_TYPES_TL
WHERE JTF_TASK_TYPES_B.TASK_TYPE_ID=JTF_TASK_TYPES_TL.TASK_TYPE_ID AND JTF_
TASK_TYPES_TL.LANGUAGE = 'US'
```

2. Using a text editor, open the domainValues_Xact_Types_Activity_Category_ora11i.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

You must ensure that the XACT_TYPE_CODE values specified in the domainValues_Xact_Types_Activity_Category_ora11i.csv file match the Activity Category codes that are used in your source system.

3. Edit the XACT_TYPE_CODE values in the CSV file to match the Activity Category codes that you returned using the query in step 1.

   Map each Activity Category code to a W_XACT_TYPE_CODE code.

   Note: Do not change the default values in the other columns in the file (for example, W_XACT_TYPE_CODE.

4. Save and close the file.

11.2.2.8 How to Configure Activity Type

This section explains how to configure the Activity Type codes using the domainValues_Xact_Types_Activity_Type_ora11i.csv configuration file.

Activity Type domain values include: 'Call - Inbound', 'Call - Outbound', 'Email - Inbound', 'Email - Outbound', 'Fax - Inbound', 'Fax - Outbound'. For a full list, refer to the out-of-the-box version of the CSV file

To configure Activity Type:

1. Using a text editor, open the domainValues_Xact_Types_Activity_Type_ora11i.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

2. Edit the XACT_TYPE_CODE values in the CSV file to match the Activity Type codes that you want to deploy.

   Map each Activity Type code to a W_XACT_TYPE_CODE and W_XACT_TYPE_DESC pair.

3. Save and close the file.

11.2.2.9 How to Configure Activity Resolution

This section explains how to configure the Activity Resolution codes using the domainValues_Xact_Types_Activity_Resolution_ora11i.csv configuration file.

To configure Activity Resolution:

1. Identify the Activity Resolution codes that are used in your source system by using the following SQL:

   ```
   SELECT DISTINCT CS_LOOKUPS.LOOKUP_CODE
   FROM CS_LOOKUPS
   WHERE CS_LOOKUPS.LOOKUP_TYPE='REQUEST_RESOLUTION_CODE'
   ```

2. Using a text editor, open the domainValues_Xact_Types_Activity_Resolution_ora11i.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   You must ensure that the XACT_TYPE_CODE values specified in the domainValues_Xact_Types_Activity_Resolution_ora11i.csv file match the Activity Resolution codes that are used in your source system.

3. Edit the XACT_TYPE_CODE values in the CSV file to match the Activity Resolution codes that you returned using the query in step 1.

   Map each Activity Resolution code to a W_XACT_TYPE_CODE and W_XACT_TYPE_DESC pair.

Note: Do not change the default values in the other columns in the file (for example, W_XACT_TYPE_CODE.

4. Save and close the file.

11.2.2.10 How to Configure Asset Cost Category

This section explains how to configure the Asset Cost Category codes using the ASSET_COST_CATEGORY.csv configuration file.

To configure Asset Cost Category:

1. Using a text editor, open the ASSET_COST_CATEGORY.csv file, located in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).

   For each Asset Cost Category that you use, you need to specify a value for TYPE, NAME, VAL, LOW, and HIGH.

   For example:

   ```
   TYPE, NAME, VAL, LOW, HIGH
   ASSET_COST_CATEGORY,2501 - 5000,2501 - 5000,2501,5001
   ASSET_COST_CATEGORY,5001 - 7500,5001 - 7500,5001,7501
   ```

   Note: The language independent code is given in the NAME column and the translated value is given in the VAL column. The NAME and VAL values are pulled in the COST_CATEGORY_I and COST_CATEGORY Of W_ASSET_D respectively, depending on the ORIG_COST of the asset.

2. Save and close the file.

11.2.2.11 How to Configure Request Severity Code

This section explains how to configure the Request Severity Code codes using the domainValues_ServiceRequest_Severity_ora11i.csv configuration file.

Request Severity Codes are: '1-Critical', High, Low, Medium.

To configure Activity Resolution:

1. Identify the Activity Resolution codes that are used in your source system by using the following SQL:

   ```
   SELECT DISTINCT incident_severity_id Severity_Code, name from CS_INCIDENT_
   SEVERITIES_B
   ```

2. Using a text editor, open the domainValues_ServiceRequest_Severity_ora11i.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   You must ensure that the XACT_TYPE_CODE values specified in the domainValues_ServiceRequest_Severity_ora11i.csv file match the Activity Resolution codes that are used in your source system.

3. Edit the XACT_TYPE_CODE values in the CSV file to match the Activity Resolution codes that you returned using the query in step 1.

   Map each Activity Resolution code to a W_XACT_TYPE_CODE and W_XACT_TYPE_DESC pair.

   Note: Do not change the default values in the other columns in the file (for example, W_XACT_TYPE_CODE.

4. Save and close the file.

11.2.2.12 How to Configure Service Request Area Code

This section explains how to configure the Activity Resolution codes using the domainValues_ServiceRequest_Area_ora11i.csv configuration file.

Service Request Area Code values are: Authorization Request, Referral Request.

To configure Activity Resolution:

1. Identify the Activity Resolution codes that are used in your source system by using the following SQL:

   ```
   SELECT DISTINCT CS_INCIDENTS_ALL_B.INCIDENT_TYPE_ID from CS_INCIDENTS_ALL_B
   ```

2. Using a text editor, open the domainValues_ServiceRequest_Area_ora11i.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   You must ensure that the XACT_TYPE_CODE values specified in the domainValues_ServiceRequest_Area_ora11i.csv file match the Activity Resolution codes that are used in your source system.

3. Edit the XACT_TYPE_CODE values in the CSV file to match the Activity Resolution codes that you returned using the query in step 1.

   Map each Activity Resolution code to a W_XACT_TYPE_CODE and W_XACT_TYPE_DESC pair.

   Note: Do not change the default values in the other columns in the file (for example, W_XACT_TYPE_CODE.

4. Save and close the file.

11.2.3 Configuring Your RPD Repository for Oracle Service Analytics With An Oracle EBS Source System

This section contains security filters that you must enable in the RPD repository for Oracle Service Analytics.

Operating Unit Org-based security must be enabled for following service facts:

Table 11–2 Operating Unit Org-Based Security to Be Enabled for Service Facts

Name	Business Model Filter
"Core"."Fact - CRM - Service Request"	Core."Dim - Operating Unit Org"."Operating Unit Org Number" = VALUEOF(NQ_SESSION."OU_ORG")
"Core"."Fact - CRM - Agree Item"	Core."Dim - Operating Unit Org"."Operating Unit Org Number" = VALUEOF(NQ_SESSION."OU_ORG")

Inventory Org-based Security must be enabled for following facts:

Table 11–3 Inventory Org-Based Security to Be Enabled for Service Facts

Name	Business Model Filter
"Core"."Fact - CRM - Asset"	Core."Dim - Inventory Org"."Inventory Org Number" = VALUEOF(NQ_SESSION."INV_ORG")

11.3 Configuration Required Before A Full Load With Siebel CRM Source Systems

This section contains configuration steps that you need to perform on Oracle Service Analytics before you do a full data load using a Siebel CRM Source System. It contains the following topics:

- Section 11.3.1, "Configuring Domain Values for Oracle Service Analytics With A Siebel CRM Source System"

11.3.1 Configuring Domain Values for Oracle Service Analytics With A Siebel CRM Source System

This section contains configuration steps that you need to perform to configure Oracle Service Analytics with a Siebel CRM Version 8.1.1 source system, and contains the following topics:

- Section 11.3.1.1, "About Domain Values and CSV Worksheet Files for Oracle Service Analytics"
- Section 11.3.1.2, "How to Configure Activity Status"
- Section 11.3.1.3, "How to Configure Asset Status"
- Section 11.3.1.4, "How to Configure Agreement Status"
- Section 11.3.1.5, "How to Configure Service Request Status"
- Section 11.3.1.6, "How to Configure Activity Priority"
- Section 11.3.1.7, "How to Configure Activity Category"
- Section 11.3.1.8, "How to Configure Activity Type"
- Section 11.3.1.9, "How to Configure Activity Resolution"
- Section 11.3.1.10, "How to Configure Request Severity Code"
- Section 11.3.1.11, "How to Configure Service Request Area Code"

11.3.1.1 About Domain Values and CSV Worksheet Files for Oracle Service Analytics

To configure Oracle Service Analytics, you specify the domain values that you want to use (for example, for Activity Status, or Asset Status).

You configure Oracle Service Analytics by mapping domain values to columns in CSV files located in the $PMServer\LkpFiles folder (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

For more information on domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

Note: When editing CSV files, ensure that you:

- Do not 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.

11.3.1.2 How to Configure Activity Status

This section explains how to configure the Activity Status codes using the domainValues_Status_Activity_sbl.csv configuration file.

Activity Status domain values are: Acknowledged, Approved, Attended, Accepted, Booked, Cancelled, Confirmed, Declined, Failed, Dispatched, Done, In Progress, On Hold, Open, Planned, Queued, Requested, Rescheduled, Scheduled, Assigned.

To configure Activity Status:

1. Identify the Activity Status codes that are used in your source system by using the following SQL:

   ```
   SELECT DISTINCT S_LST_OF_VAL.NAME, S_LST_OF_VAL.VAL
   FROM S_LST_OF_VAL WHERE TYPE = 'EVENT_STATUS'
   AND LANG_ID = Language_Code
   ```

   Where *Language_Code* is a three-letter Language Code, for example, ENU for US English, or FRA for French.

2. Using a text editor, open the domainValues_Status_Activity_sbl.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   You must ensure that the STATUS_CODE and STATUS_NAME values specified in the domainValues_Status_Activity_sbl.csv file match the Activity Status codes that are used in your source system.

3. Edit the STATUS_CODE and STATUS_NAME values in the CSV file to match the Status Codes and Status Names values that you returned using the query in step 1.

   Map each Activity Status code to a W_STATUS_CODE and W_STATUS_DESC pair.

   Note: Do not change the default values in the other columns in the file (for example, W_STATUS_CODE.

4. Save and close the file.

11.3.1.3 How to Configure Asset Status

This section explains how to configure the Asset Status codes using the domainValues_Status_Asset_sbl.csv configuration file.

Asset Status domain values are: Renewal, In Force, In Inventory, In Repair, In Transit, In Use, Inactive, Installed, Lapsed, Leased Loaner, Purchased, Release, Removed, Rented, Sold, Transferred.

To configure Asset Status:

1. Identify the Asset Status codes that are used in your source system by using the following SQL:

   ```
   SELECT DISTINCT S_LST_OF_VAL.NAME, S_LST_OF_VAL.VAL
   FROM S_LST_OF_VAL WHERE TYPE = 'IMPL_PHASE'
   AND LANG_ID = '$$DFLT_LANG'
   ```

2. Using a text editor, open the domainValues_Status_Asset_sbl.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   You must ensure that the STATUS_CODE and STATUS_NAME values specified in the domainValues_Status_Asset_sbl.csv file match the Asset Status codes that are used in your source system.

3. Edit the STATUS_CODE and STATUS_NAME values in the CSV file to match the Status Codes and Status Names values that you returned using the query in step 1.

   Map each Asset Status code to a W_STATUS_CODE and W_STATUS_DESC pair.

   Note: Do not change the default values in the other columns in the file (for example, W_STATUS_CODE.

4. Save and close the file.

11.3.1.4 How to Configure Agreement Status

This section explains how to configure the Agreement Status codes using the domainValues_Status_AgreeItem_sbl.csv configuration file.

Agreement Status domain values are: Active, Approved, Cancelled, Closed, Current, Draft, Expired, In Process, Inactive, Open, Pending, Quoted, Rejected, Signed, Suspended, Under Negotiation, On Hold.

To configure Agreement Status:

1. Identify the Agreement Status codes that are used in your source system by using the following SQL:

   ```
   SELECT DISTINCT S_LST_OF_VAL.NAME, S_LST_OF_VAL.VAL
   FROM S_LST_OF_VAL WHERE TYPE = 'SRV_AGREE_STATUS'
   AND LANG_ID = '$$DFLT_LANG'
   ```

2. Using a text editor, open the domainValues_Status_AgreeItem_sbl.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   You must ensure that the STATUS_CODE and STATUS_NAME values specified in the domainValues_Status_AgreeItem_sbl.csv file match the Agreement Status codes that are used in your source system.

3. Edit the STATUS_CODE and STATUS_NAME values in the CSV file to match the Status Codes and Status Names values that you returned using the query in step 1.

   Map each Agreement Status code to a W_STATUS_CODE and W_STATUS_DESC pair.

   Note: Do not change the default values in the other columns in the file (for example, W_STATUS_CODE.

4. Save and close the file.

11.3.1.5 How to Configure Service Request Status

This section explains how to configure the Service Request Status codes using the domainValues_Status_ServiceRequest_sbl.csv configuration file.

Service Request Status domain values are: Closed, Approved, Cancelled, Closed, Completed, Exception Handling, In Progress, Not Started, Open, Pending, Quoted, Rejected, Revised, Submitted.

To configure Service Request Status:

1. Identify the Service Request Status codes that are used in your source system by using the following SQL:

   ```
   SELECT DISTINCT S_LST_OF_VAL.NAME, S_LST_OF_VAL.VAL
   FROM S_LST_OF_VAL WHERE TYPE = 'SR_STATUS
   AND LANG_ID = '$$DFLT_LANG'
   ```

2. Using a text editor, open the domainValues_Status_ServiceRequest_sbl.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   You must ensure that the STATUS_CODE and STATUS_NAME values specified in the domainValues_Status_ServiceRequest_sbl.csv file match the Service Request Status codes that are used in your source system.

3. Edit the STATUS_CODE and STATUS_NAME values in the CSV file to match the Status Codes and Status Names values that you returned using the query in step 1.

   Map each Service Request Status code to a W_STATUS_CODE and W_STATUS_DESC pair.

   Note: Do not change the default values in the other columns in the file (for example, W_STATUS_CODE.

4. Save and close the file.

11.3.1.6 How to Configure Activity Priority

This section explains how to configure the Activity Priority codes using the domainValues_Xact_Types_Activity_Priority_sbl.csv configuration file.

Activity Priority domain values are: 1-ASAP, 2-High, 3-Medium, 4-Low, 5-Optional, 6-Unprioritized.

To configure Activity Priority:

1. Identify the Activity Priority codes that are used in your source system by using the following SQL:

   ```
   SELECT DISTINCT S_LST_OF_VAL.NAME, S_LST_OF_VAL.VAL
   FROM S_LST_OF_VAL WHERE TYPE = 'ACTIVITY_PRIORITY'
   AND LANG_ID = '$$DFLT_LANG'
   ```

2. Using a text editor, open the domainValues_Xact_Types_Activity_Priority_sbl.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   You must ensure that the XACT_TYPE_CODE values specified in the domainValues_Xact_Types_Activity_Priority_sbl.csv file match the Activity Priority codes that are used in your source system.

3. Edit the XACT_TYPE_CODE values in the CSV file to match the Activity Priority codes that you returned using the query in step 1.

   Map each Activity Priority code to a W_XACT_TYPE_CODE and W_XACT_TYPE_DESC pair.

   Note: Do not change the default values in the other columns in the file (for example, W_XACT_TYPE_CODE.

4. Save and close the file.

11.3.1.7 How to Configure Activity Category

This section explains how to configure the Activity Category codes using the domainValues_Xact_Types_Activity_Category_sbl.csv configuration file.

Activity Category domain values are: Research, Follow-Up, Urgent, Outsourced, Customer Retention Activity, Sales Activity, Service Activity, Customer Satisfaction, Diagnostic, Field Engineer Activity, Other, Preventive Maintenance, Repair Activity.

To configure Activity Category:

1. Identify the Activity Category codes that are used in your source system by using the following SQL:

   ```
   SELECT DISTINCT S_LST_OF_VAL.NAME, S_LST_OF_VAL.VAL
   FROM S_LST_OF_VAL WHERE TYPE = 'FS_ACTIVITY_CLASS'
   AND LANG_ID = '$$DFLT_LANG'
   ```

2. Using a text editor, open the domainValues_Xact_Types_Activity_Category_sbl.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   You must ensure that the XACT_TYPE_CODE values specified in the domainValues_Xact_Types_Activity_Category_sbl.csv file match the Activity Category codes that are used in your source system.

3. Edit the XACT_TYPE_CODE values in the CSV file to match the Activity Category codes that you returned using the query in step 1.

   Map each Activity Category code to a W_XACT_TYPE_CODE code.

   Note: Do not change the default values in the other columns in the file (for example, W_XACT_TYPE_CODE.

4. Save and close the file.

11.3.1.8 How to Configure Activity Type

This section explains how to configure the Activity Type codes using the domainValues_Xact_Types_Activity_Type_sbl.csv configuration file.

Activity Type domain values include: 'Call - Inbound', 'Call - Outbound', 'Email - Inbound', 'Email - Outbound', 'Fax - Inbound', 'Fax - Outbound'. For a full list, refer to the out-of-the-box version of the CSV file.

To configure Activity Type:

1. Identify the Activity Resolution codes that are used in your source system by using the following SQL:

   ```
   SELECT DISTINCT S_LST_OF_VAL.NAME, S_LST_OF_VAL.VAL
   FROM S_LST_OF_VAL WHERE TYPE = 'TODO_TYPE'
   AND LANG_ID = '$$DFLT_LANG'
   ```

2. Using a text editor, open the domainValues_Xact_Types_Activity_Type_sbl.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

3. Edit the XACT_TYPE_CODE values in the CSV file to match the Activity Type codes that you want to deploy.

   Map each Activity Type code to a W_XACT_TYPE_CODE and W_XACT_TYPE_DESC pair.

4. Save and close the file.

11.3.1.9 How to Configure Activity Resolution

This section explains how to configure the Activity Resolution codes using the domainValues_Xact_Types_Activity_Resolution_sbl.csv configuration file.

To configure Activity Resolution:

1. Identify the Activity Resolution codes that are used in your source system by using the following SQL:

   ```
   SELECT DISTINCT S_LST_OF_VAL.NAME, S_LST_OF_VAL.VAL
   FROM S_LST_OF_VAL WHERE TYPE = 'SR_RESOLUTION
   AND LANG_ID = '$$DFLT_LANG'
   ```

2. Using a text editor, open the domainValues_Xact_Types_Activity_Resolution_sbl.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   You must ensure that the XACT_TYPE_CODE values specified in the domainValues_Xact_Types_Activity_Resolution_sbl.csv file match the Activity Resolution codes that are used in your source system.

3. Edit the XACT_TYPE_CODE values in the CSV file to match the Activity Resolution codes that you returned using the query in step 1.

   Map each Activity Resolution code to a W_XACT_TYPE_CODE and W_XACT_TYPE_DESC pair.

   Note: Do not change the default values in the other columns in the file (for example, W_XACT_TYPE_CODE.

4. Save and close the file.

11.3.1.10 How to Configure Request Severity Code

This section explains how to configure the Request Severity Code codes using the domainValues_ServiceRequest_Severity_sbl.csv configuration file.

To configure Activity Resolution:

1. Identify the Activity Resolution codes that are used in your source system by using the following SQL:

   ```
   SELECT DISTINCT S_LST_OF_VAL.NAME, S_LST_OF_VAL.VAL
   FROM S_LST_OF_VAL WHERE TYPE = 'SR_SEVERITY
   AND LANG_ID = '$$DFLT_LANG'
   ```

2. Using a text editor, open the domainValues_ServiceRequest_Severity_sbl.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   You must ensure that the XACT_TYPE_CODE values specified in the domainValues_ServiceRequest_Severity_sbl.csv file match the Activity Resolution codes that are used in your source system.

3. Edit the XACT_TYPE_CODE values in the CSV file to match the Activity Resolution codes that you returned using the query in step 1.

   Map each Activity Resolution code to a W_XACT_TYPE_CODE and W_XACT_TYPE_DESC pair.

   Note: Do not change the default values in the other columns in the file (for example, W_XACT_TYPE_CODE.

4. Save and close the file.

11.3.1.11 How to Configure Service Request Area Code

This section explains how to configure the Activity Resolution codes using the domainValues_ServiceRequest_Area_sbl.csv configuration file.

To configure Activity Resolution:

1. Identify the Activity Resolution codes that are used in your source system by using the following SQL:

   ```
   SELECT DISTINCT S_LST_OF_VAL.NAME, S_LST_OF_VAL.VAL
   FROM S_LST_OF_VAL WHERE TYPE = 'SR_AREA'
   AND LANG_ID = '$$DFLT_LANG'
   ```

2. Using a text editor, open the domainValues_ServiceRequest_Area_sbl.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   You must ensure that the XACT_TYPE_CODE values specified in the domainValues_ServiceRequest_Area_sbl.csv file match the Activity Resolution codes that are used in your source system.

3. Edit the XACT_TYPE_CODE values in the CSV file to match the Activity Resolution codes that you returned using the query in step 1.

   Map each Activity Resolution code to a W_XACT_TYPE_CODE and W_XACT_TYPE_DESC pair.

   Note: Do not change the default values in the other columns in the file (for example, W_XACT_TYPE_CODE).

4. Save and close the file.

12

Configuring Oracle Marketing Analytics

This section describes how to configure Oracle Marketing Analytics and contains the following topics:

- Section 12.1, "Overview of Oracle Marketing Analytics"
- Section 12.2, "Configuration Required Before A Full Load"

12.1 Overview of Oracle Marketing Analytics

Oracle Marketing Analytics is a comprehensive analytic solution that provides timely fact-based insight into the marketing activities of the entire organization. It provides new levels of information richness, usability, and reach to marketing professionals throughout the enterprise. It provides actionable intelligence in the following Marketing areas:

- Marketing Planning
- Marketing Effectiveness
- Customer Insight (B2B and B2C)
- Marketing Events
- Leads

12.2 Configuration Required Before A Full Load

This section contains configuration steps that you need to perform on Oracle Marketing Analytics before you do a full data load. It contains the following topic:

- Section 12.2.1, "Configuring Oracle Marketing Analytics For Siebel Version 8.1.1"

12.2.1 Configuring Oracle Marketing Analytics For Siebel Version 8.1.1

This section contains configuration steps that you need to perform to configure Oracle Marketing Analytics for Siebel CRM Version 8.1.1, and contains the following topics:

- Section 12.2.1.1, "About Domain Values and CSV Worksheet Files for Oracle Marketing Analytics"
- Section 12.2.1.2, "How to Configure the Lead Quality Codes"
- Section 12.2.1.3, "How to Configure Lead Status Codes"

12.2.1.1 About Domain Values and CSV Worksheet Files for Oracle Marketing Analytics

To configure Oracle Marketing Analytics, you specify the Lead Quality codes and Lead Status codes that you want to use.

You configure Oracle Marketing Analytics by mapping domain values to columns in CSV files located in the $PMServer\LkpFiles folder (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

Table 12–1 lists the CSV worksheet files and the domain values for Oracle Marketing Analytics in the $PMServer\LkpFiles folder.

Table 12–1 Domain Values and CSV Worksheet Files for Oracle Marketing Analytics

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_MarketingLeadQuality_SEBL81.csv	W_MKTG_LEAD_D.W_QUALITY_CODE	Lists the lead quality codes and their corresponding domain values for the Siebel 8.1.1 Application.	SDE_SBL_MarketingLeadDimension
domainValues_MarketingLeadStatus_SEBL81.csv	W_STATUS_D.W_STATUS_CODE	Lists the lead status codes and their corresponding domain values for the Siebel 8.1.1 Application.	SDE_SBL_StatusDimension_MarketingLeadStatus

For more information on domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

Note: When editing CSV files:

- You can add new values to W_ columns or map new values to existing W_ columns. Do not change the W_ values in the spreadsheet that is installed out-of-the-box.
- Do not change the case of values in the CSV file.

  For example, do not change 'CONTRACTOR' to 'Contractor'.

12.2.1.2 How to Configure the Lead Quality Codes

This section explains how to configure the Lead Quality codes using the domainValues_MarketingLeadQuality_SEBL81.csv configuration file.

1. Identify the Lead Quality Codes in your source system by using the following SQL:

   ```
   SELECT distinct NAME FROM S_LST_OF_VAL WHERE TYPE = 'LEAD_QUALITY'
   ```

2. Using a text editor, open the domainValues_MarketingLeadQuality_SEBL81.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   You must ensure that the Lead Quality Codes used in your source match the QUALITY_CODE values specified in the domainValues_MarketingLeadQuality_SEBL81.csv file.

3. Edit the QUALITY_CODE values in the CSV file to match the NAME values that you returned using the query in step 1.

   If required, you can add new QUALITY_CODE, W_QUALITY_CODE, and W_QUALITY_DESC values to this CSV file.

   Note: Do not change the default values in the other columns in the file (for example, W_QUALITY_CODE or W_QUALITY_DESC).

   For example, if your source system uses 'Good' as a Quality Code rather than the out-of-the-box value 'High', you might change the appropriate row from '3-High,3-High,3-High' to '3-Good,3-High,3-High'.

4. Save and close the file.

12.2.1.3 How to Configure Lead Status Codes

This section explains how to configure the Lead Status codes using the domainValues_MarketingLeadStatus_SEBL81.csv configuration file.

1. Identify the Lead Status Codes in your source system by using the following SQL:

   ```
   SELECT distinct NAME FROM S_LST_OF_VAL WHERE TYPE = 'LEAD_STATUS'
   ```

2. Using a text editor, open the domainValues_MarketingLeadStatus_SEBL81.csv file, located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   You must ensure that the Lead Status Codes used in your source match the STATUS_CODE values specified in the domainValues_MarketingLeadStatus_SEBL81.csv file.

3. Edit the STATUS_CODE values in the CSV file to match the NAME values that you returned using the query in step 1.

   If required, you can also add new STATUS_CODE, W_STATUS_CODE, and W_STATUS_DESC values to this CSV file.

   Note: Do not change the default values in the other columns in the file (for example, W_STATUS_CODE or W_STATUS_DESC).

   For example, if your source system uses 'Not Qualified' as a Status Code rather than the out-of-the-box value 'Unqualified', you might change the appropriate row from 'Unqualified, Unqualified, Unqualified' to 'Not Qualified, Unqualified, Unqualified'.

4. Save and close the file.

13

Configuring Oracle Loyalty Analytics

This section describes how to configure Oracle Loyalty Analytics used with a Siebel CRM source system. It contains the following topics:

- Section 13.1, "Overview of Oracle Loyalty Analytics"
- Section 13.2, "Configuration Required Before a Full Load"
- Section 13.3, "Configuration Steps for Controlling Your Data Set"
- Section 13.4, "Configuring SQL Transformations in Informatica for Database Dependencies"

13.1 Overview of Oracle Loyalty Analytics

Oracle Loyalty Analytics helps you analyze the effectiveness of your customer loyalty programs. It is designed to help loyalty organizations leverage the power of timely, actionable information to improve the quality of decisions and ultimately optimize performance. Oracle Loyalty Analytics provides insight into the following Siebel Loyalty areas:

- Loyalty programs
- Membership activities
- Transactions
- Effectiveness of promotions
- Partner activities
- Finance

Oracle Loyalty Analytics supports Siebel 8.1.1 and the Universal adapter.

13.1.1 Oracle Loyalty Analytics Roles

The RPD contains preconfigured roles for Oracle Loyalty Analytics dashboard users. Each role is based on a particular Siebel Loyalty role and the responsibilities defined for that role. The roles determine which dashboards users can access. Table 13–1 describes the roles in the RPD for Oracle Loyalty Analytics.

Table 13–1 Oracle Loyalty Analytics Roles

Role	Description	Experience Level	Usability Requirements and Expectations
Loyalty Marketing Manager	This person is responsible for the overall health and status of the Loyalty program. Responsible for creating loyalty promotions and measuring the effectiveness of the program and promotions.	This person is an experienced and savvy marketing manager.	Expects to review the current status of the Loyalty Program on a daily basis. Expects to review the Loyalty Program Membership trends and Member activity regularly and to measure the effectiveness of the loyalty program and promotions.
Loyalty Partner Manager	This person is responsible for managing the partners and partner relationships in Loyalty Program. This person decides who should stay a partner in the program.	This person is an experienced marketing manager who is responsible for maintaining relationships with the partners.	Expects to be able to review the current status of partner accounts and points liability. Expects to be able to review financial aspects of partner billing and member activity with respect to the partner products.
Loyalty Program General Manager	Business head of the Loyalty program and is responsible for the program's effectiveness in achieving the underlying CRM and business goals.	An experienced user.	The Loyalty program GM is expected to monitor the following areas: ■ Promotion effectiveness ■ Outstanding program liability ■ Partner profitability ■ Effectiveness of Loyalty program offerings (Products and Services offered for Accrual and Redemption)
Loyalty Administrator	Administrator	An experienced user.	This user has access to all dashboards and reports.

13.2 Configuration Required Before a Full Load

This section contains configuration steps that you need to perform on Oracle Loyalty Analytics before you do a full data load. It contains the following topics:

- Section 13.2.1, "About Configuring Domain Values and CSV Worksheet Files for Oracle Loyalty Analytics"
- Section 13.2.2, "How to Configure the domainValues_Loyalty_Statement_Type.csv"
- Section 13.2.3, "How to Configure the domainValues_Member_Status.csv"
- Section 13.2.4, "How to Configure the domainValues_Transaction_Status_Types.csv"
- Section 13.2.5, "How to Configure the domainValues_Loyalty_Tier_movement_Type.csv"
- Section 13.2.6, "How to Configure the file_Redeemed_Unit_Cost.csv"

Note: In addition to the configuration information in this section, refer to the *Oracle Loyalty Analytics Bus Matrix* on the My Oracle Support Web site for detailed information about the dimensions that are used for the Oracle Loyalty Analytics metrics.

13.2.1 About Configuring Domain Values and CSV Worksheet Files for Oracle Loyalty Analytics

Table 13–2 lists the CSV worksheet files and the domain values for Oracle Loyalty Analytics.

Table 13–2 Domain Values and CSV Worksheet Files for Oracle Loyalty Analytics

Worksheet File Name	Description	Session
domainValues_Loyalty	Lists the Loyalty Statement Types and corresponding domain values for the Siebel application.	SDE_SBL_LoyStatementFact
domainValues_Member_Status.csv	Lists the Member Statuses and corresponding domain values for the Siebel application.	SDE_SBL_LoyCodeDimension_MemberStatus
domainValues_Transaction_Status_Types.csv	Lists the Transaction Status Types, Transaction Types, and Transaction Sub Types and corresponding domain values for the Siebel application.	SDE_SBL_LoyTransactionIdDimension SDE_SBL_LoyActivityFact
domainValues_Loyalty_Tier_movement_Type.csv	Lists the Loyalty Tier Movement Types and corresponding domain values for the Siebel application.	SIL_LoyTierMoveTypeDimension SIL_LoyTierMoveTypeDimension_Full
file_Redeemed_Unit_Cost.csv	Lists the Unit Rates and corresponding domain values for the Siebel application.	SDE_SBL_LoyCostFact

13.2.2 How to Configure the domainValues_Loyalty_Statement_Type.csv

This section explains how to configure the domainValues_Loyalty_Statement_Type.csv.

1. Identify the Loyalty Statement Type in your Oracle source system by using the following SQL:

```
SELECT
S_LST_OF_VAL.CREATED,
S_LST_OF_VAL.LAST_UPD,
S_LST_OF_VAL.LAST_UPD_BY,
S_LST_OF_VAL.NAME,
S_LST_OF_VAL.VAL,
S_LST_OF_VAL.LANG_ID,
S_LST_OF_VAL.CREATED_BY
FROM
S_LST_OF_VAL
WHERE
TYPE = 'LOY_STMT_TYPE_CD'
```

2. Using a text editor, open the domainValues_Loyalty_Statement_Type.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the values in the VAL column to the STMT_TYPE_CODE and W_STMT_TYPE_CODE columns in the CSV file.
4. Copy the values in the NAME column to the STMT_TYPE_NAME and W STMT_TYPE_NAME columns in the CSV file.
5. Save and close the file.

Table 13–3 Sample Domain Values Defined in the domainValues_Loyalty_Statement_Type.csv File

STMT_TYPE_ CODE	STMT_TYPE_ NAME	W_STMT_TYPE_ CLASS	W_STMT_TYPE_ CODE	W_STMT_TYPE_ NAME
Accrual Dispute	Accrual Dispute	STATEMENT_TYPE	Accrual Dispute	Accrual Dispute
Member Gift Points	Member Gift Points	STATEMENT_TYPE	Member Gift Points	Member Gift Points
Member Points Adjustments	Member Points Adjustments	STATEMENT_TYPE	Member Points Adjustments	Member Points Adjustments
Member Points Purchase	Member Points Purchase	STATEMENT_TYPE	Member Points Purchase	Member Points Purchase
Member Points Transfer	Member Points Transfer	STATEMENT_TYPE	Member Points Transfer	Member Points Transfer
Member Statement	Member Statement	STATEMENT_TYPE	Member Statement	Member Statement
Membership Cancellation	Membership Cancellation	STATEMENT_TYPE	Membership Cancellation	Membership Cancellation
Membership Enrollment	Membership Enrollment	STATEMENT_TYPE	Membership Enrollment	Membership Enrollment
Membership Renewal	Membership Renewal	STATEMENT_TYPE	Membership Renewal	Membership Renewal
Partner Billing	Partner Billing	STATEMENT_TYPE	Partner Billing	Partner Billing
Partner Statement	Partner Statement	STATEMENT_TYPE	Partner Statement	Partner Statement
Redemption Voucher Issue	Redemption Voucher Issue	STATEMENT_TYPE	Redemption Voucher Issue	Redemption Voucher Issue
Status of Membership	Status of Membership	STATEMENT_TYPE	Status of Membership	Status of Membership
Tier Change	Tier Change	STATEMENT_TYPE	Tier Change	Tier Change
Voucher Cancellation	Voucher Cancellation	STATEMENT_TYPE	Voucher Cancellation	Voucher Cancellation

13.2.3 How to Configure the domainValues_Member_Status.csv

This section explains how to configure the domainValues_Member_Status.csv file.

1. Identify the Member Status in your Oracle source system by using the following SQL:

```
SELECT S_LST_OF_VAL.CREATED,
S_LST_OF_VAL.LAST_UPD,
S_LST_OF_VAL.LAST_UPD_BY,
S_LST_OF_VAL.NAME,
S_LST_OF_VAL.VAL,
S_LST_OF_VAL.LANG_ID,
S_LST_OF_VAL.CREATED_BY
FROM
S_LST_OF_VAL
WHERE
TYPE = 'LOY_MEMBER_STATUS'
```

2. Using a text editor, open the domainValues_Member_Status.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

3. Copy the values in the VAL column to the STATUS_CODE and W_STATUS_CODE columns in the CSV file.
4. Copy the values in the NAME column to the STATUS_NAME and W STATUS_NAME columns in the CSV file.
5. Save and close the file.

Table 13–4 Sample Domain Values Defined in the domainValues_Member_Status.csv File

STATUS_CODE	STATUS_NAME	W_STATUS_CLASS	W_STATUS_CODE	W_STATUS_NAME
Active	Active	LOY_MEMBER_STATUS	ACTIVE	Active
Inactive	Inactive	LOY_MEMBER_STATUS	INACTIVE	Inactive
Pending	Pending	LOY_MEMBER_STATUS	ACTIVE	Active
Cancelled	Cancelled	LOY_MEMBER_STATUS	CANCELLED	Cancelled
Payment Pending	Payment Pending	LOY_MEMBER_STATUS	ACTIVE	Active
Dormant	Dormant	LOY_MEMBER_STATUS	INACTIVE	Inactive
Merged	Merged	LOY_MEMBER_STATUS	ACTIVE	Active
Pre Allotted	Pre Allotted	LOY_MEMBER_STATUS	ACTIVE	Active

13.2.4 How to Configure the domainValues_Transaction_Status_Types.csv

This section explains how to configure the domainValues_Transaction_Status_types.csv file.

1. Identify Transaction Status Types, Transaction Types, and Transaction Sub Types in your Oracle source system by entering the first of four SQL statements in this set of steps:

```
SELECT S_LST_OF_VAL.CREATED,
S_LST_OF_VAL.LAST_UPD,
S_LST_OF_VAL.LAST_UPD_BY,
S_LST_OF_VAL.NAME,
S_LST_OF_VAL.VAL,
S_LST_OF_VAL.LANG_ID,
S_LST_OF_VAL.CREATED_BY
FROM
S_LST_OF_VAL
WHERE
TYPE = 'LOY_TXN_STATUS_CD'
```

2. Using a text editor, open the domainValues_Transaction_Status_Types.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the values in the VAL column to the STATUS_CODE and W_STATUS_CODE columns in the CSV file.

4. Copy the values in the NAME column to the STATUS_NAME and W STATUS_NAME columns in the CSV file.

5. Enter this SQL statement:

```
SELECT S_LST_OF_VAL.CREATED,
S_LST_OF_VAL.LAST_UPD,
S_LST_OF_VAL.LAST_UPD_BY,
S_LST_OF_VAL.NAME,
S_LST_OF_VAL.VAL,
S_LST_OF_VAL.LANG_ID,
S_LST_OF_VAL.CREATED_BY
FROM
S_LST_OF_VAL
WHERE
TYPE = 'LOY_TXN_TYPE_CD'
```

6. Copy the values in the VAL column to the STATUS_CODE and W_STATUS_CODE columns in the CSV file.

7. Copy the values in the NAME column to the STATUS_NAME and W STATUS_NAME columns in the CSV file.

8. Enter this SQL statement:

```
SELECT
S_LST_OF_VAL.CREATED,
S_LST_OF_VAL.LAST_UPD,
S_LST_OF_VAL.LAST_UPD_BY,
S_LST_OF_VAL.NAME,
S_LST_OF_VAL.VAL,
S_LST_OF_VAL.LANG_ID,
S_LST_OF_VAL.CREATED_BY
FROM
S_LST_OF_VAL
WHERE
TYPE = 'LOY_TXN_SUB_TYPE_CD'
```

9. Copy the values in the VAL column to the STATUS_CODE and W_STATUS_CODE columns in the CSV file.

10. Copy the values in the NAME column to the STATUS_NAME and W STATUS_NAME columns in the CSV file.

11. Enter this SQL statement:

```
SELECT S_LST_OF_VAL.CREATED,
S_LST_OF_VAL.LAST_UPD,
S_LST_OF_VAL.LAST_UPD_BY,
S_LST_OF_VAL.NAME,
S_LST_OF_VAL.VAL,
S_LST_OF_VAL.LANG_ID,
S_LST_OF_VAL.CREATED_BY
FROM
S_LST_OF_VAL
WHERE
TYPE = 'LOY_REDEMPTION_TYPE_CD'
```

12. Copy the values in the VAL column to the STATUS_CODE and W_STATUS_CODE columns in the CSV file.

13. Copy the values in the NAME column to the STATUS_NAME and W STATUS_NAME columns in the CSV file.

14. Save and close the file.

    The following tables show sample domain values defined in the domainValues_Transaction_Status_Types.csv for these Loyalty status classes:

    - Transaction Status
    - Transaction Status Type
    - Transaction Status Sub Type
    - Redemption Type

Table 13–5 Sample Domain Values for Transaction Status Defined in the domainValues_Transaction_Status_Types.csv File

STATUS_CODE	STATUS_NAME	W_STATUS_CLASS	W_STATUS_CODE	W_STATUS_NAME
Cancel - Pending	Cancel - Pending	LOY_TXN_STATUS_CD	Cancel - Pending	Cancel - Pending
Rejected - Manager	Rejected - Manager	LOY_TXN_STATUS_CD	Rejected - Manager	Rejected - Manager
Submitted	Submitted	LOY_TXN_STATUS_CD	Submitted	Submitted
Processed	Processed	LOY_TXN_STATUS_CD	Processed	Processed
Accepted	Accepted	LOY_TXN_STATUS_CD	Accepted	Accepted
Cancelled	Cancelled	LOY_TXN_STATUS_CD	Cancelled	Cancelled
Partner Pending	Partner Pending	LOY_TXN_STATUS_CD	Partner Pending	Partner Pending
Queued	Queued	LOY_TXN_STATUS_CD	Queued	Queued
Rejected - Engine	Rejected - Engine	LOY_TXN_STATUS_CD	Rejected - Engine	Rejected - Engine
In Progress	In Progress	LOY_TXN_STATUS_CD	In Progress	In Progress
Pending Next Approval	Pending Next Approval	LOY_TXN_STATUS_CD	Pending Next Approval	Pending Next Approval
Unspecified	Unspecified	LOY_TXN_STATUS_CD	Unspecified	Unspecified
Approved	Approved	LOY_TXN_STATUS_CD	Approved	Approved
Acceptable	Acceptable	LOY_TXN_STATUS_CD	Acceptable	Acceptable
Manually Cancelled	Manually Cancelled	LOY_TXN_STATUS_CD	Manually Cancelled	Manually Cancelled
Partner Approved	Partner Approved	LOY_TXN_STATUS_CD	Partner Approved	Partner Approved
Partner Rejected	Partner Rejected	LOY_TXN_STATUS_CD	Partner Rejected	Partner Rejected
Unspecified	Unspecified	LOY_TXN_STATUS_CD	Unspecified	Unspecified

Table 13–6 Sample Domain Values for Transaction Status Type Defined in the domainValues_Transaction_Status_Types.csv File

STATUS_CODE	STATUS_NAME	W_STATUS_ CLASS	W_STATUS_CODE	W_STATUS_NAME
Accrual	ACCRUAL	LOY_TXN_TYPE_ CD	Accrual	Accrual
Unspecified	Unspecified	LOY_TXN_TYPE_ CD	Unspecified	Unspecified
Redemption	REDEMPTION	LOY_TXN_TYPE_ CD	Redemption	Redemption
Service	SERVICE	LOY_TXN_TYPE_ CD	Service	Service
Partner Price	PARTNER_PRICE	LOY_TXN_TYPE_ CD	Partner Price	Partner Price

Table 13–7 Sample Domain Values for Transaction Status Sub Type Defined in the domainValues_Transaction_Status_Types.csv File

STATUS_CODE	STATUS_NAME	W_STATUS_ CLASS	W_STATUS_CODE	W_STATUS_NAME
Availment	Availment	LOY_TXN_SUB_ TYPE_CD	Availment	Availment
Incorrect Accrual	Incorrect Accrual	LOY_TXN_SUB_ TYPE_CD	Incorrect Accrual	Incorrect Accrual
Manual Credit	Manual Credit	LOY_TXN_SUB_ TYPE_CD	Manual Credit	Manual Credit
Point Purchase	Point Purchase	LOY_TXN_SUB_ TYPE_CD	Point Purchase	Point Purchase
Manual Debit	Manual Debit	LOY_TXN_SUB_ TYPE_CD	Manual Debit	Manual Debit
Reactivation	Reactivation	LOY_TXN_SUB_ TYPE_CD	Reactivation	Reactivation
Transfer	Transfer	LOY_TXN_SUB_ TYPE_CD	Transfer	Transfer
Transfer Points	Transfer Points	LOY_TXN_SUB_ TYPE_CD	Transfer Points	Transfer Points
Cancellation	Cancellation	LOY_TXN_SUB_ TYPE_CD	Cancellation	Cancellation
Lounge Purchase	Lounge Purchase	LOY_TXN_SUB_ TYPE_CD	Lounge Purchase	Lounge Purchase
PromotionEnrolment	PromotionEnrolment	LOY_TXN_SUB_ TYPE_CD	PromotionEnrolment	PromotionEnrolment
Voucher Reissue	Voucher Reissue	LOY_TXN_SUB_ TYPE_CD	Voucher Reissue	Voucher Reissue
Unspecified	Unspecified	LOY_TXN_SUB_ TYPE_CD	Unspecified	Unspecified
Enrolment	Enrolment	LOY_TXN_SUB_ TYPE_CD	Enrolment	Enrolment
Loan	Loan	LOY_TXN_SUB_ TYPE_CD	Loan	Loan

Table 13–7 (Cont.) Sample Domain Values for Transaction Status Sub Type Defined in the domainValues_Transaction_Status_Types.csv File

STATUS_CODE	STATUS_NAME	W_STATUS_CLASS	W_STATUS_CODE	W_STATUS_NAME
Auto Accrual	Auto Accrual	LOY_TXN_SUB_TYPE_CD	Auto Accrual	Auto Accrual
Gift	Gift	LOY_TXN_SUB_TYPE_CD	Gift	Gift
Missing Accrual	Missing Accrual	LOY_TXN_SUB_TYPE_CD	Missing Accrual	Missing Accrual
Loan Repayment	Loan Repayment	LOY_TXN_SUB_TYPE_CD	Loan Repayment	Loan Repayment
Membership Renewal	Membership Renewal	LOY_TXN_SUB_TYPE_CD	Membership Renewal	Membership Renewal
Product	Product	LOY_TXN_SUB_TYPE_CD	Product	Product
Gift Points	Gift Points	LOY_TXN_SUB_TYPE_CD	Gift Points	Gift Points
Membership Cancellation	Membership Cancellation	LOY_TXN_SUB_TYPE_CD	Membership Cancellation	Membership Cancellation

Table 13–8 Sample Domain Values for Redemption Type Defined in the domainValues_Transaction_Status_Types.csv File

STATUS_CODE	STATUS_NAME	W_STATUS_CLASS	W_STATUS_CODE	W_STATUS_NAME
Product	Product	LOY_REDEMPTION_TYPE_CD	Product	Product
Loan Repayment	Loan Repayment	LOY_REDEMPTION_TYPE_CD	Loan Repayment	Loan Repayment
Expired	Expired	LOY_REDEMPTION_TYPE_CD	Expired	Expired

13.2.5 How to Configure the domainValues_Loyalty_Tier_movement_Type.csv

This section explains how to configure the domainValues_Loyalty_Tier_movement_Type.csv file.

> **Note:** This table is used after staging and therefore cannot be configured until after staging is complete.

1. Identify the Loyalty Tier Movement Type in your Oracle source system by using the following SQL:

```
SELECT
W_CODE_D.SOURCE_CODE,
W_CODE_D.SOURCE_CODE_1,
W_CODE_D.SOURCE_NAME_1,
W_CODE_D.CATEGORY,
W_CODE_D.LANGUAGE_CODE,
W_CODE_D.MASTER_CODE,
W_CODE_D.MASTER_VALUE
FROM
W_CODE_D
WHERE
CATEGORY = 'LOY_MEM_TIER_APPR_STATUS_CD'
```

2. Using a text editor, open the domainValues_Loyalty_Tier_movement_Type.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the values in the SOURCE_CODE column to the TIER_MOVE_CODE and W_TIER_MOVE_CODE columns in the CSV file.
4. Copy the values in the SOURCE_NAME_1 column to the TIER_MOVE_NAME and W_TIER_MOVE_NAME columns in the CSV file.
5. Save and close the file.

13.2.6 How to Configure the file_Redeemed_Unit_Cost.csv

The file_Redeemed_Unit_Cost.csv flat file is used to retrieve a unit rate associated with a program, product, and partner. The unit rate is used to calculate the amount of redemption points. This calculation is in accordance to the International Financial Reporting Interpretations Committee (IFRIC) guidelines on how to recognize revenue in Loyalty.

To configure the file_Redeemed_Unit_Cost.csv:

1. Using a text editor, open the file_Redeemed_unit_cost.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
2. Enter the appropriate data for the following columns:

Column Name	Description
PARTNER_ID	Name of the partner of the product.
PRODUCT_ID	Name of the product being redeemed.
PROGRAM_ID	Name of the program the product is associated with.
CURCY_CD	Currency code of the points.
COST_PER_POINT	The cost per point for redeemed points.

3. Save and close the file.

13.3 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle Loyalty Analytics. It contains the following topic:

- Section 13.3.1, "How to Configure the LoyaltyTotalRevenueInput.xls"

13.3.1 How to Configure the LoyaltyTotalRevenueInput.xls

The LoyaltyTotalRevenueInput.xls file is used to retrieve data about the total revenue, which is used in comparison metrics under the revenue and member analysis areas. This flat file contains the following columns:

- Total Revenue
- Document Currency Code
- Exchange Rate
- Simple Time Dimension attributes

> **Note:** This flat file is used to retrieve the Total Revenue amount; the only dimension analysis that can be performed is on date and time.

You import the data from this flat file into the RPD Physical layer using the Excel spreadsheet option. You need to create an ODBC data source named "Loyalty Input Data Source" and point it to the flat file for the data import.

For information about how to configure an ODBC data source name (DSN), see *Oracle Fusion Middleware Integrator's Guide for Oracle Business Intelligence Enterprise Edition*.

You must also enter this table under Loyalty Input Connection Pool in the Physical layer of the RPD.

To configure the LoyaltyTotalRevenueInput.xls:

1. Using a text editor, open the LoyaltyTotalRevenueInput.xls located in the *MW_HOME*\biapps\dwrep\Loyalty Analytics folder.
2. Enter the appropriate data for the following columns:

Column Name	Description
Total Amt	Total revenue amount for the year, quarter, and month.
DOC_Curr_Code	The currency code.
Global1_Exchange_Rate	The exchange rate to be used to exchange amount from currency to company currency.
Date Key	DATE_KEY in the spreadsheet should be the same ROW_WID value from W_DAY_D, for example 20081231, which stands for December 31, 2008.
Year	The applicable year.
Qtr	The applicable quarter.
Month	The applicable month.

3. Save and close the file.

13.4 Configuring SQL Transformations in Informatica for Database Dependencies

This section describes how to configure SQL transformations for the "PLP_LoyMemberTierMovementQtrAggr" task in Informatica to work with Microsoft SQL Server, DB2, and Teradata databases.

To configure SQL transformations in Informatica for database dependencies:

1. In Informatica PowerCenter Designer, connect to the repository.
2. Open the PLP folder.
3. Check out the mapping PLP_LoyMemberTierMovementQtrAggr.

4. Perform the following steps for your database version:

   - For data warehouses on Microsoft SQL Server databases:

     – Open the SQL_LoyMemberTierMove_Agg SQL transformation and display the SQL Settings tab. Then, change the Database Type to Microsoft SQL Server.

     – Display the SQL Ports tab. Then, change the Native Type datatype for ALL columns that read 'bit' and change the value to 'varchar'.

   - For data warehouses on DB2 databases:

     – Open the SQL_LoyMemberTierMove_Agg SQL transformation and display the SQL Settings tab. Then, change the Database Type to DB2.

     – Display the SQL Ports tab. Then, change the Native Type datatype for ALL columns that read 'char' and change the value to 'varchar'.

   - For data warehouses on Teradata databases:

     – Open the SQL_LoyMemberTierMove_Agg SQL transformation and display the SQL Settings tab. Change the Database Type to TeraData.

     – Display the SQL Ports tab. Then, change the Native Type datatype for ALL columns that read 'char' and change the value to 'varchar'.

5. Check in the mapping PLP_LoyMemberTierMovementQtrAggr.

14

Configuring Oracle Pharma Analytics

This section describes how to configure Oracle Pharma Analytics. It contains the following topics:

14.1 Overview of Oracle Business Analytics Warehouse for Life Sciences

This section describes the key features and architecture of the Oracle Business Analytics Warehouse for Life Sciences Syndicated and Call Activity data.

- ETL for direct load of syndicated data.

  Syndicated data can be directly loaded into the Oracle Business Analytics Warehouse.

  A set of prebuilt processes against common external syndicated data types allows direct load of syndicated data, such as direct sales data, indirect sales data, prescription data, plan level Rx data, physician plan level Rx data, and weekly early view data.

- Syndicated data staging architecture.

  - Syndicated data is loaded against a staging architecture. The staging table architecture uses a denormalized time dimension that improves data loading by a factor of how many periods are loaded.
  - A set of cross-reference files for external key matching are supplied to resolve cross referencing and loading external syndicated data where the accounts, products, territories and contacts IDs are different from the Siebel IDs.

- Syndicated data and Call Activity data single grain fact tables.

  - Single grain syndicated data star schemas for power users and operational users.
  - Operational syndicated data star schemas for operational reporting.
  - Single grain call activity fact tables - account calls, account-attendee calls, and contact calls.
- Syndicated data Moving Annual Total (MAT) metrics calculated on ETL.
  - Calculation of rolling period calculations takes advantage of the syndicated data staging architecture to populate common pharmaceutical time aggregate metrics within the Oracle Business Analytics Warehouse.
- ETL for syndicated data market aggregation.
  - ETL process to load the Syndicated Data market fact tables based on aggregations on the product-market hierarchy. Minimizes the requirement to purchase pre-aggregated data by market.
- Set of prebuilt aggregation tables for high performance reporting.
  - Sales level aggregation tables for product-based syndicated data facts.
  - Sales level aggregation tables for market-based syndicated data facts.
  - Call activity aggregation tables for reports requiring top-level sales level activities.
  - De-duplication tables to de-duplicate data when the same contact is assigned to multiple territories for the district level for Physician Plan Level Rx syndicated data and market, Indirect Sales syndicated data, and market only.
- Mini dimensions.
  - Account and contact rank mini dimensions.
  - Representative specialty LOV dimensions.
  - W_POSTN_CON_D used only for contact primary address.

14.2 Importing Syndicated Data into Oracle Business Analytics Warehouse

Pharmaceutical companies purchase weekly and monthly sales and prescription data, known as syndicated data, from third-party vendors such as IMS, NDC, and Cegedim. Syndicated data vendors acquire data from drug wholesalers and retailers on a daily, weekly, and monthly basis and compile a master file of customers (wholesalers, pharmacies, hospitals, and doctors) and sales or prescription transactions for customers. Measures include indirect sales, indirect units, and prescriptions, and differ by vendor and periodicity. It is used for sales force analysis reporting and customer targeting.

The data is derived from panels of physicians, pharmacies, and so on, and projected nationally. Since the panels may change on a monthly basis, syndicated data suppliers tend to change their projections of sources on a monthly basis leading to full restatements of historical data. Thus, pharmaceutical companies are required to refresh fully the data in their data warehouses. In addition, weekly data requires incremental loading.

After it is delivered by the vendor, the syndicated data must be fully reloaded into the Oracle Business Analytics Warehouse in a timely manner and made available to users in order for them to make use of sales force analysis reporting.

This section includes the following topics:

- Section 14.2.1, "Syndicated Loading Definitions"
- Section 14.2.2, "Data Types Supported in the Oracle Business Analytics Warehouse"
- Section 14.2.3, "Loading Syndicated Data into the Oracle Business Analytics Warehouse"

14.2.1 Syndicated Loading Definitions

Table 14–1 provides terms and definitions related to syndicated loading.

Table 14–1 Syndicated Loading Definitions

Term	Definition
Syndicated data	Third-party data from vendors that shows sales and prescription results for client defined markets.
Brick	Micro sales geographic area defined by vendor that contains prescription and sales trends for clients' product groupings or markets. Bricks do not exist in the US where the micro sales geographic area is commonly the postal code or zip code.
NRx	Abbreviation of new prescriptions. A new prescription is defined as dispensed prescriptions given a new number by the pharmacy, not necessarily new therapy for the patient.
TRx	Abbreviation of total prescriptions. TRx = NRx + Refills. After the first time a prescription is filled, when it is refilled, the pharmacy refers back to the previous drug ID number and notes this as a refill.
Indirect sales	Total drug wholesalers product sales values to drug retailers (pharmacies) by brick or zip code. Sales values are calculated as units multiplied by client-selected price, where the price is the warehouse acquisition cost unless the client has chosen to have a different price applied to its direct sales or government depot sales.
Indirect units	Total drug wholesalers product sales units to drug retailers (pharmacies) by brick or zip code. Sales unit quantity is determined by multiplying package factor to obtain unit value (for example, one carton x 24 bottles = 24 units).

14.2.2 Data Types Supported in the Oracle Business Analytics Warehouse

The Oracle Business Analytics Warehouse supports multiple data types defined by data source type and periodicity, as described in Table 14–2.

Table 14–2 Data Types Supported in Oracle Business Analytics Warehouse

Data Type	Description
Prescription data by contact	Monthly NRx and TRx data for client defined markets that include competitor products data by physician.
Indirect sales brick level	Monthly wholesaler sales and unit values data for client defined markets that include competitor products aggregated by brick.
Indirect sales account level	Monthly wholesaler sales and unit values for company products by pharmacy.
Indirect sales zip level	Monthly wholesaler sales and unit values for client defined markets that include competitor products aggregated by postal code.

Table 14–2 (Cont.) Data Types Supported in Oracle Business Analytics Warehouse

Data Type	Description
Direct sales account level	Monthly direct factory sales data and unit values to wholesalers for company's products by wholesaler.
Direct sales brick level	Monthly direct factory sales data and unit values to wholesalers for company's products by brick.
Direct sales zip level	Monthly direct factory sales data and unit values to wholesalers for company's products by postal code.
Weekly prescription data	Weekly NRx and TRx data for client defined markets that include competitor products by physician.
Plan level prescription data	Monthly prescription data by managed care plan for client defined markets that includes competitor products.
Sales market	**Incentives.** Monthly incentive compensation data sourced from internal incentive compensation systems and loaded as a new data type in the Oracle Business Analytics Warehouse.
Sales market	**Modified.** Monthly incentive compensation data sourced from internal incentive compensation systems and loaded as a new data type in the Oracle Business Analytics Warehouse.
Physician plan level Rx data	Monthly prescription data for physicians associated with a managed care plan.
Prescription data by zip code	Monthly prescription data for client defined markets that includes competitor products aggregated by postal code.
Prescription data by brick	Monthly prescription data for client defined markets that include competitor products aggregated by brick.

14.2.3 Loading Syndicated Data into the Oracle Business Analytics Warehouse

The following options are available for loading syndicated data into the Oracle Business Analytics Warehouse:

- Section 14.2.3.1, "Loading From a Flat File Source Using Syndicated Data Staging Tables and Cross-Referencing Tables"
- Section 14.2.3.2, "Loading From a Flat File Source Using Syndicated Market Staging Tables"

14.2.3.1 Loading From a Flat File Source Using Syndicated Data Staging Tables and Cross-Referencing Tables

This option is supported in Oracle Business Intelligence Applications Version 7.8.3. To load syndicated data with this option, you have to prepare external data files for syndicated data and cross-reference data as described in tables, Table 14–4 through Table 14–9 and Table 14–15 through Table 14–19 in Section 14.3, "Syndicated Data Flat File Formats."

The ETL process will load syndicated data source files as a full load each time, but cross-reference data files will be loaded incrementally. Cross-reference data files must contain only new and updated information each time to support proper incremental loading. When the cross-reference data files are prepared, the data must be already loaded into the transactional database so the proper Siebel row IDs and the corresponding external source keys are resolved and provided in the data files.

Setting the correct alignment type to be used with the syndicated data is critical to loading data through the flat files. The Alignment Type is set in the external file AlignmentType.csv which is located in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles). Before running the ETL, the Alignment Type must be set to one of the following options under ALIGN_TYPE field.

- Zipcode
- Account/Contact
- Brick

In previous releases, only one Alignment Type could be set for use with all Syndicated Data Types. In Release 7.9, multiple Alignment Types can be set for different Syndicated Data Types.

For example, Physician Rx data can be loaded with an Account/Contact Alignment Type while Brick level Indirect Sales data can be loaded with a Brick Alignment Type. In addition, the same target Syndicated data table (for example, the Physician Rx table) can be loaded with data defined in multiple Alignment Types such as both Account/Contact and Zipcode.

The Syndicated Data Type and Alignment Type rule mappings are set in the AlignmentType.csv file and this data is loaded into the W_ALIGN_TYPE_G table at the start of the ETL process for use in the Syndicated data load. The format of the AlignmentType.csv file is shown in Table 14–3.

> **Note:** Before running the ETL, the values in the 'ALIGN_TYPE' column in this file should be updated to reflect the correct Alignment Types to be used with the appropriate Syndicated Data Types listed in the 'DATA_SRC_CD' column.

Table 14–3 AlignmentType.csv Mapping File

ROW_WID	DATA_SRC_CD	ALIGN_TYPE
1	RXBrk	Brick
2	RXEVM	Zipcode
3	RXPT	Account/Contact
4	RXPrf	Account/Contact
5	RXSMI	Zipcode
6	RXSMM	Zipcode
7	RXXPT	Account/Contact
8	RXZip	Zipcode
9	SlsDirAct	Account/Contact
10	SlsIndAct	Account/Contact
11	SlsIndBrk	Brick
12	SlsIndZip	Zipcode

You can change the Alignment Types only when you choose to run the full ETL load process.

Once the syndicated data and cross-reference source files are prepared, then the data can be loaded directly into Oracle Business Analytics Warehouse staging tables. If any invalid data source type is used in the syndicated data files, then the ETL will load as it is. You have to prepare the correct data source type in the data files. Only the data that should not be nullable but is null in data files will be filtered out separately so you can review the rejected data for the next ETL run.

When the syndicated data is loaded from the Oracle Business Analytics Warehouse staging tables to target base tables, such as W_SYNDD_RX_F, the external source keys are replaced with Oracle Business Intelligence keys and the data is loaded in the normalized format.

Then, based on what you defined as a primary alignment type above, the syndicated data in base target tables is replicated into the proper owner position and loaded into the main target tables at the territory level, such as W_SYND_RX_T_F, which contains all base table attributes plus ranking and position information.

14.2.3.1.1 Flat File Data Population Rules for Populating TRx Data This section provides information about various major columns in the W_SYNDD_RX_F table related to populating TRx data.

- INTEGRATION_ID
  - Data type is varchar(30).
  - Value can be any unique number or combination of values coming from the external file source with the appended postfix -##, such as -1, if the data is corresponding to the first month of bulk syndicated data, and -2 if the data is corresponding to the second month of bulk data, and so on.
  - INTEGRATION_ID manipulation is done by LS_EXP_FLATFILE_GENERATE_INTEGRATION_ID transformation. It is required because the incoming INTEGRATION_ID in flat file represents 26 different bulk loads in denormalized format. Therefore, the format must be changed to normalized for the Oracle Business Analytics Warehouse target table.
- DATASOURCE_NUM_ID
  - ETL run specific (a numeric value that indicates the source for the data).
  - Used in standard Siebel mappings to ensure each record carries a value indicating its source.
  - Used in source qualifiers to ensure that lookups on dimension tables when populating fact tables area referencing data that was loaded from a uniform data source.
  - Data source number 1 is reserved for the Siebel transactional database. The external source should use a number other than 1.
  - For syndicated data flat file loads, the external source number (other than 1) is loaded into staging tables. When the data is loaded from the staging tables to target tables and external source keys are matched to Siebel IDs, the data source number is converted as 1 in the target table.
  - Combination of INTEGRATION_ID and DATASOURCE_NUM_ID must be a unique value (mandated by a unique index on both columns).

    This means that the system will not accept the same record from the same source with identical unique IDs.
- DATA_LEVEL_WID
  - Stores a foreign key in the W_LOV_UD table for the record corresponding to that syndicated data source type for the data.

  - Value comes from DATA_SRC_CD in the flat file. For example, in the table W_SYNDD_RX_F, the proper data source type defined in flat file must be one of the following:
    - RXPrf
    - RXZip
    - RXBrk
    - RXSMI
    - RXSMM
  - The standard mapping uses the incoming DATA_SRC_CD code to do a lookup against the W_LOV_UD table to secure the ROW_WID of that data source type and inserts it into the W_SYNDD_RX_F table DATA_LEVEL_WID column.
- PAYER_TYPE_WID
  - Stores a foreign key in the W_INS_PLAN_D table for the record corresponding to that payer type of the data such as Cash, Total.
  - Value comes from PAYER_TYPE_ID in the flat file.
- PERIOD_TYPE_WID
  - Stores a foreign key in the W_LOV_D table for the record corresponding to the period for the data.
  - Value comes from PERIOD_CD in the flat file, such as Month, Week.
  - The standard mapping uses the incoming PERIOD_CD code to do a lookup against the W_LOV_D table to secure the ROW_WID of that period type and inserts it into the W_SYNDD_RX_F table PERIOD_TYPE_WID column
- GEO_WID
  - GEO_WID is a foreign key to the W_GEO_D dimension, which contains a geographical hierarchy based on zip code as the lowest level of detail. It is preconfigured to allow users to roll up a TRx measure to levels such as city, state or province, county, country, and continent.
  - If the flat file source of TRx data load is zip code level, then the ZIPCODE field in the flat file must be provided. Then, the value is performed using the following logic to get loaded in the target GEO_WID:
    - MPLT_GEO_WID_ZIPCODE takes zip code as the incoming value, and the literal NO_DUP_FLG = 'N' ports to do a lookup against W_GEO_D for any record where the ZIPCODE column matches the incoming zip code value and the DUP_ZIPCODE column = 'N'. (Sample data in the W_GEO_D table contains multiple city records per zip code, only one of which is marked as DUP_ZIPCODE = 'N'). The mapping also ensures that even with multiple matches only the first record is retrieved, and with no matches, the data comes back with the appropriate Unspecified code.
- CON_GEO_WID
  - CON_GEO_WID is a foreign key to the W_GEO_D dimension, which contains a geographical hierarchy based on zip code as the lowest level of detail.
  - The following three attributes, city, country, and zip code, are the main keys to determine CON_GEO_WID, and are brought from flat file CON_CITY, CON_COUNTRY, and CON_ZIPCODE fields.

  - If the flat file source of TRx data load is contact level, then CON_CITY, CON_COUNTRY, and CON_ZIPCODE fields in flat file must be provided. Then, the value is performed the following logic to get loaded in target CON_GEO_WID:
    - LS_EXP_FLATFILE_CITY_COUNTRY_ZIP_CHKNULL and LS_MPLT_FLATFILE_GEO_WID use the combination of city, country, and zip code to retrieve the first available record from W_GEO_D even if duplicates exist and regardless of the NO_DUP_FLG.

- ALIGN_WID
  - Depending on your selected alignment item type, the following transformation is performed:
    - When alignment item type is Zipcode:

```
IIF(DATA_SRC_CD = 'RXPrf', IN_CON_GEO_WID,
   IIF(DATA_SRC_CD = 'RXBrk' OR DATA_SRC_CD = 'RXSMI' OR DATA_SRC_CD =
'RXSMM' OR DATA_SRC_CD = 'RXZip', IN_GEO_WID,
      ETL_UNSPEC_NUM))
```

    - When alignment item type is Account/Contact:

```
IIF(DATA_SRC_CD = 'RXBrk' OR DATA_SRC_CD = 'RXPrf' OR DATA_SRC_CD =
'RXSMI' OR DATA_SRC_CD = 'RXSMM' OR DATA_SRC_CD = 'RXZip', IN_CONTACT_
WID, ETL_UNSPEC_NUM)
```

    - When alignment item type is Brick:

```
IIF(DATA_SRC_CD = 'RXPrf', IN_CON_AREA_WID,
   IIF(DATA_SRC_CD = 'RXBrk' OR DATA_SRC_CD = 'RXSMI' OR DATA_SRC_CD =
'RXSMM' OR DATA_SRC_CD = 'RXZip', IN_AREA_WID,
      ETL_UNSPEC_NUM))
```

    The resulting value is used to populate the ALIGN_WID column in the fact table, which should also be found in W_ALIGNMT_DH table ALIGN_WID column.

    Based on this ALIGN_WID column, Oracle Business Analytics Warehouse replicates proper territory and populates W_SYND_RX_T_F table as well.

Note: This section discusses the W_SYNDD_RX_F table as an example only. In the Oracle Business Analytics Warehouse there are six tables used for the different syndicated data sources which are loaded in the same manner: W_SYNDD_DS_F, W_SYNDD_IDS_F, W_SYNDD_RX_F, W_SYNDD_PT_F, W_SYNDD_XPT_F, and W_SYNDD_W_F tables.

14.2.3.2 Loading From a Flat File Source Using Syndicated Market Staging Tables

The Oracle Business Analytics Warehouse supports loading syndicated market data using flat files. You have to prepare flat files of syndicated market data source and cross-referencing data source files as described in tables, Table 14–10 through Table 14–14 and Table 14–15 through Table 14–19 in Section 14.3, "Syndicated Data Flat File Formats."

By default this option is disabled. To use this feature, you must turn on the option manually in DAC. Turning on this option allows the flat file syndicated market data to be loaded directly into the Oracle Business Analytics Warehouse staging table W_SYNDM_RX_FS, and then loaded into the target table W_SYNDM_RX_F.

To load syndicated market data using flat files:

1. In DAC, go to the Design view, and display the Tasks tab.
2. Enter the following query:

   ```
   LS Load into SyndicatedMarket Fact for*
   ```

   Eleven tasks are returned.
3. Deactivate the following tasks, which, by default, are turned on:
   - LS Load into SyndicatedMarket Fact for Calculated Exponent Plantrak
   - LS Load into SyndicatedMarket Fact for Calculated Indirect Sales
   - LS Load into SyndicatedMarket Fact for Calculated Plantrak
   - LS Load into SyndicatedMarket Fact for Calculated Rx
   - LS Load into SyndicatedMarket Fact for Calculated Weekly Early View

   Note: Do not deactivate LS Load into SyndicatedMarket Fact for Calculated Direct Sales.

4. Activate the following tasks:
   - LS Load into SyndicatedMarket Fact for Exponent Plantrak
   - LS Load into SyndicatedMarket Fact for Indirect Sales
   - LS Load into SyndicatedMarket Fact for Plantrak
   - LS Load into SyndicatedMarket Fact for Rx
   - LS Load into SyndicatedMarket Fact for Weekly Early View
5. Save your work.

14.3 Syndicated Data Flat File Formats

Prepare the flat files based on the formats provided in this section and copy the formatted flat files to the SrcFiles directory on the Informatica server computer (for example, *INFA_HOME*\server\infa_shared\SrcFiles) before starting the ETL run.

This section includes descriptions for the following flat files used in the syndicated data load process:

- Section 14.3.1, "Syndicated Data Flat File for Direct Sales Data"
- Section 14.3.2, "Syndicated Data Flat File for Indirect Sales Data"
- Section 14.3.3, "Syndicated Data Flat File for Prescription Data"
- Section 14.3.4, "Syndicated Data Flat File for Plan Level Rx Data"
- Section 14.3.5, "Syndicated Data Flat File for Physician Plan Level Rx Data"
- Section 14.3.6, "Syndicated Data Flat File for Weekly Early View Data"
- Section 14.3.7, "Syndicated Market Flat File for Indirect Sales Data"
- Section 14.3.8, "Syndicated Market Flat File for Prescription Data"

- Section 14.3.9, "Syndicated Market Flat File for Plan Level Rx Data"
- Section 14.3.10, "Syndicated Market Flat File for Physician Plan Level Rx Data"
- Section 14.3.11, "Syndicated Market Flat File for Weekly Early View Data"
- Section 14.3.12, "Flat File Cross-Reference Data Population Rules"

14.3.1 Syndicated Data Flat File for Direct Sales Data

Table 14–4 provides information on the syndicated data flat file for the Direct Sales data. Save the flat file as LoadSyndData_DS.txt, use the pipe (|) symbol as the field separator.

Table 14–4 Syndicated Data Flat File - Direct Sales Data

Field Name	Data Type	Field Required	Field Description
DATASOURCE_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	Enter the following Siebel data source type code: SlsDirAct
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
AMT_DT	VARCHAR2(15)	-	Exchange date for currency code. Date format YYYYMMDD.
ACCNT_ID	VARCHAR2(30)	-	External account ID. **Note:** Corresponding Siebel S_ORG_EXT.ROW_ID should be provided in cross-reference file.
ACCNT_AREA_ID	VARCHAR2(30)	-	External brick ID of account's primary address for account level. **Note:** Corresponding Siebel S_REGION_EXT.ROW_ID (S_CON_ADDR.BRICK_ID) should be provided in cross-reference file.
ACCNT_CITY	VARCHAR2(50)	-	Account city. Initial capital format; for example, Short Hills.
ACCNT_COUNTRY	VARCHAR2(30)	-	Account country. Initial capital format; for example, France; or abbreviate format, such as USA.
ACCNT_ZIPCODE	VARCHAR2(30)	-	Account zip code.
AMT_CURCY_CD	VARCHAR2(30)	-	Currency code for Amount.
PAYER_TYPE_ID	VARCHAR2(30)	-	External payer type ID. This column is used to define payment type, such as Cash, Government, Third Party, Total, and so on. **Note:** Corresponding Siebel S_INS_PLAN.ROW_ID should be provided in cross-reference file.
PERIOD_CD	VARCHAR2(30)	-	Enter the following Siebel period code: Month

Table 14–4 (Cont.) Syndicated Data Flat File - Direct Sales Data

Field Name	Data Type	Field Required	Field Description
PROD_ID	VARCHAR2(30)	-	External product ID. **Note:** Corresponding Siebel S_PROD_INT.ROW_ID should be provided in cross-reference file.
S_AMT_M01	NUMBER(22,7)	-	Stores sales amounts of current month data.
S_AMT_M02	NUMBER(22,7)	-	Stores sales amounts of previous month data.
S_AMT_M03 - S_AMT_M26	NUMBER(22,7)	-	-
S_UNIT_M01	NUMBER(22,7)	-	Stores sales units of current month data (for example, 20050301 data).
S_UNIT_M02	NUMBER(22,7)	-	Stores sales units of previous month data (for example, 20050201 data).
S_UNIT_M03	NUMBER(22,7)	-	Stores sales units of 20050101 data.
S_UNIT_M04 - S_UNIT_M26	NUMBER(22,7)	-	-
COUNTRY_REGION_NAME	VARCHAR2(30)	-	Country or Region name.[1]

[1] Use the field to provide the partition key when partition is enabled. Leave this field blank if partition is not used. see Section 14.4, "Supporting Database Partitioning in Syndicated Data and Market Load."

14.3.2 Syndicated Data Flat File for Indirect Sales Data

Table 14–5 provides information on the syndicated data flat file for the Indirect Sales data. Save the flat file as LoadSyndData_IDS.txt, use the pipe (|) symbol as the field separator.

Table 14–5 Syndicated Data Flat File - Indirect Sales Data

Field Name	Data Type	Field Required	Field Description
DATASOURCE_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	Enter the following Siebel data source type code: SlsIndAct (Indirect sales account level) SlsIndZip (Indirect sales zip code level) SlsIndBrk (Indirect sales brick level)
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
AMT_DT	VARCHAR2(15)	-	Exchange date for currency code. Date format YYYYMMDD.

Table 14–5 (Cont.) Syndicated Data Flat File - Indirect Sales Data

Field Name	Data Type	Field Required	Field Description
ACCNT_ID	VARCHAR2(30)	-	External account ID. **Note:** Corresponding Siebel S_ORG_EXT.ROW_ID should be provided in cross-reference file.
ACCNT_AREA_ID	VARCHAR2(30)	-	External brick ID of account's primary address for account level. **Note:** Corresponding Siebel S_REGION_EXT.ROW_ID (S_CON_ADDR.BRICK_ID) should be provided in cross-reference file.
ACCNT_CITY	VARCHAR2(50)	-	Account city. Enter value if DATA_SRC_CD is account level. Initial capital format; for example, Short Hills.
ACCNT_COUNTRY	VARCHAR2(30)	-	Account country. Enter value if DATA_SRC_CD is account level. Initial capital format; for example, France; or abbreviate format, such as USA.
ACCNT_ZIPCODE	VARCHAR2(30)	-	Account zip code. Enter value if DATA_SRC_CD is account level.
AMT_CURCY_CD	VARCHAR2(30)	-	Applies to syndicated data only. Currency code for Amount.
AREA_ID	VARCHAR2(30)	-	External syndicate brick ID. Enter value if DATA_SRC_CD is brick level. **Note:** Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
PAYER_TYPE_ID	VARCHAR2(30)	-	External payer type ID. This column is used to define payment type, such as Cash, Government, Third Party, Total, and so on. **Note:** Corresponding Siebel S_INS_PLAN.ROW_ID should be provided in cross-reference file.
PERIOD_CD	VARCHAR2(30)	-	Enter the following Siebel period code: Month
PROD_ID	VARCHAR2(30)	-	Applies to syndicated data only. External product ID. **Note:** Corresponding Siebel S_PROD_INT.ROW_ID should be provided in cross-reference file.
ZIPCODE	VARCHAR2(30)	-	Syndicate zip code. Enter value if DATA_SRC_CD is zip code level.
S_AMT_M01	NUMBER(22,7)	-	Stores sales amounts of current month data.
S_AMT_M02	NUMBER(22,7)	-	Stores sales amounts of previous month data.
S_AMT_M03	NUMBER(22,7)	-	Stores sales amounts of 20050101 data.
S_AMT_M04 - S_AMT_M26	NUMBER(22,7)	-	-

Table 14–5 (Cont.) Syndicated Data Flat File - Indirect Sales Data

Field Name	Data Type	Field Required	Field Description
S_UNIT_M01	NUMBER(22,7)	-	Stores sales units of current month data.
S_UNIT_M02	NUMBER(22,7)	-	Stores sales units of previous month data.
S_UNIT_M03	NUMBER(22,7)	-	Stores sales units of 20050101 data.
S_UNIT_M04 - S_UNIT_M26	NUMBER(22,7)	-	-
COUNTRY_REGION_NAME	VARCHAR2(30)	-	Country or Region name.[1]

[1] Use the field to provide the partition key when partition is enabled. Leave this field blank if partition is not used. see Section 14.4, "Supporting Database Partitioning in Syndicated Data and Market Load."

14.3.3 Syndicated Data Flat File for Prescription Data

Table 14–6 provides information on the syndicated data flat file for the Prescription data. Save the flat file as LoadSyndData_RX.txt, use the pipe (|) symbol as the field separator.

Table 14–6 Syndicated Data Flat File - Prescription Data

Field Name	Data Type	Field Required	Field Description
DATASOURCE_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	RXPrf (Prescription data by Contact) RXZip (Prescription data by Zip Code) RXBrk (Prescription data by Brick) RXSMI (Sales Market – Incentives) RXSMM (Sales Market – Modified)
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
AREA_ID	VARCHAR2(30)	-	External syndicate brick ID. Enter value if DATA_SRC_CD is brick level. **Note:** Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
CONTACT_ID	VARCHAR2(30)	-	External contact ID. **Note:** Corresponding Siebel S_CONTACT.ROW_ID should be provided in cross-reference file.
CON_AREA_ID	VARCHAR2(30)	-	External syndicate brick ID of contact's primary address. Enter value if DATA_SRC_CD is brick level. **Note:** Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.

Table 14–6 (Cont.) Syndicated Data Flat File - Prescription Data

Field Name	Data Type	Field Required	Field Description
CON_CITY	VARCHAR2(50)	-	Contact city. Enter value if DATA_SRC_CD is contact level. Initial capital format; for example, Short Hills
CON_COUNTRY	VARCHAR2(30)	-	Contact country. Enter value if DATA_SRC_CD is contact level. Initial capital format; for example, France; or abbreviate format, such as USA.
CON_ZIPCODE	VARCHAR2(30)	-	Contact zip code. Enter value if DATA_SRC_CD is contact level.
PAYER_TYPE_ID	VARCHAR2(30)	-	External payer type ID. This column is used to define payment type, such as Cash, Government, Third Party, Total, and so on. **Note:** Corresponding Siebel S_INS_PLAN.ROW_ID should be provided in cross-reference file.
PERIOD_CD	VARCHAR2(30)	-	Enter the following Siebel period code: Month
PROD_ID	VARCHAR2(30)	-	External product ID. **Note:** Corresponding Siebel S_PROD_INT.ROW_ID should be provided in cross-reference file.
ZIPCODE	VARCHAR2(30)	-	Syndicate zip code. Enter value if DATA_SRC_CD is zip code level.
CONV_FACTOR	NUMBER(22,7)	-	Applies to Rx data only. Used to convert Rx volume to measurable units which will be used to dollarize the Rx volume.
NRX_M01	NUMBER(22,7)	-	Stores New Rx of current month data.
NRX_M02	NUMBER(22,7)	-	Stores New Rx of previous month data.
NRX_M03	NUMBER(22,7)	-	Stores New Rx of 20050101 data.
NRX_M04 - NRX_M26	NUMBER(22,7)	-	-
TRX_M01	NUMBER(22,7)	-	Stores Total Rx of current month data.
TRX_M02	NUMBER(22,7)	-	Stores Total Rx of previous month data.
TRX_M03	NUMBER(22,7)	-	Stores Total Rx of 20050101 data.
TRX_M04 - TRX_M26	NUMBER(22,7)	-	-
COUNTRY_REGION_NAME	VARCHAR2(30)	-	Country or Region name.[1]

[1] Use the field to provide the partition key when partition is enabled. Leave this field blank if partition is not used. see Section 14.4, "Supporting Database Partitioning in Syndicated Data and Market Load."

14.3.4 Syndicated Data Flat File for Plan Level Rx Data

Table 14–7 provides information about the syndicated data flat file for the Plan Level Rx data. Save the flat file as LoadSyndData_PT.txt, use the pipe (|) symbol as the field separator.

Table 14–7 Syndicated Data Flat File - Plan Level Rx Data

Field Name	Data Type	Field Required	Field Description
DATASOURCE_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	Enter the following Siebel data source type code: RXPT
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
ACCNT_ID	VARCHAR2(30)	-	External account and plan ID. **Note:** Corresponding Siebel S_ORG_EXT.ROW_ID should be provided in cross-reference file.
ACCNT_AREA_ID	VARCHAR2(30)	-	External brick ID of account's primary address. Enter value if DATA_SRC_CD is account level. **Note:** Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
ACCNT_CITY	VARCHAR2(50)	-	Account city. Enter value if DATA_SRC_CD is account level. Initial capital format; for example, Short Hills
ACCNT_COUNTRY	VARCHAR2(30)	-	Account country. Enter value if DATA_SRC_CD is account level. Initial capital format; for example, France; or abbreviate format, such as USA.
ACCNT_ZIPCODE	VARCHAR2(30)	-	Account zip code. Enter value if DATA_SRC_CD is account level.
PERIOD_CD	VARCHAR2(30)	-	Enter the following Siebel period code: Month
PROD_ID	VARCHAR2(30)	-	Applies to syndicated data only. External product ID. **Note:** Corresponding Siebel S_PROD_INT.ROW_ID should be provided in cross-reference file.
CONV_FACTOR	NUMBER(22,7)	-	Applies to Rx data only. Used to convert Rx volume to measurable units which will be used to dollarize the Rx volume.
NRX_M01	NUMBER(22,7)	-	Stores New Rx of current month data.
NRX_M02	NUMBER(22,7)	-	Stores New Rx of previous month data.

Table 14–7 (Cont.) Syndicated Data Flat File - Plan Level Rx Data

Field Name	Data Type	Field Required	Field Description
NRX_M03	NUMBER(22,7)	-	Stores New Rx of 20050101 data.
NRX_M04 - NRX_M26	NUMBER(22,7)	-	-
TRX_M01	NUMBER(22,7)	-	Stores Total Rx of current month data.
TRX_M02	NUMBER(22,7)	-	Stores Total Rx of previous month data.
TRX_M03	NUMBER(22,7)	-	Stores Total Rx of 20050101 data.
TRX_M04 - TRX_M26	NUMBER(22,7)	-	-
COUNTRY_REGION_NAME	VARCHAR2(30)	-	Country or Region name.[1]

[1] Use the field to provide the partition key when partition is enabled. Leave this field blank if partition is not used. see Section 14.4, "Supporting Database Partitioning in Syndicated Data and Market Load."

14.3.5 Syndicated Data Flat File for Physician Plan Level Rx Data

Table 14–8 provides information on the syndicated data flat file for the Physician Plan Level Rx data. Save the flat file as LoadSyndData_XPT.txt, use the pipe (|) symbol as the field separator.

Table 14–8 Syndicated Data Flat File - Physician Plan Level Rx Data

Field Name	Data Type	Field Required	Field Description
DATASOURCE_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	Enter the following Siebel data source type code: RXXPT
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
ACCNT_ID	VARCHAR2(30)	-	External account ID. **Note:** Corresponding Siebel S_ORG_EXT.ROW_ID should be provided in cross-reference file.
ACCNT_AREA_ID	VARCHAR2(30)	-	External brick ID of account's primary address. Enter value if DATA_SRC_CD is brick level. **Note:** Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
ACCNT_CITY	VARCHAR2(50)	-	Account city. Enter a value if DATA_SRC_CD is account level. Initial capital format; for example, Short Hills

Table 14–8 (Cont.) Syndicated Data Flat File - Physician Plan Level Rx Data

Field Name	Data Type	Field Required	Field Description
ACCNT_ COUNTRY	VARCHAR2(30)	-	Account country. Enter value if DATA_ SRC_CD is account level. Initial capital format; for example, France; or abbreviate format, such as USA.
ACCNT_ ZIPCODE	VARCHAR2(30)	-	Account zip code. Enter value if DATA_ SRC_CD is account level.
CONTACT_ID	VARCHAR2(30)	-	External contact ID. **Note:** Corresponding Siebel S_ CONTACT.ROW_ID should be provided in cross-reference file.
CON_AREA_ID	VARCHAR2(30)	-	External brick ID of contact's primary address. Enter value if DATA_SRC_CD is contact level. **Note:** Corresponding Siebel S_ REGION.ROW_ID should be provided in cross-reference file.
CON_CITY	VARCHAR2(50)	-	Contact city. Enter value if DATA_SRC_ CD is contact level. Initial capital format; for example, Short Hills
CON_COUNTRY	VARCHAR2(30)	-	Contact country. Enter value if DATA_ SRC_CD is contact level. Initial capital format; for example, France; or abbreviate format, such as USA.
CON_ZIPCODE	VARCHAR2(30)	-	Contact zip code. Enter value if DATA_ SRC_CD is contact level.
PERIOD_CD	VARCHAR2(30)	-	Enter the following Siebel period code: Month
PROD_ID	VARCHAR2(30)	-	Applies to syndicated data only. External product ID. **Note:** Corresponding Siebel S_PROD_ INT.ROW_ID should be provided in cross-reference file.
CONV_FACTOR	NUMBER(22,7)	-	Applies to Rx data only. Used to convert Rx volume to measurable units which will be used to dollarize the Rx volume.
NRX_M01	NUMBER(22,7)	-	Stores New Rx of current month data.
NRX_M02	NUMBER(22,7)	-	Stores New Rx of previous month data.
NRX_M03	NUMBER(22,7)	-	Stores New Rx of 20050101 data.
NRX_M04 - NRX_M26	NUMBER(22,7)	-	-
TRX_M01	NUMBER(22,7)	-	Stores Total Rx of current month data.
TRX_M02	NUMBER(22,7)	-	Stores Total Rx s of previous month data.
TRX_M03	NUMBER(22,7)	-	Stores Total Rx of 20050101 data.
TRX_M04 - TRX_ M26	NUMBER(22,7)	-	-

Table 14–8 (Cont.) Syndicated Data Flat File - Physician Plan Level Rx Data

Field Name	Data Type	Field Required	Field Description
COUNTRY_ REGION_NAME	VARCHAR2(30)	-	Country or Region name.[1]

[1] Use the field to provide the partition key when partition is enabled. Leave this field blank if partition is not used. see Section 14.4, "Supporting Database Partitioning in Syndicated Data and Market Load."

14.3.6 Syndicated Data Flat File for Weekly Early View Data

Table 14–9 provides information on the syndicated data flat file for Weekly Early View data. Save the flat file as LoadSyndData_W.txt, use the pipe (|) symbol as the field separator.

Table 14–9 Syndicated Data Flat File - Weekly Early View Data

Field Name	Data Type	Field Required	Field Description
DATASOURCE_ NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	Enter the following Siebel data source type code: RXEVM
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
CONTACT_ID	VARCHAR2(30)	-	External contact ID. **Note:** Corresponding Siebel S_ CONTACT.ROW_ID should be provided in cross-reference file.
CON_AREA_ID	VARCHAR2(30)	-	External brick ID of contact's primary address. Enter value if DATA_SRC_CD is brick level. **Note:** Corresponding Siebel S_ REGION.ROW_ID should be provided in cross-reference file.
CON_CITY	VARCHAR2(50)	-	Contact city. Enter a value if DATA_ SRC_CD is contact level. Initial capital format; for example, Short Hills
CON_COUNTRY	VARCHAR2(30)	-	Contact country. Enter value if DATA_ SRC_CD is contact level. Initial capital format; for example, France; or abbreviate format, such as USA.
CON_ZIPCODE	VARCHAR2(30)	-	Contact zip code. Enter value if DATA_ SRC_CD is contact level.

Table 14–9 (Cont.) Syndicated Data Flat File - Weekly Early View Data

Field Name	Data Type	Field Required	Field Description
PAYER_TYPE_ID	VARCHAR2(30)	-	External payer type ID. This column is used to define payment type, such as Cash, Government, Third Party, Total, and so on. **Note:** Corresponding Siebel S_INS_PLAN.ROW_ID should be provided in cross-reference file.
PERIOD_CD	VARCHAR2(30)	-	Enter the following Siebel period code: Week
PROD_ID	VARCHAR2(30)	-	Applies to syndicated data only. External product ID. **Note:** Corresponding Siebel S_PROD_INT.ROW_ID should be provided in cross-reference file.
CONV_FACTOR	NUMBER(22,7)	-	Applies to Rx data only. Used to convert Rx volume to measurable units which will be used to dollarize the Rx volume.
NRX_W01	NUMBER(22,7)	-	Stores New Rx of current week data.
NRX_W02	NUMBER(22,7)	-	Stores New Rx of previous week data.
NRX_W03	NUMBER(22,7)	-	Stores New Rx of 20050101 data.
NRX_W04 - NRX_W13	NUMBER(22,7)	-	-
TRX_W01	NUMBER(22,7)	-	Stores Total Rx of current week data.
TRX_W02	NUMBER(22,7)	-	Stores Total Rx of previous week data.
TRX_W03	NUMBER(22,7)	-	Stores Total Rx of 20050101 data.
TRX_W04 - TRX_W13	NUMBER(22,7)	-	-
COUNTRY_REGION_NAME	VARCHAR2(30)	-	Country or Region name.[1]

[1] Use the field to provide the partition key when partition is enabled. Leave this field blank if partition is not used. see Section 14.4, "Supporting Database Partitioning in Syndicated Data and Market Load."

14.3.7 Syndicated Market Flat File for Indirect Sales Data

Table 14–10 provides information on the syndicated market flat file for Indirect Sales data. Save the flat file as LoadSyndMarket_IDS.txt, use the pipe (|) symbol as the field separator.

Table 14–10 Syndicated Market Flat File - Indirect Sales Data

Field Name	Data Type	Field Required	Field Description
DATASOURCE_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.

Table 14–10 (Cont.) Syndicated Market Flat File - Indirect Sales Data

Field Name	Data Type	Field Required	Field Description
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	Enter the following Siebel data source type code: SlsIndAct (Indirect sales account level) SlsIndZip (Indirect sales zip code level) SlsIndBrk (Indirect sales brick level)
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
MKT_AMT_DT	VARCHAR2(15)	-	Exchange date for currency code. Date format YYYYMMDD.
ACCNT_ID	VARCHAR2(30)	-	External account ID. **Note:** Corresponding Siebel S_ORG_EXT.ROW_ID should be provided in cross-reference file.
ACCNT_AREA_ID	VARCHAR2(30)	-	External brick ID of account's primary address for account level. **Note:** Corresponding Siebel S_REGION_EXT.ROW_ID (S_CON_ADDR.BRICK_ID) should be provided in cross-reference file.
ACCNT_CITY	VARCHAR2(50)	-	Account city. Enter value if DATA_SRC_CD is account level. Initial capital format; for example, Short Hills.
ACCNT_COUNTRY	VARCHAR2(30)	-	Account country. Enter value if DATA_SRC_CD is account level. Initial capital format; for example, France; or abbreviate format, such as USA.
ACCNT_ZIPCODE	VARCHAR2(30)	-	Account zip code. Enter value if DATA_SRC_CD is account level.
AREA_ID	VARCHAR2(30)	-	External syndicate brick ID. Enter value if DATA_SRC_CD is brick level. **Note:** Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
MARKET_ID	VARCHAR2(30)	-	External market ID for the product. **Note:** Corresponding Siebel S_PROD_INT.ROW_ID should be provided in cross-reference file.
MKT_AMT_CURCY_CD	VARCHAR2(30)	-	Currency code for Amount.

Table 14–10 (Cont.) Syndicated Market Flat File - Indirect Sales Data

Field Name	Data Type	Field Required	Field Description
PAYER_TYPE_ID	VARCHAR2(30)	-	External payer type ID. This column is used to define payment type, such as Cash, Government, Third Party, Total, and so on. **Note:** Corresponding Siebel S_INS_PLAN.ROW_ID should be provided in cross-reference file.
PERIOD_CD	VARCHAR2(30)	-	Enter the following Siebel period code: Month
ZIPCODE	VARCHAR2(30)	-	Syndicate zip code. Enter value if DATA_SRC_CD is zip code level.
MKT_S_AMT_M01	NUMBER(22,7)	-	Stores sales amounts of current month data.
MKT_S_AMT_M02	NUMBER(22,7)	-	Stores sales amounts of previous month data.
MKT_S_AMT_M03	NUMBER(22,7)	-	Stores sales amounts of 20050101 data.
MKT_S_AMT_M04 - MKT_S_AMT_M26	NUMBER(22,7)	-	-
MKT_S_UNIT_M01	NUMBER(22,7)	-	Stores sales units of current month data.
MKT_S_UNIT_M02	NUMBER(22,7)	-	Stores sales units of previous month data.
MKT_S_UNIT_M03	NUMBER(22,7)	-	Stores sales units of 20050101 data.
MKT_S_UNIT_M04 - MKT_S_UNIT_M26	NUMBER(22,7)	-	-
COUNTRY_REGION_NAME	VARCHAR2(30)	-	Country or Region name.[1]

[1] Use the field to provide the partition key when partition is enabled. Leave this field blank if partition is not used. see Section 14.4, "Supporting Database Partitioning in Syndicated Data and Market Load."

14.3.8 Syndicated Market Flat File for Prescription Data

Table 14–11 provides information on the syndicated market flat file for Prescription data. Save the flat file as LoadSyndMarket_RX.txt, use the pipe (|) symbol as the field separator.

Table 14–11 Syndicated Market Flat File - Prescription Data

Field Name	Data Type	Field Required	Field Description
DATASOURCE_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	RXPrf (Prescription data by Contact) RXZip (Prescription data by Zip Code) RXBrk (Prescription data by Brick) RXSMI (Sales Market – Incentives) RXSMM (Sales Market – Modified)

Table 14–11 (Cont.) Syndicated Market Flat File - Prescription Data

Field Name	Data Type	Field Required	Field Description
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
AREA_ID	VARCHAR2(30)	-	External syndicate brick ID. Enter value if DATA_SRC_CD is brick level. **Note:** Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
CONTACT_ID	VARCHAR2(30)	-	External contact ID. **Note:** Corresponding Siebel S_CONTACT.ROW_ID should be provided in cross-reference file.
CON_AREA_ID	VARCHAR2(30)	-	External syndicate brick ID of contact's primary address. Enter value if DATA_SRC_CD is brick level. **Note:** Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
CON_CITY	VARCHAR2(50)	-	Contact city. Enter value if DATA_SRC_CD is contact level. Initial capital format; for example, Short Hills
CON_COUNTRY	VARCHAR2(30)	-	Contact country. Enter value if DATA_SRC_CD is contact level. Initial capital format; for example, France; or abbreviate format, such as USA.
CON_ZIPCODE	VARCHAR2(30)	-	Contact zip code. Enter value if DATA_SRC_CD is contact level.
MARKET_ID	VARCHAR2(30)	-	External market ID for the product. **Note:** Corresponding Siebel S_PROD_INT.ROW_ID should be provided in cross-reference file.
PAYER_TYPE_ID	VARCHAR2(30)	-	External payer type ID. This column is used to define payment type, such as Cash, Government, Third Party, Total, and so on. **Note:** Corresponding Siebel S_INS_PLAN.ROW_ID should be provided in cross-reference file.
PERIOD_CD	VARCHAR2(30)	-	Enter the following Siebel period code: Month
ZIPCODE	VARCHAR2(30)	-	Syndicate zip code. Enter value if DATA_SRC_CD is zip code level.
CONV_FACTOR	NUMBER(22,7)	-	Applies to Rx data only. Used to convert Rx volume to measurable units which will be used to dollarize the Rx volume.

Table 14–11 (Cont.) Syndicated Market Flat File - Prescription Data

Field Name	Data Type	Field Required	Field Description
MKT_NRX_M01	NUMBER(22,7)	-	Stores New Rx of current month data.
MKT_NRX_M02	NUMBER(22,7)	-	Stores New Rx of previous month data.
MKT_NRX_M03	NUMBER(22,7)	-	Stores New Rx of 20050101 data.
MKT_NRX_M04 - MKT_NRX_M26	NUMBER(22,7)	-	-
MKT_TRX_M01	NUMBER(22,7)	-	Stores Total Rx of current month data.
MKT_TRX_M02	NUMBER(22,7)	-	Stores Total Rx of previous month data.
MKT_TRX_M03	NUMBER(22,7)	-	Stores Total Rx of 20050101 data.
MKT_TRX_M04 - MKT_TRX_M26	NUMBER(22,7)	-	-
COUNTRY_ REGION_NAME	VARCHAR2(30)	-	Country or Region name.[1]

[1] Use the field to provide the partition key when partition is enabled. Leave this field blank if partition is not used. see Section 14.4, "Supporting Database Partitioning in Syndicated Data and Market Load."

14.3.9 Syndicated Market Flat File for Plan Level Rx Data

Table 14–12 provides information on the syndicated market flat file for Plan Level Rx data. Save the flat file as LoadSyndMarket_PT.txt, use the pipe (|) symbol as the field separator.

Table 14–12 Syndicated Market Flat File - Plan Level Rx Data

Field Name	Data Type	Field Required	Field Description
DATASOURCE_ NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	Enter the following Siebel data source type code: RXPT
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
ACCNT_ID	VARCHAR2(30)	-	External account and plan ID. **Note:** Corresponding Siebel S_ORG_ EXT.ROW_ID should be provided in cross-reference file.
ACCNT_AREA_ID	VARCHAR2(30)	-	External brick ID of account's primary address. Enter value if DATA_SRC_CD is account level. **Note:** Corresponding Siebel S_ REGION.ROW_ID should be provided in cross-reference file.

Table 14–12 (Cont.) Syndicated Market Flat File - Plan Level Rx Data

Field Name	Data Type	Field Required	Field Description
ACCNT_CITY	VARCHAR2(50)	-	Account city. Enter value if DATA_SRC_CD is account level. Initial capital format; for example, Short Hills
ACCNT_COUNTRY	VARCHAR2(30)	-	Account country. Enter value if DATA_SRC_CD is account level. Initial capital format; for example, France; or abbreviate format, such as USA.
ACCNT_ZIPCODE	VARCHAR2(30)	-	Account zip code. Enter value if DATA_SRC_CD is account level.
MARKET_ID	VARCHAR2(30)	-	External market ID for the product. **Note:** Corresponding Siebel S_PROD_INT.ROW_ID should be provided in cross-reference file.
PERIOD_CD	VARCHAR2(30)	-	Enter the following Siebel period code: Month
CONV_FACTOR	NUMBER(22,7)	-	Applies to Rx data only. Used to convert Rx volume to measurable units which will be used to dollarize the Rx volume.
MKT_NRX_M01	NUMBER(22,7)	-	Stores New Rx of current month data (for example, 20050301 data).
MKT_NRX_M02	NUMBER(22,7)	-	Stores New Rx of previous month data (for example, 20050201 data).
MKT_NRX_M03	NUMBER(22,7)	-	Stores New Rxof 20050101 data.
MKT_NRX_M04 - MKT_NRX_M26	NUMBER(22,7)	-	-
MKT_TRX_M01	NUMBER(22,7)	-	Stores Total Rx of current month data (for example, 20050301 data).
MKT_TRX_M02	NUMBER(22,7)	-	Stores Total Rx of previous month data (for example, 20050301 data).
MKT_TRX_M03	NUMBER(22,7)	-	Stores Total Rx of 20050101 data.
MKT_TRX_M04 - MKT_TRX_M26	NUMBER(22,7)	-	-
COUNTRY_REGION_NAME	VARCHAR2(30)	-	Country or Region name.[1]

[1] Use the field to provide the partition key when partition is enabled. Leave this field blank if partition is not used. see Section 14.4, "Supporting Database Partitioning in Syndicated Data and Market Load."

14.3.10 Syndicated Market Flat File for Physician Plan Level Rx Data

Table 14–13 provides information about the syndicated market flat file for Physician Plan Level Rx data. Save the flat file as LoadSyndMarket_XPT.txt, use the pipe (|) symbol as the field separator.

Table 14–13 Syndicated Market Flat File - Physician Plan Level Rx Data

Field Name	Data Type	Field Required	Field Description
DATASOURCE_ NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	Enter the following Siebel data source type code: RXXPT
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
ACCNT_ID	VARCHAR2(30)	-	External account ID. **Note:** Corresponding Siebel S_ORG_ EXT.ROW_ID should be provided in cross-reference file.
ACCNT_AREA_ID	VARCHAR2(30)	-	External brick ID of account's primary address. Enter value if DATA_SRC_CD is brick level. **Note:** Corresponding Siebel S_ REGION.ROW_ID should be provided in cross-reference file.
ACCNT_CITY	VARCHAR2(50)	-	Account city. Enter a value if DATA_ SRC_CD is account level. Initial capital format; for example, Short Hills
ACCNT_ COUNTRY	VARCHAR2(30)	-	Account country. Enter value if DATA_ SRC_CD is account level. Initial capital format; for example, France; or abbreviate format, such as USA.
ACCNT_ZIPCODE	VARCHAR2(30)	-	Account zip code. Enter value if DATA_ SRC_CD is account level.
CONTACT_ID	VARCHAR2(30)	-	External contact ID. **Note:** Corresponding Siebel S_ CONTACT.ROW_ID should be provided in cross-reference file.
CON_AREA_ID	VARCHAR2(30)	-	External brick ID of contact's primary address. Enter value if DATA_SRC_CD is contact level. **Note:** Corresponding Siebel S_ REGION.ROW_ID should be provided in cross-reference file.
CON_CITY	VARCHAR2(50)	-	Contact city. Enter value if DATA_SRC_ CD is contact level. Initial capital format; for example, Short Hills
CON_COUNTRY	VARCHAR2(30)	-	Contact country. Enter value if DATA_ SRC_CD is contact level. Initial capital format; for example, France; or abbreviate format, such as USA.

Table 14–13 (Cont.) Syndicated Market Flat File - Physician Plan Level Rx Data

Field Name	Data Type	Field Required	Field Description
CON_ZIPCODE	VARCHAR2(30)	-	Contact zip code. Enter value if DATA_SRC_CD is contact level.
MARKET_ID	VARCHAR2(30)	-	External market ID for product. **Note:** Corresponding Siebel S_PROD_INT.ROW_ID should be provided in cross-reference file.
PERIOD_CD	VARCHAR2(30)	-	Enter the following Siebel period code: Month
CONV_FACTOR	NUMBER(22,7)	-	Applies to Rx data only. Used to convert Rx volume to measurable units which will be used to dollarize the Rx volume.
MKT_NRX_M01	NUMBER(22,7)	-	Stores New Rx of current month data (for example, 20050301 data).
MKT_NRX_M02	NUMBER(22,7)	-	Stores New Rx of previous month data (for example, 20050201 data).
MKT_NRX_M03	NUMBER(22,7)	-	Stores New Rx of 20050101 data (for example, 20050301 data).
MKT_NRX_M04 - MKT_NRX_M26	NUMBER(22,7)	-	-
MKT_TRX_M01	NUMBER(22,7)	-	Stores Total Rx of current month data.
MKT_TRX_M02	NUMBER(22,7)	-	Stores Total Rx of previous month data.
MKT_TRX_M03	NUMBER(22,7)	-	Stores Total Rx of 20050101 data.
MKT_TRX_M04 - MKT_TRX_M26	NUMBER(22,7)	-	-
COUNTRY_REGION_NAME	VARCHAR2(30)	-	Country or Region name.[1]

[1] Use the field to provide the partition key when partition is enabled. Leave this field blank if partition is not used. see Section 14.4, "Supporting Database Partitioning in Syndicated Data and Market Load."

14.3.11 Syndicated Market Flat File for Weekly Early View Data

Table 14–14 provides information on the syndicated market flat file for the Weekly Early View data. Save the flat file as LoadSyndMarket_W.txt, use the pipe (|) symbol as the field separator.

Table 14–14 Syndicated Market Flat File - Weekly Early View Data

Field Name	Data Type	Field Required	Field Description
DATASOURCE_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	Enter the following Siebel data source type code: RXEVM
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.

Table 14–14 (Cont.) Syndicated Market Flat File - Weekly Early View Data

Field Name	Data Type	Field Required	Field Description
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
CONTACT_ID	VARCHAR2(30)	-	External contact ID. **Note:** Corresponding Siebel S_CONTACT.ROW_ID should be provided in cross-reference file.
CON_AREA_ID	VARCHAR2(30)	-	External brick ID of contact's primary address. Enter value if DATA_SRC_CD is brick level. **Note:** Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
CON_CITY	VARCHAR2(50)	-	Contact city. Enter a value if DATA_SRC_CD is contact level. Initial capital format; for example, Short Hills
CON_COUNTRY	VARCHAR2(30)	-	Contact country. Enter value if DATA_SRC_CD is contact level. Initial capital format; for example, France; or abbreviate format, such as USA.
CON_ZIPCODE	VARCHAR2(30)	-	Contact zip code. Enter value if DATA_SRC_CD is contact level.
MARKET_ID	VARCHAR2(30)	-	External market ID for the product. **Note:** Corresponding Siebel S_PROD_INT.ROW_ID should be provided in the cross-reference file.
PAYER_TYPE_ID	VARCHAR2(30)	-	External payer type ID. This column is used to define payment type, such as Cash, Government, Third Party, Total, and so on. **Note:** Corresponding Siebel S_INS_PLAN.ROW_ID should be provided in cross-reference file.
PERIOD_CD	VARCHAR2(30)	-	Enter the following Siebel period code: Week
CONV_FACTOR	NUMBER(22,7)	-	Applies to Rx data only. Used to convert Rx volume to measurable units which will be used to dollarize the Rx volume.
MKT_NRX_W01	NUMBER(22,7)	-	Stores New Rx of current data.
MKT_NRX_W02	NUMBER(22,7)	-	Stores New Rx of previous data.
MKT_NRX_W03	NUMBER(22,7)	-	Stores New Rx of 20050101 data.
MKT_NRX_W04 - MKT_NRX_W13	NUMBER(22,7)	-	-
MKT_TRX_W01	NUMBER(22,7)	-	Stores Total Rx of current week data.
MKT_TRX_W02	NUMBER(22,7)	-	Stores Total Rx of previous week data.
MKT_TRX_W03	NUMBER(22,7)	-	Stores Total Rx of 20050214 data.

Table 14–14 (Cont.) Syndicated Market Flat File - Weekly Early View Data

Field Name	Data Type	Field Required	Field Description
MKT_TRX_W04 - MKT_TRX_W13	NUMBER(22,7)	-	-
COUNTRY_ REGION_NAME	VARCHAR2(30)	-	Country or Region name.[1]

[1] Use the field to provide the partition key when partition is enabled. Leave this field blank if partition is not used. see Section 14.4, "Supporting Database Partitioning in Syndicated Data and Market Load."

14.3.12 Flat File Cross-Reference Data Population Rules

The following tables provide information about flat file cross-reference data population rules. Each file should be prepared with the pipe (|) symbol as the field separator.

- Table 14–15, " Flat File Cross-Reference Data Population Rules for Area (Brick) Data, LoadXRef_Area.txt"
- Table 14–16, " Flat File Cross-Reference Data Population Rules for Account and Plan Data, LoadXRef_Account.txt"
- Table 14–17, " Flat File Cross-Reference Data Population Rules for Payer Type Data, LoadXRef_Payer.txt"
- Table 14–18, " Flat File Cross-Reference Data Population Rules for Contact Data, LoadXRef_Contact.txt"
- Table 14–19, " Flat File Cross-Reference Data Population Rules for Product and Market Data, LoadXRef_Product.txt"

Table 14–15 Flat File Cross-Reference Data Population Rules for Area (Brick) Data, LoadXRef_Area.txt

Field Name	Data Type	Field Required	Field Description
EXT_DATASRC_ NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
EXT_ INTEGRATION_ID	VARCHAR2(30)	NOT NULL	External area (brick) ID. **Note:** This same ID should be used in ACCNT_AREA_ID, AREA_ID, AND CON_AREA_ID syndicated data and syndicated market flat files.
EXT_AREA_ NAME	VARCHAR2(50)	-	Area name. **Note:** The same value should already be loaded in Siebel S_REGION.NAME.
EXT_PROVINCE	VARCHAR2(50)	-	Area province. **Note:** The same value should already be loaded in Siebel S_REGION.PROVINCE.
EXT_STATE	VARCHAR2(50)	-	Area state. **Note:** The same value should already be loaded in Siebel S_REGION.STATE. Use the format all capitals and abbreviated, such as NJ.

Table 14–15 (Cont.) Flat File Cross-Reference Data Population Rules for Area (Brick) Data, LoadXRef_Area.txt

Field Name	Data Type	Field Required	Field Description
EXT_COUNTRY	VARCHAR2(30)	-	Area country. **Note:** The same value should already be loaded in Siebel S_REGION.COUNTRY. Use the initial capital format; for example, France.
INTEGRATION_ID	VARCHAR2(30)	NOT NULL	Siebel area (brick) ID. **Note:** This value should come from Siebel S_REGION.ROW_ID.

Table 14–16 Flat File Cross-Reference Data Population Rules for Account and Plan Data, LoadXRef_Account.txt

Field Name	Data Type	Field Required	Field Description
EXT_DATASRC_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1. **Note:** The same value should be used in the DATASOURCE_NUM_ID field of the syndicated data and syndicated market flat files.
EXT_INTEGRATION_ID	VARCHAR2(30)	NOT NULL	External account and plan ID. **Note:** The same ID should be used in ACCNT_AREA_ID, AREA_ID, AND CON_AREA_ID syndicated data and syndicated market flat files.
EXT_ACCNT_NAME	VARCHAR2(100)	-	Account and plan name. **Note:** The same value should already be loaded in Siebel S_ORG_EXT.NAME.
EXT_ACCNT_NUM	VARCHAR2(30)	-	Account outlet number.
INTEGRATION_ID	VARCHAR2(30)	NOT NULL	Siebel account ID. **Note:** This value should come from Siebel S_REGION.ROW_ID.

Table 14–17 Flat File Cross-Reference Data Population Rules for Payer Type Data, LoadXRef_Payer.txt

Field Name	Data Type	Field Required	Field Description
EXT_DATASRC_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1. **Note:** The same value should be used in the DATASOURCE_NUM_ID field of the syndicated data and syndicated market flat files.

Table 14–17 (Cont.) Flat File Cross-Reference Data Population Rules for Payer Type Data, LoadXRef_Payer.txt

Field Name	Data Type	Field Required	Field Description
EXT_INTEGRATION_ID	VARCHAR2(30)	NOT NULL	External payer ID. **Note:** The same value should be used in the PAYER_TYPE_ID field of the syndicated data and syndicated market flat files.
EXT_PLAN_TYPE	VARCHAR2(30)	-	Plan type, such as Payer, Plan, and so on. **Note:** The same value should already be loaded in Siebel S_INS_PLAN.PLAN_TYPE.
EXT_PAYER_NAME	VARCHAR2(100)	-	Payer type, such as Cash, Government, Third Party, Total, and so on. **Note:** The same value should already be loaded in Siebel S_INS_PLAN.NAME.
INTEGRATION_ID	VARCHAR2(30)	NOT NULL	Siebel payer ID. **Note:** This value should come from Siebel S_REGION.ROW_ID.

Table 14–18 Flat File Cross-Reference Data Population Rules for Contact Data, LoadXRef_Contact.txt

Field Name	Data Type	Field Required	Field Description
EXT_DATASRC_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1. **Note:** The same value should be used in the DATASOURCE_NUM_ID field of the syndicated data and syndicated market flat files.
EXT_INTEGRATION_ID	VARCHAR2(30)	NOT NULL	External contact ID. **Note:** The same value should be used in the CONTACT_ID field of the syndicated data and syndicated market flat files.
EXT_CON_FULL_NAME	VARCHAR2(102)	-	Contact name. **Note:** The same value should already be loaded in Siebel S_CONTACT.LAST_NAME and S_CONTACT.FST_NAME.
EXT_ME_NUM	VARCHAR2(50)	-	Contact ME number. **Note:** The same value should already be loaded in Siebel S_CONTACT.CSN.
INTEGRATION_ID	VARCHAR2(30)	NOT NULL	Siebel contact ID. **Note:** This value should come from Siebel S_CONTACT.ROW_ID.

Table 14–19 Flat File Cross-Reference Data Population Rules for Product and Market Data, LoadXRef_Product.txt

Field Name	Data Type	Field Required	Field Description
EXT_DATASRC_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1. **Note:** The same value should be used in the DATASOURCE_NUM_ID field of the syndicated data and syndicated market flat files.
EXT_INTEGRATION_ID	VARCHAR2(30)	NOT NULL	External product and market ID. **Note:** The same value should be used in the PAYER_TYPE_ID field of the syndicated data and syndicated market flat files.
EXT_PROD_TYPE	VARCHAR2(30)	-	Product and market type. **Note:** The same value should already be loaded in Siebel S_PROD_INT.
EXT_PROD_NAME	VARCHAR2(50)	-	Product and market name. **Note:** The same value should already be loaded in Siebel PROD_INT.NAME.
INTEGRATION_ID	VARCHAR2(30)	NOT NULL	Siebel product ID. **Note:** This value should come from Siebel S_PROD_INT.ROW_ID.

14.4 Supporting Database Partitioning in Syndicated Data and Market Load

This section contains the following topics:

- Section 14.4.1, "Overview"
- Section 14.4.2, "Definitions for Database Partition Components"
- Section 14.4.3, "Identify Partition Name and Value"
- Section 14.4.4, "How to Create Partition Table"
- Section 14.4.5, "How to Prepare Flat Files for Database Partition"
- Section 14.4.6, "How to Enable Database Partition Feature for Syndicated Data and Market Fact"
- Section 14.4.7, "How to Disable Database Partition Feature for Syndicated Data and Market Fact"
- Section 14.4.8, "Only the Oracle Platform is Supported for the Database Partition Feature in Default"

14.4.1 Overview

Oracle Business Analytics Warehouse for Life Sciences is designed to process six different types of syndicated data at high level. During syndicated data loading, ETL always does a full load on syndicated data and market that truncates the existing data and reloads with new data. This is because syndicated data loading does not support incremental load by nature. But there are user requirements to maintain syndicated data on multiple countries or regions and load only one country's or one region's data at a time without affecting the data of other countries or regions in the table. The difficulties faced in the previous releases when doing this type of data loading are addressed in this release.

In Release 7.9.6.3, Oracle Business Analytics Warehouse for Life Sciences supports syndicated data and market on multiple countries or regions by using the database partitioning feature. Without impacting other countries' data in the same target table, you have the flexibility to drop and reload syndicated data for one or more than one country or region within one ETL.

When preparing the syndicated data and market flat files, you must provide values in a new field for the partition key, COUNTRY_REGION_NAME, for the country's or region's name. Based on the values provided in COUNTRY_REGION_NAME, each country's or region's data is loaded into the specified partition respectively.

Note: Syndicated data partitioning is supported only on the Oracle platform. If you require partition on other database platforms, then customize the partition feature to suite your requirements.

14.4.2 Definitions for Database Partition Components

Table 14–20 provides terms and definitions related to Database Partition.

Table 14–20 Database Partition Definition for Syndicated Data and Market Load

Term	Definition
Partition Column [or, Partition Key]	The key column name that is used for partition in a table. The partition column, COUNTRY_REGION_NAME, is created for each table by default for the syndicated data and market load.
Partition Method	The way of defining partition. There are RANGE, LIST, HASH, and COMPOSITE partitioning methods. For syndicated data and market load, the LIST partitioning method is used.
Partition Name	The name to specify each partition for COUNTRY_REGION_NAME.[1]
Partition Value	The actual value that is used in each partition name and each flat file.[2]
Default Partition	Each table should have a DEFAULT partition value. Unmatched data found when comparing the flat file with the partition value is stored to the DEFAULT partition value.[3]

[1] Do not provide space in a partition name and all characters must be in upper case. The partition name must be equal to the partition value. For example, use partition name such as USA, UNITEDKINGDOM, or FRANCE.

[2] Do not provide space in partition values and all characters must be in upper case. The partition value must be equal to the partition name. Also, the same value must be used in the COUNTRY_REGION_NAME field in each flat file source. For example, use the partition name such as 'USA', 'UNITEDKINGDOM', or 'FRANCE'.

[3] Set the partition name as UNSPECIFIED. Set the partition value as DEFAULT for the data with partition key not present in the partition list.

14.4.3 Identify Partition Name and Value

To use the Database Partition feature, decide on the following, depending on the business requirement:

- the syndicated data type that requires a partition.
- how many partitions you require for the COUNTRY_REGION_NAME.
- the partition name that must be used.
- the partition value for each partition name that must be used.

In Release 7.9.6.3, there are several important restrictions that must be followed to select a partition name and partition value.

- One partition name can have only one partition value defined.
- The partition name must be equal to the partition value.
- There must be no space in the partition name and value, and all text characters must be in upper case.

For example, if you want to load the syndicated data and market for direct sales data every month, and the data is for three different countries—U.S.A, United Kingdom, and France, and you decide to use the Database Partition for the direct sales data, then use the following partition name and value.

Table 14–21 Example of Partition Name and Value for Syndicated Direct Sales Data

Syndicated Data for Direct Sales	Partition Name	Partition Value
U.S.A	USA	'USA'
United Kingdom	UNITEDKINGDOM [or, UK]	'UNITEDKINGDOM' [or, 'UK']
France	FRANCE	'FRANCE'

14.4.4 How to Create Partition Table

Based on the business requirement in pharmaceutical companies, the countries or regions to be partitioned in syndicated data and market tables can be very different. So, you must rebuild these related tables when the Partition feature is enabled.

The tables, Table 14–22 through Table 14–27 summarizes the fact tables in group for each syndicated data type. For example, to load syndicated data and market for direct sales data as partitioned, all the fact table group listed in Table 14–22 must be recreated as partitioned.

The following steps are recommended to recreate tables with partition:

1. Create a data warehouse schema in DAC.

   Note: For detailed information on how to create data warehouse tables using DAC, see *Oracle Business Intelligence Applications Installation Guide*.

   Decide on whether you want to use Database Partition for the entire syndicated data and market fact tables or selectively use a few syndicated data types.

   For example, use the following steps to partition only the direct sales data.

2. Use any utility such as Oracle SQL Developer to export the DDL script from existing schema in the data warehouse based on the list provided in Table 14–22 for the direct sales data fact tables group.

3. After the DDL script is exported, to recreate the partition table, modify the script based on the following syntax:

```
DROP TABLE [table_name];
CREATE TABLE [table_name] ( [column_1],[column_2],…[column_N] )
PARTITION BY LIST ( COUNTRY_REGION_NAME )
( PARTITION [partition_name_1] VALUES ( [partition_value_1] ), … PARTITION
[partition_name_N] VALUES ( [partition_value_N] ),PARTITION UNSPECIFIED VALUES
(DEFAULT));
```

> **Note:** A partition name can have only one partition value defined.
>
> The partition name must be equal to the partition value.
>
> There must be no space in the partition name and value, and the text must be in upper case.

For example, if you want to enable partition on the W_SYNDD_DS_F fact table for three countries—USA, United Kingdom, and France—then use the following SQL statement:

```
DROP TABLE W_SYNDD_DS_F;
CREATE TABLE W_SYNDD_DS_F
(
ACCNT_AREA_WID      NUMBER (10)  DEFAULT 0 NOT NULL,
ACCNT_GEO_WID       NUMBER (10)  DEFAULT 0 NOT NULL,
ACCNT_WID           NUMBER (10)  DEFAULT 0 NOT NULL,
ALIGN_WID           NUMBER (10)  DEFAULT 0 NOT NULL,
DATASOURCE_NUM_ID   NUMBER (10)  DEFAULT 0 NOT NULL,
DATA_LEVEL_WID      NUMBER (10)  DEFAULT 0 NOT NULL,
ETL_PROC_WID        NUMBER (10)  DEFAULT 0 NOT NULL,
EXT_INTEGRATION_ID  VARCHAR2 (30 CHAR)  DEFAULT '0' NOT NULL,
INTEGRATION_ID      VARCHAR2 (30 CHAR)  DEFAULT '0' NOT NULL,
MONTH_WID           NUMBER (10)  DEFAULT 0 NOT NULL,
PAYER_TYPE_WID      NUMBER (10)  DEFAULT 0 NOT NULL,
PERIOD_DAY_WID      NUMBER (10)  DEFAULT 0 NOT NULL,
PERIOD_TYPE_WID     NUMBER (10)  DEFAULT 0 NOT NULL,
PROD_WID            NUMBER (10)  DEFAULT 0 NOT NULL,
ROW_WID             NUMBER (10)  DEFAULT 0 NOT NULL,
MAT_S_AMT           NUMBER (22,7),
MAT_S_UNIT          NUMBER (22,7),
S_AMT               NUMBER (22,7),
S_UNIT              NUMBER (22,7),
U_AMT_DT            DATE,
U_SALES_AMT         NUMBER (22,7),
U_AMT_CURCY_CD      VARCHAR2 (30 CHAR),
ACCNT_ID            VARCHAR2 (30 CHAR),
PROD_ID             VARCHAR2 (30 CHAR),
PERIOD_DT           DATE,
X_CUSTOM            VARCHAR2 (10 CHAR),
COUNTRY_REGION_NAME VARCHAR2 (30 CHAR) DEFAULT 'UNSPECIFIED' NOT NULL
)
PARTITION BY LIST (COUNTRY_REGION_NAME)
(PARTITION USA VALUES ('USA'),
PARTITION UNITEDKINGDOM VALUES ('UNITEDKINGDOM'),
PARTITION FRANCE VALUES ('FRANCE'),
PARTITION UNSPECIFIED VALUES (DEFAULT)
);
```

4. Repeat step 3 for all fact tables for direct sales group in Table 14–22.

5. Execute the DDL script to recreate the partitioned tables.

Table 14–22 Syndicated Data and Market Fact Tables Group for Direct Sales Data

Type	Table Name
Syndicated Data Level	W_SYNDD_DS_F
	W_SYND_DS_T_F
	W_SD_DS_PAR1_A
	W_SD_DS_PAR2_A
	W_SD_DS_PAR3_A
	W_SD_DS_TOP_A
Syndicated Market Level	W_SYNDM_DS_F
	W_SYNM_DS_T_F
	W_SM_DS_PAR1_A
	W_SM_DS_PAR2_A
	W_SM_DS_PAR3_A
	W_SM_DS_TOP_A

Table 14–23 Syndicated Data and Market Fact Tables Group for Indirect Sales Data

Type	Table Name
Syndicated Data Level	W_SYNDD_IDS_F
	W_SYND_IDS_T_F
	W_SYND_IDS_D_F
	W_SD_IDS_PAR1_A
	W_SD_IDS_PAR2_A
	W_SD_IDS_PAR3_A
	W_SD_IDS_TOP_A
Syndicated Market Level	W_SYNDM_IDS_F
	W_SYNM_IDS_T_F
	W_SYNM_IDS_D_F
	W_SM_IDS_PAR1_A
	W_SM_IDS_PAR2_A
	W_SM_IDS_PAR3_A
	W_SM_IDS_TOP_A

Table 14–24 Syndicated Data and Market Fact Tables Group for Prescription Data

Type	Table Name
Syndicated Data Level	W_SYNDD_RX_F
	W_SYND_RX_T_F
	W_SD_RX_PAR1_A
	W_SD_RX_PAR2_A
	W_SD_RX_PAR3_A
	W_SD_RX_TOP_A

Table 14–24 (Cont.) Syndicated Data and Market Fact Tables Group for Prescription Data

Type	Table Name
Syndicated Market Level	W_SYNDM_RX_F
	W_SYNM_RX_T_F
	W_SM_RX_PAR1_A
	W_SM_RX_PAR2_A
	W_SM_RX_PAR3_A
	W_SM_RX_TOP_A

Table 14–25 Syndicated Data and Market Fact Tables Group for Physician Plan Level Rx Data

Type	Table Name
Syndicated Data Level	W_SYNDD_XPT_F
	W_SYND_XPT_D_F
	W_SYND_XPT_T_F
	W_SD_XPT_PAR1_A
	W_SD_XPT_PAR2_A
	W_SD_XPT_PAR3_A
	W_SD_XPT_TOP_A
Syndicated Market Level	W_SYNDM_XPT_F
	W_SYNM_XPT_D_F
	W_SYNM_XPT_T_F
	W_SM_XPT_PAR1_A
	W_SM_XPT_PAR2_A
	W_SM_XPT_PAR3_A
	W_SM_XPT_TOP_A

Table 14–26 Syndicated Data and Market Fact Tables Group for Plan Level Rx Data

Type	Table Name
Syndicated Data Level	W_SYNDD_PT_F
	W_SYND_PT_T_F
	W_SD_PT_PAR1_A
	W_SD_PT_PAR2_A
	W_SD_PT_PAR3_A
	W_SD_PT_TOP_A
Syndicated Market Level	W_SYNDM_PT_F
	W_SYNM_PT_T_F
	W_SM_PT_PAR1_A
	W_SM_PT_PAR2_A
	W_SM_PT_PAR3_A
	W_SM_PT_TOP_A

Table 14–27 Syndicated Data and Market Fact Tables Group for Weekly Early View Data

Type	Table Name
Syndicated Data Level	W_SYNDD_W_F
	W_SYND_W_T_F
	W_SD_W_PAR1_A
	W_SD_W_PAR2_A
	W_SD_W_PAR3_A
	W_SD_W_TOP_A
Syndicated Market Level	W_SYNDM_W_F
	W_SYNM_W_T_F
	W_SM_W_PAR1_A
	W_SM_W_PAR2_A
	W_SM_W_PAR3_A
	W_SM_W_TOP_A

14.4.5 How to Prepare Flat Files for Database Partition

This section contains the following topics:

- Section 14.4.5.1, "Syndicated Data and Market Flat File Format"
- Section 14.4.5.2, "Define the Relation Between Top Level Position and Partition Value in Flat File"

14.4.5.1 Syndicated Data and Market Flat File Format

The flat file format for syndicated data and market fact data load supports the Database Partition feature. The new field, COUNTRY_REGION_NAME, is added in each syndicated data and market flat file as the LAST position of the field.

If you are not using Database Partition, then in the flat file, the value for the COUNTRY_REGION_NAME field should be empty. If you want to use Database Partition, then the value of the COUNTRY_REGION_NAME field in each flat file must be equal to one of the partition names (and partition values) that are provided in the DDL script of Section 14.4.4, "How to Create Partition Table."

Table 14–28 represents the field definition used in each syndicated data and market flat file.

Note: For information about the flat file formats, see Section 14.3, "Syndicated Data Flat File Formats."

For syndicated data, see Section 14.3.1, "Syndicated Data Flat File for Direct Sales Data" through Section 14.3.6, "Syndicated Data Flat File for Weekly Early View Data."

For syndicated market, see Section 14.3.7, "Syndicated Market Flat File for Indirect Sales Data" through Section 14.3.11, "Syndicated Market Flat File for Weekly Early View Data."

Table 14–28 Field Definition for Partition Column in Syndicated Data and Market Flat File

Field Name	Data Type	Field Required	Field Description
COUNTRY_ REGION_NAME	VARCHAR2(30)	-	Country or Region name.[1]

[1] This field is the partition column and the value for this filed is null if Database Partition is disabled; else, enter the value that is provided as the partition value (and partition name) used in the DDL script. There must be no space in the string and all the characters must be in upper case. For example, text characters to be used in the partition column can in this format: 'USA', 'UNITEDKINGDOM' or 'FRANCE'.

14.4.5.2 Define the Relation Between Top Level Position and Partition Value in Flat File

Users logging in to the analytics server for Life Sciences, after Database Partition is enabled, require specific information with regard to the country or region to which they belong. This information is important for the user to access the relevant country or region related partition data. To provide this information to the user, you must prepare the Load_TopPostnPartition.txt flat file based on the format provided in Table 14–29.

In the flat file, provide unique and complete list of relation between the top level sales force position and partition value, use the pipe (|) symbol as the field separator, save the file as Load_TopPostnPartition.txt, and then copy the file into the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles), before starting to load syndicated data and market.

Table 14–29 Flat File Layout for Load_TopPostnPartition.txt

Field Name	Data Type	Field Required	Field Description
CURRENT_TOP_ LVL_POSTN	VARCHAR2 (50)	Not Null	Current Top Level Position Name[1]
COUNTRY_ REGION_NAME	VARCHAR2 (30)	Not Null	Country or Region name[2]

[1] Corresponding W_POSITION_DH. CURRENT_TOP_LVL_POSTN should be provided in the flat file. This field is case sensitive and matched case must be provided.

[2] The value of COUNTRY_REGION_NAME field must be equal to one of the partition names (and partition values) in the DDL script. There must be no space in the character string and all characters must be upper case. For example, use text like 'USA', 'UNITEDKINGDOM', or 'FRANCE'.

Note: Follow these rules to prepare the flat file. Failure to adhere to these critical rules when preparing the flat file may result in unexpected behavior when you try to log in to the analytics server configured for Life Sciences.

- The Current Top Level Position must have a 1:1 relationship with the COUNTRY_REGION_NAME value and the relationship is mandatory for each top level position.
- All subordinates to each top level position should have M:1 relationship with the parent positions.
- The CURRENT_TOP_LVL_POSTN field is case sensitive. The CURRENT_TOP_LVL_POSTN field value must be equal to the W_POSITION_DH.CURRENT_TOP_LVL_POSTN column value.
- The COUNTRY_REGION_NAME field value must be equal to the partition name and partition value defined in the DDL script that is used to create the partition table.
- The COUNTRY_REGION_NAME field values must not have spaces. And, the characters in the field must be all upper case.
- The flat file data is always loaded as full load into the W_TOP_POSTN_PARTITION_M table. There are no incremental insert or update supported.

14.4.6 How to Enable Database Partition Feature for Syndicated Data and Market Fact

After you create the partition tables, see Section 14.4.4, "How to Create Partition Table," you must enable Database Partition before loading the syndicated data and market fact table data. This feature is disabled by default in the Oracle Business Analytics Warehouse for Life Sciences.

Database Partition is dependent on the parameters in DAC. Table 14–30 contains the parent-level parameter list with descriptions to enable or disable Database Partition.

Table 14–30 Parent-Level Parameter for DB Partition

Parameter Name	Description
$$SYND_DS_PARTITION_ENABLED	Enable or disable (Y/N) DB Partition of Syndicated Data and Market facts for Direct Sales data.
$$SYND_IDS_PARTITION_ENABLED	Enable or disable (Y/N) DB Partition of Syndicated Data and Market facts for Indirect Sales data.
$$SYND_RX_PARTITION_ENABLED	Enable or disable (Y/N) DB Partition of Syndicated Data and Market facts for Prescription data.
$$SYND_XPT_PARTITION_ENABLED	Enable or disable (Y/N) DB Partition of Syndicated Data and Market facts for Physician Plan Level Rx data.
$$SYND_PT_PARTITION_ENABLED	Enable or disable (Y/N) DB Partition of Syndicated Data and Market facts for Plan Level Rx data.
$$SYND_W_PARTITION_ENABLED	Enable or disable (Y/N) DB Partition of syndicated data and market facts for Weekly Early View data.

Note: Each parameter in Table 14–30 is shared by both syndicated data and market fact tables. For example, if you set the value 'Y' for the $$SYND_DS_PARTITION_ENABLED parameter, then you have enabled Database Partition of both syndicated data and market fact tables for direct sales data.

Using the procedures provided in Section 14.4.3, "Identify Partition Name and Value" and Section 14.4.4, "How to Create Partition Table," and based on your decision to partition the syndicated data type, you can create proper partition tables. However, to activate Database Partition, after creating the partition tables, you must enable proper parameters based on the information provided in Table 14–30.

For example, if you want the partition feature enabled for only the direct sales data, then you must set the value 'Y' for the $$SYND_DS_PARTITION_ENABLED parameter. If you decide to have partition for all the syndicated data types, then you must set value 'Y' for all parameters listed in Table 14–30.

To enable Partition in DAC metadata:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab.
3. In the Name field, query for the $$SYND_*_PARTITION_ENABLED value.
4. From the query result, select the parameter to which you want to enable Partition.
5. For example, if you want to enable Partition for all the syndicated data types, then, select the $$SYND_DS_PARTITION_ENABLED parameter, set the Value field to Y, and then click Save.
6. Repeat the previous step for the other parameters defined in Table 14–30.
   - $$SYND_IDS_PARTITION_ENABLED
   - $$SYND_RX_PARTITION_ENABLED
   - $$SYND_XPT_PARTITION_ENABLED
   - $$SYND_PT_PARTITION_ENABLED
   - $$SYND_W_PARTITION_ENABLED

14.4.7 How to Disable Database Partition Feature for Syndicated Data and Market Fact

To disable Database Partition in DAC metadata (reverting to default.)

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab.
3. In the Name field, query for the $$SYND_*_PARTITION_ENABLED value.
4. From the query result, select the $$SYND_DS_PARTITION_ENABLED parameter, set the Value field to N, and then click Save.
5. Repeat the previous step for the other parameters defined in Table 14–30:
   - $$SYND_IDS_PARTITION_ENABLED
   - $$SYND_RX_PARTITION_ENABLED

- $$SYND_XPT_PARTITION_ENABLED
- $$SYND_PT_PARTITION_ENABLED
- $$SYND_W_PARTITION_ENABLED

14.4.8 Only the Oracle Platform is Supported for the Database Partition Feature in Default

Database Partition for syndicated data and market supports only the Oracle database platform; however, to use Database Partition for other database platforms, you must customize the current feature accordingly.

The major Database Partition features in Oracle Business Analytics Warehouse for Life Sciences are:

- The ability to track the active partition value for the current ETL during run time
- The ability to truncate the partition only by active partition value
- The ability to create index in partitioned table

The following sections describe the general design logic applied for Database Partition for the Oracle Business Analytics Warehouse for Life Sciences. Use this information to modify Partition for other database Platforms.

14.4.8.1 The W_SYND_PARTITION_TMP Table

The Active Partition (or Partitions) is the partition that is being processed in current ETL. The table W_SYND_PARTITION_TMP is used to process the active partition (or partitions) in the current ETL. This temporary table captures the active partition values from each syndicated data staging table that was already loaded from the flat file source. The information in the table is also applied to the syndicated market level data loading.

Table 14–31 Schema Definition of W_SYND_PARTITION_TMP Table

Column Name	Data Type	Column Required	Column Description
SOURCE	VARCHAR2(30)	NOT NULL	Indicates the data source[1]
COUNTRY_ REGION_NAME	VARCHAR2(30)	NOT NULL	Country or Region name, which is also the Active Partition Value (and Partition Name).

[1] Hard-coded values are used, refer Table 14–32 for the actual values.

- Source

  Indicates the source of the syndicated data staging table and type data.

Table 14–32 Value Definition Used in W_SYND_PARTITION_TMP.SOURCE

Value of Source	Description
W_SYNDD_DS_FS	Partition value comes from staging table source of Syndicated Data for Direct Sales data.
W_SYNDD_IDS_FS	Partition value comes from staging table source of Syndicated Data for Indirect Sales data.

Table 14–32 (Cont.) Value Definition Used in W_SYND_PARTITION_TMP.SOURCE

Value of Source	Description
W_SYNDD_RX_FS	Partition value comes from staging table source of Syndicated Data for Prescription data.
W_SYNDD_PT_FS	Partition value comes from staging table source of Syndicated Data for Plan Level Rx data.
W_SYNDD_XPT_FS	Partition value comes from staging table source of Syndicated Data for Physician Plan Level Rx data.
W_SYNDD_W_FS	Partition value comes from staging table source of Syndicated Data for Weekly Early View data.

- COUNTRY_REGION_NAME

  Active and unique partition value (same as partition name) used in syndicated data flat file during the current ETL run.

14.4.8.2 Dynamically Generate SQL for Active Partition to Truncate Partitioned Table

Depending on the active partition selected for the current ETL run, the SQL text to truncate the partitioned table is dynamically generated for the active partition.

For example, if there are three partitions of the syndicated data and syndicated market for direct sales data: USA, UNITEDKINGDOM, and FRANCE partitions, and for the current ETL run, the data available to load is only for U.S.A. Then, the active partition is USA and the ETL process must truncate the partitioned direct sales tables only for the USA partition.

The dynamic SQL text generation is done using the parameters available in DAC. Table 14–33 defines the parent-level parameters that are used for SQL text generation. The parent-level parameters simply call its child-level parameter listed in Table 14–34 respectively.

Table 14–33 Parent-Level Parameter for SQL Generation

Parent Name	Description
$$SYND_DS_PARTITION_TRUNCATE_SQL	Call child-level parameter of Syndicated Data and Market facts for Direct Sales data.
$$SYND_IDS_PARTITION_TRUNCATE_SQL	Call child-level parameter of Syndicated Data and Market facts for Indirect Sales data.
$$SYND_RX_PARTITION_TRUNCATE_SQL	Call child-level parameter of Syndicated Data and Market facts for Prescription data.
$$SYND_XPT_PARTITION_TRUNCATE_SQL	Call child-level parameter of Syndicated Data and Market facts for Physician Plan Level Rx data.
$$SYND_PT_PARTITION_TRUNCATE_SQL	Call child-level parameter of Syndicated Data and Market facts for Plan Level Rx data.
$$SYND_W_PARTITION_TRUNCATE_SQL	Call child-level parameter of Syndicated Data and Market facts for Weekly Early View data.

Table 14–34 defines the child-level parameters available for SQL generation. Based on the data captured for the active partition in the W_SYND_PARTITION_TMP table, the parameter generates actual SQL text to truncate the partitioned table for all active partitions as concatenating each SQL statement. In the previous example, if all the

partitions for the three countries are active during the current ETL run, then the child-level parameter is dynamically generated as follows:

```
execute immediate 'ALTER TABLE getTableName() TRUNCATE PARTITION "USA"';
execute immediate 'ALTER TABLE getTableName() TRUNCATE PARTITION "UNITEDKINGDOM"';
execute immediate 'ALTER TABLE getTableName() TRUNCATE PARTITION "FRANCE"';
```

The child-level parameters are designed to use Oracle specific syntax and are executed only in the Oracle database platform. In other database platforms, the parameters will do nothing.

Table 14–34 Child-Level Parameter for SQL Generation

Parameter Name	Description
$$SYND_DS_PARTITION_TRUNCATE_SQL_TEXT	Concatenated SQL text to truncate partitioned Syndicated Data and Market facts for Direct Sales data. The SQL is only for active partitions.
$$SYND_IDS_PARTITION_TRUNCATE_SQL_TEXT	Concatenated SQL text to truncate partitioned Syndicated Data and Market facts for Indirect Sales data. The SQL is only for active partitions.
$$SYND_RX_PARTITION_TRUNCATE_SQL_TEXT	Concatenated SQL text to truncate partitioned Syndicated Data and Market facts for Prescription data. The SQL is only for active partitions.
$$SYND_XPT_PARTITION_TRUNCATE_SQL_TEXT	Concatenated SQL text to truncate partitioned Syndicated Data and Market facts for Physician Plan Level Rx data. The SQL is only for active partitions.
$$SYND_PT_PARTITION_TRUNCATE_SQL_TEXT	Concatenated SQL text to truncate partitioned Syndicated Data and Market facts for Plan Level Rx data. The SQL is only for active partitions.
$$SYND_W_PARTITION_TRUNCATE_SQL_TEXT	Concatenated SQL text to truncate partitioned Syndicated Data and Market facts for Weekly Early View data. The SQL is only for active partitions.

14.4.8.3 Method to Truncate Partitioned Table for Syndicated Data and Market Base Fact Tables

In Release 7.9.6.3 of Oracle Business Analytics Warehouse, the Actions function is provided in DAC. Actions are used to override the existing truncate table function embedded by default in DAC, execute the SQL generated by Section 14.4.8.2, "Dynamically Generate SQL for Active Partition to Truncate Partitioned Table" to truncate partitioned tables.

Since syndicated data and market base fact tables are affected by both Database Partition and Restartability features; the syndicated data and market base fact tables are designed slightly different from the other fact tables. For more details on the Restartability feature, see Section 14.5, "Supporting Restartability in Syndicated Data and Market Load."

Table 14–35 provides information on the Task Action for syndicated data and market base fact tables.

Table 14–35 Task Action for Syndicated Data and Market Base Fact Tables

Task Action Name	Description
Task Action: LS Syndicated Data Fact_DirectSales Truncate Table	Task Action to truncate table of Syndicated Data and Market facts for Direct Sales data.

Table 14–35 (Cont.) Task Action for Syndicated Data and Market Base Fact Tables

Task Action Name	Description
Task Action: LS Syndicated Data Fact_IndirectSales Truncate Table	Task Action to truncate table of Syndicated Data and Market facts for Indirect Sales data.
Task Action: LS Syndicated Data Fact_Rx Truncate Table	Task Action to truncate table of Syndicated Data and Market facts for Prescription data.
Task Action: LS Syndicated Data Fact_ExponentPlantrak Truncate Table	Task Action to truncate table of Syndicated Data and Market facts for Physician Plan Level Rx data.
Task Action: LS Syndicated Data Fact_Plantrak Truncate Table	Task Action to truncate table of Syndicated Data and Market facts for Plan Level Rx data.
Task Action: LS Syndicated Data Fact_ WeeklyEarlyView Truncate Table	Task Action to truncate table of Syndicated Data and Market facts for Weekly Early View data.

The main logic of the Task Action is as follows:

- In the Oracle Database platform

  When Database Partition is enabled, execute the parent-level parameter such as $$SYND_DS_PARTITION_TRUNCATE_SQL listed in Table 14–33 to truncate the partitioned table.

  When Database Partition is disabled, execute the normal truncation for the complete table.

- In other Database platforms

  Execute normal truncation for the complete table.

The Task Action definition can be found in DAC. To view the definition:

1. In DAC, choose Tools, then Seed Data, then Actions, then Task Actions.
2. Query the Name field for the Task Action: LS Syndicated Data Fact*Truncate Table value.
3. From the query result, select the Task Action that you want to review.
4. Click the Value field to see the definition.

The Task Action is called by each task of syndicated data and market territory or district level and aggregate facts to execute the definition:

Example to review for syndicated data direct sales base fact.

1. In DAC, go to the Design view, and display the Tasks tab.
2. Query the Name field, as follows:
   - When you enable DB Partition but disable Restartability, query the Name field for the LS Load into SyndicatedData Fact for Direct Sales value.
   - When you enable both DB Partition and Restartability, query the Name field for the LS Load into SyndicatedData Fact for Direct Sales_Restart value.
3. From the query result, select the Actions tab on the bottom.

4. Locate the Task Action: LS Syndicated Data Fact_DirectSales Truncate Table value embedded in the Action field.

When you start to run ETL, the task action embedded in the task is executed.

14.4.8.4 Method to Truncate Partitioned Table for Syndicated Data and Market Territory or District Level and Aggregate Fact Tables

Syndicated data and market territory or district level and aggregate fact tables are not affected by Restartability. Instead of Task Action, Table Action is used to truncate partitioned tables. Table 14–36 represents the Table Action for the syndicated data and market territory or district level and aggregate fact tables and description.

Table 14–36 Table Action for Syndicated Data and Market Territory or District Level and Aggregate Fact Tables

Table Action Name	Description
Table Action: LS Syndicated Data Fact_DirectSales Truncate Table	Table Action to truncate table of Syndicated Data and Market facts for Direct Sales data.
Table Action: LS Syndicated Data Fact_IndirectSales Truncate Table	Table Action to truncate table of Syndicated Data and Market facts for Indirect Sales data.
Table Action: LS Syndicated Data Fact_RX Truncate Table	Table Action to truncate table of Syndicated Data and Market facts for Prescription data.
Table Action: LS Syndicated Data Fact_ExponentPlantrak Truncate	Table Action to truncate table of Syndicated Data and Market facts for Physician Plan Level Rx data.
Table Action: LS Syndicated Data Fact_Plantrak Truncate Table	Table Action to truncate table of Syndicated Data and Market facts for Plan Level Rx data.
Table Action: LS Syndicated Data Fact_ WeeklyEarlyView Truncate Table	Table Action to truncate table of Syndicated Data and Market facts for Weekly Early View data.

Table Action logic is similar to Task Action logic. For more details on Task Action, see Section 14.4.8.3, "Method to Truncate Partitioned Table for Syndicated Data and Market Base Fact Tables":

- In Oracle DB platform

  When Database Partition is enabled, execute the parent-level parameter such as $$SYND_DS_PARTITION_TRUNCATE_SQL listed in Table 14–33 to truncate the partitioned table.

  When Database Partition is disabled, execute the normal truncation for the entire table.

- In other DB platform

  Execute normal truncation for entire table.

To find the Table Action definition in DAC:

1. In DAC, choose Tools, then Seed Data, then Actions, then Table Actions.
2. Query the Name field for the Table Action: LS Syndicated Data Fact*Truncate Table value.

3. From the query result, select the Table Action that you want to review.
4. Click the Value field to see the definition.

The Table Action is called by each table of syndicated data and market territory or district level and aggregate facts to execute the definition:

Example to review for syndicated data direct sales territory level fact.

1. In DAC, go to the Design view, and then display the Tables tab.
2. Query the Name field for the W_SYND_DS_T_F value.
3. From the query result, select the Actions tab on the bottom.
4. Locate the Table Action: LS Syndicated Data Fact_DirectSales Truncate Table value embedded in the Action field.

Once you start to run ETL, the Table Action embedded in the table is executed.

14.4.8.5 Method to Create Index for Partitioned Syndicated Data and Market Fact Tables

Two Index Actions (the Local Bitmap and Local Unique indexes) are provided for index creation on the partitioned table. The Local Bitmap index is available within each partition and the Local Unique index includes the COUNTRY_REGION_NAME partition column as the first leading unique index column within each partition.Table 14–37 represents the Index Action for Syndicated Data and Market fact tables and description.

Table 14–37 Index Action for Syndicated Data and Market Fact Tables

Syndicated Data Type	Index Action Name	Description
Syndicated Data and Market facts for Direct Sales data.	1. Index Action:LS Syndicated Data Fact_ DirectSales Create BITMAP Index 2. Index Action:LS Syndicated Data Fact_ DirectSales Create UNIQUE Index	1. Index action to create BITMAP index. 2. Index action to create UNIQUE index.
Syndicated Data and Market facts for Indirect Sales data.	1. Index Action:LS Syndicated Data Fact_ IndirectSales Create BITMAP Index 2. Index Action:LS Syndicated Data Fact_ IndirectSales Create UNIQUE Index	1. Index action to create BITMAP index. 2. Index action to create UNIQUE index.
Syndicated Data and Market facts for Prescription data.	1. Index Action:LS Syndicated Data Fact_RX Create BITMAP Index 2. Index Action:LS Syndicated Data Fact_RX Create UNIQUE Index	1. Index action to create BITMAP index. 2. Index action to create UNIQUE index.

Table 14–37 (Cont.) Index Action for Syndicated Data and Market Fact Tables

Syndicated Data Type	Index Action Name	Description
Syndicated Data and Market facts for Physician Plan Level Rx data.	1. Index Action:LS Syndicated Data Fact_ ExponentPlantrak Create BITMAP Index 2. Index Action:LS Syndicated Data Fact_ ExponentPlantrak Create UNIQUE Index	1. Index action to create BITMAP index. 2. Index action to create UNIQUE index.
Syndicated Data and Market facts for Plan Level Rx data.	1. Index Action:LS Syndicated Data Fact_ Plantrak Create BITMAP Index 2. Index Action:LS Syndicated Data Fact_ Plantrak Create UNIQUE Index	1. Index action to create BITMAP index. 2. Index action to create UNIQUE index.
Syndicated Data and Market facts for Weekly Early View data.	1. Index Action:LS Syndicated Data Fact_ WeeklyEarlyView Create BITMAP Index 2. Index Action:LS Syndicated Data Fact_ WeeklyEarlyView Create UNIQUE Index	1. Index action to create BITMAP index. 2. Index action to create UNIQUE index.

The main logic of the Index Action for the BITMAP index is:

- In Oracle database platform

  If Database Partition is enabled, then execute Index Action to create the BITMAP index as LOCAL.

  If Database Partition is disabled, then execute Index Action to create the BITMAP index as normal.

- In other database platform

  Execute Index Action to create the BITMAP index as normal.

The main logic of the Index Action for UNIQUE index is:

- In all database platforms

  If Database Partition is enabled, then execute the Index Action to create the UNIQUE index as LOCAL; including COUNTRY_REGION_NAME as the first leading index column value.

  If Database Partition is disabled, then execute the Index Action to create the UNIQUE index as normal.

To locate the Index Action definition in DAC:

1. In DAC, choose Tool, then Seed Data, then Actions, then Index Actions.
2. Query the Name field, as follows:
   - For the BITMAP, index as Index Action:LS Syndicated Data Fact*Create BITMAP Index

- For the UNIQUE, index as Index Action:LS Syndicated Data Fact*Create UNIQUE Index

3. From the query result, select the Index Action that you want to review.
4. Click the Value field to see the definition.

The Index Action is called by each index of syndicated data and market facts to execute the definition:

Example to review for syndicated data direct sales base fact.

1. In DAC, go to the Design view, and then display the Indices tab.
2. Query the Table Name field for the W_SYNDD_DS_F value.
3. From the query result, select the W_SYNDD_DS_F_F1 index and then navigate to the Actions tab on the bottom.

   You can find that the Index Action:LS Syndicated Data Fact_DirectSales Create BITMAP Index value is embedded in Action field.
4. From the query result, select the W_SYNDD_DS_F_U1 index and then navigate to the Actions tab on the bottom.

   You can find that the Index Action:LS Syndicated Data Fact_DirectSales Create UNIQUE Index value is embedded in the Action field.

   When you start to run ETL, the Index Action embedded in the indexes is executed.

14.5 Supporting Restartability in Syndicated Data and Market Load

This section contains the following sections:

- Section 14.5.1, "Overview"
- Section 14.5.2, "How to Enable Restartability Feature in DAC for Syndicated Data and Market Base Fact Tables"
- Section 14.5.3, "How to Disable Restartability in DAC for Syndicated Data and Market Base Fact Tables"

14.5.1 Overview

In Oracle Business Analytics Warehouse for Life Sciences, syndicated data and market always do full load only. This is because syndicated data and market do not come with a key to uniquely track each transaction among ETL runs. In previous releases, if ETL failed in the middle and data was loaded partially, you had to rerun the ETL to load data from the start again. Data that had been loaded before the failed point would be removed as well. This is not very efficient.

In Release 7.9.6.3, instead of loading the data from the very beginning, if you have enabled Restartability, you can resume the ETL and continue from the last failed point. However, Restartability is disabled by default in the DAC metadata. To enable Restartability, you must activate the feature in DAC after installing the Oracle Business Analytics Warehouse.

> **Note:** Restartability is available when you are loading the syndicated data and market data into the base fact tables. The feature is not available when you are loading data into territory or district level and aggregate fact tables.

14.5.2 How to Enable Restartability Feature in DAC for Syndicated Data and Market Base Fact Tables

This section contains the following sections:

- Section 14.5.2.1, "How to Enable Restartability Feature in DAC for Syndicated Data Base Fact"
- Section 14.5.2.2, "How to Enable Restartability Feature in DAC for Syndicated Market Base Fact"
- Section 14.5.2.3, "How to Reassemble Subject Areas and Rebuild Execution Plans"

14.5.2.1 How to Enable Restartability Feature in DAC for Syndicated Data Base Fact

To enable Restartability for flat file loading method for the syndicated data base facts:

1. In DAC, go to the Design view, and then display the Tasks tab.
2. Query the Name field for the LS Load into SyndicatedData Fact* value.
3. From the query result, select the LS Load into SyndicatedData Fact for Direct Sales task, select Inactive, and then click Save.
4. Repeat the previous step for the following tasks:
   - LS Load into SyndicatedData Fact for Indirect Sales
   - LS Load into SyndicatedData Fact for Exponent Plantrak
   - LS Load into SyndicatedData Fact for Plantrak
   - LS Load into SyndicatedData Fact for Rx
   - LS Load into SyndicatedData Fact for Weekly Early View
5. Select the LS Load into SyndicatedData Fact for Direct Sales_Restart task, clear Inactive, and then click Save.
6. Repeat the previous step for the following tasks:
   - LS Load into SyndicatedData Fact for Indirect Sales_Restart
   - LS Load into SyndicatedData Fact for Exponent Plantrak_Restart
   - LS Load into SyndicatedData Fact for Plantrak_Restart
   - LS Load into SyndicatedData Fact for Rx_Restart

14.5.2.2 How to Enable Restartability Feature in DAC for Syndicated Market Base Fact

There are two methods to load the syndicated market base fact tables in Oracle Business Analytics Warehouse for Life Sciences.

- Calculated Loading Method

  Use this method if you have purchased flat files for only syndicated data from the third party vendor. To do Calculated Loading, load the syndicated data with the flat files into the data warehouse for the Life Sciences Analytics, syndicated data is aggregated from product level, calculate the data based on the product-market hierarchy, then load the data as market level data into the syndicated market base facts. See Section 14.2, "Importing Syndicated Data into Oracle Business Analytics Warehouse."

- Flat File Loading Method

Use this method if you have purchased flat files for syndicated data and syndicated market from the third party vendor. The flat files are loaded directly into syndicated data and market base facts for Life Sciences Analytics.

Calculated Loading Method for the syndicated market base facts is enabled by default. However, due to difficulty to track the same unique key value in aggregated data, Restartability is not supported for the Calculated Loading Method for syndicated market base facts.

Restartability is supported only for Flat File Loading Method for syndicated market base facts.

Similar to syndicated data base facts, Restartability is disabled by default in DAC metadata. To use Restartability, activate the feature in DAC.

To enable Restartability for Flat File Loading Method for the syndicated market base Facts:

1. In DAC, go to the Design view, and then display the Tasks tab.
2. Query the Name field for the LS Load into SyndicatedMarket Fact* value.
3. From the query result, select the LS Load into SyndicatedMarket Fact for Calculated Exponent Plantrak task, select Inactive, and then click Save.

   Note: Calculated Loading Method for syndicated market base fact table is disabled.
4. Repeat the previous step for the following tasks:
   - LS Load into SyndicatedMarket Fact for Calculated Indirect Sales
   - LS Load into SyndicatedMarket Fact for Calculated Plantrak
   - LS Load into SyndicatedMarket Fact for Calculated Rx
   - LS Load into SyndicatedMarket Fact for Calculated Weekly Early View

   Do not change existing settings for the LS Load into SyndicatedMarket Fact for Calculated Direct Sales task. : There is no flat file available for syndicated market for direct sales data from the third party vendor. Use only Calculated Loading Method to load direct sales data.
5. Select the LS Load into SyndicatedMarket Fact for Exponent Plantrak_Restart task, clear Inactive, and then click Save.

   Note: For the syndicated market base fact, Flat File Loading Method with Restartability is enabled.
6. Repeat the previous step for the following tasks:
   - LS Load into SyndicatedMarket Fact for Indirect Sales_Restart
   - LS Load into SyndicatedMarket Fact for Plantrak_Restart
   - LS Load into SyndicatedMarket Fact for Rx_Restart
   - LS Load into SyndicatedMarket Fact for Weekly Early View_Restart

Note: Depending on your business requirement, select the syndicated market type that should have Restartability.

For example, if you want to have Restartability only for the Physician Plan Level Rx data, then disable the LS Load into SyndicatedMarket Fact for Calculated Exponent Plantrak task in Step 4 and enable only the LS Load into SyndicatedMarket Fact for Exponent Plantrak_ Restart task in Step 6. Do not repeat the procedure for Step 5 and 7 for other syndicated data types.

14.5.2.3 How to Reassemble Subject Areas and Rebuild Execution Plans

After enabling Restartability for both the syndicated data and market base facts, reassemble Subject Areas and rebuild Execution Plans to include the changes into the next ETL run. For information about assembling Subject Areas and building Execution Plans, see *Oracle Business Intelligence Data Warehouse Administration Console Guide*.

Note: If you enable Restartability only for the direct sales data, then reassemble only the corresponding Subject Areas and Execution Plans.

For example, reassemble CRM ePharma–Direct Sales, and CRM ePharma–Objective Subject Areas.

To reassemble Subject Areas:

1. In DAC, go to the Design view, and then display the Subject Areas tab.
2. Query the Name field for the CRM ePharma*Sales or CRM ePharma*Prescription or CRM ePharma*Objective values.
3. Click the Assemble button.
4. In the pop up window, select All records in the list and then click OK.
5. For each Subject Area, click Calculate Task List.

   A list of tasks that must be assembled is displayed.
6. Check the task list and click Accept.
7. After the Subject Areas are assembled for Life Sciences Analytics, check if Restartability is properly included in each Subject Area.
   a. For example, select the CRM ePharma - Direct Sales Subject Area and display the Tasks tab on the bottom.
   b. Query the Name field for the *Restart value.

Note: If the query returns results, then reassembling the Subject Area is successful.

To rebuild Execution Plans:

1. In DAC, go to the Execute view, and then display the Execution Plans tab.
2. Select the Execution Plan that you want to run, for example, select the CRM ePharma - Siebel 8.1.1 Execution Plan and click Build.
3. In the pop up window, select the Selected record only option, and then click OK.

4. After the Execution Plan is re-built for Life Sciences Analytics, check the Execution Plan.

    a. Select the CRM ePharma - Siebel 8.1.1 Execution Plan, and display the Ordered Tasks tab on the bottom.

    b. Query the Name field for the *Restart value.

> **Note:** If the query returns result(s), then the Execution Plan is re-built successfully and Restartability is enabled for ETL.

14.5.3 How to Disable Restartability in DAC for Syndicated Data and Market Base Fact Tables

This section explains how to disable Restartability for Syndicated Data and Market Base Fact Tables.

14.5.3.1 How to Disable Restartability in DAC for Syndicated Data Base Fact

To disable Restartability for Flat File Loading Method for syndicated data:

1. In DAC, go to the Design view, and then display the Tasks tab.
2. Query the Name field for the LS Load into the SyndicatedData Fact* value.
3. From the query result, select the LS Load into SyndicatedData Fact for Direct Sales task, clear Inactive, and then click Save.
4. Repeat the previous step for the following tasks:
    - LS Load into SyndicatedData Fact for Indirect Sales
    - LS Load into SyndicatedData Fact for Exponent Plantrak
    - LS Load into SyndicatedData Fact for Plantrak
    - LS Load into SyndicatedData Fact for Rx
    - LS Load into SyndicatedData Fact for Weekly Early View
5. Select the LS Load into SyndicatedData Fact for Direct Sales_Restart task, select Inactive, and then click Save.
6. Repeat the previous step for the following tasks:
    - LS Load into SyndicatedData Fact for Indirect Sales_Restart
    - LS Load into SyndicatedData Fact for Exponent Plantrak_Restart
    - LS Load into SyndicatedData Fact for Plantrak_Restart
    - LS Load into SyndicatedData Fact for Rx_Restart
    - LS Load into SyndicatedData Fact for Weekly Early View_Restart

14.5.3.2 How to Disable Restartability in DAC for Syndicated Market Base Fact

To disable Restartability for Flat File Loading Method for syndicated market base facts (reset to default):

1. In DAC, go to the Design view, and then display the Tasks tab.
2. Query the Name field for the LS Load into SyndicatedMarket Fact* value.
3. From the query result, select the LS Load into SyndicatedMarket Fact for Calculated Exponent Plantrak task, clear Inactive, and then click Save.

> **Note:** For the syndicated market base fact, Calculated Loading Method is enabled.

4. Repeat the previous step for the following tasks:
   - LS Load into SyndicatedMarket Fact for Calculated Indirect Sales
   - LS Load into SyndicatedMarket Fact for Calculated Plantrak
   - LS Load into SyndicatedMarket Fact for Calculated Rx
   - LS Load into SyndicatedMarket Fact for Calculated Weekly Early View

   Note: The LS Load into SyndicatedMarket Fact for Calculated Direct Sales task must be active all the time.
5. Select the LS Load into SyndicatedMarket Fact for Exponent Plantrak_Restart task, select Inactive, and then click Save.
6. Repeat the previous step for the following tasks:
   - LS Load into SyndicatedMarket Fact for Indirect Sales_Restart
   - LS Load into SyndicatedMarket Fact for Plantrak_Restart
   - LS Load into SyndicatedMarket Fact for Rx_Restart
   - LS Load into SyndicatedMarket Fact for Weekly Early View_Restart

14.5.3.3 How to Reassemble Subject Areas and Rebuild Execution Plans

After disabling Restartability for both syndicated data and market base facts, reassemble Subject Areas and rebuild Execution Plans to include the changes into the next ETL run. For information about assembling Subject Areas and building Execution Plans, see *Oracle Business Intelligence Data Warehouse Administration Console Guide*.

> **Note:** If you disabled Restartability for only direct sales data, then reassemble only the corresponding Subject Areas and Execution Plans.
>
> For example, reassemble CRM ePharma – Direct Sales, and CRM ePharma – Objective Subject Areas.

To reassemble Subject Areas:

1. In DAC, go to the Design view, and then display the Subject Areas tab.
2. Query the Name field for the "CRM ePharma*Sales" or "CRM ePharma*Prescription" or "CRM ePharma*Objective" values.
3. Click the Assemble button.
4. In the pop up window, select All records in the list and then click OK.
5. For each Subject Area, click Calculate Task List.

   A list of tasks that must be assembled is displayed.
6. Check the task list and click Accept.
7. After the Subject Areas are assembled for Life Sciences Analytics, check if Restartability is removed in each Subject Area.

To rebuild Execution Plans:

1. In DAC, go to the Execute view, and then display the Execution Plans tab.
2. Select the Execution Plan that you want to rebuild, for example, select the CRM ePharma - Siebel 8.1.1 Execution Plan and click Build.
3. In the pop up window, select the Selected record only option, and then click OK.
4. After the Execution Plan is re-built for Life Sciences Analytics, check the Execution Plan to see if Restartability is removed.

14.6 Life Sciences Data Loading Issues with Oracle Business Analytics Warehouse

This issue is specific to Analytics for Life Sciences and does not affect other products.

The ETL process updates the Oracle Business Analytics Warehouse for Life Sciences with a full refresh or using incremental updates. In DAC, you can run a full load by selecting Tools, then ETL Management, then Reset Data Warehouse. This procedure is usually only used for the initial build. If you run the same Execution Plan without confirming the Reset Data Warehouse again, then the ETL incrementally updates the Oracle Business Analytics Warehouse.

Running a full refresh load by confirming the Reset Data Warehouse feature deletes all existing information stored in the fact and dimension tables. The following types of incremental update rules are supported on the fact tables.

- **Account Call Fact.** Adds new records (call activity for account) to the fact table.
- **Attendee Call Fact.** Adds new records (call activity for attendee) to the fact table.
- **Contact Call Fact.** Adds new records (call activity for contact) to the fact table.
- **Medical Education Fact**. Adds new records to the fact table.
- **Objective Fact**. Adds new records or updates existing records to the fact table.
- **Profile Ranking Fact**. Adds new records or updates existing records to the fact table.
- **Syndicated Data and Market - Base Direct Sales Fact, Territory Direct Sales Fact, Base Indirect Sales Fact, Territory Indirect Sales Fact, District Indirect Sales Fact, Base Physician Rx Fact, Territory Physician Rx Fact, Base Rx Plan Fact, Territory Rx Plan Fact, Base Weekly Physician Rx Fact, Territory Weekly Physician Rx Fact, Base Physician Rx Plan Fact, Territory Physician Rx Plan Fact, District Physician Rx Plan Fact.** ETL does not support incremental updates. When running the full refresh ETL, all records in the fact and dimension tables are deleted. To maintain a history in the dimension tables (such as multiple alignments), use the incremental ETL. If you need to incrementally update the syndicates data fact tables for incremental syndicated data loading, use one of the following strategies:
  - **For incremental insert.** Prepare flat file source data that has new data with the new INTEGRATION_ID. Load the data directly into the staging table and modify the session not to truncate the fact tables. Then use the existing ETL to load data into fact tables.
  - **For incremental update.** Create new mappings that do a lookup against the fact tables and run the update. Ensure that the INTEGRATION_ID used in the flat file source and in the target tables are not identical. Because the incoming syndicated data in the flat file going to the target tables is in the normalized format, the INTEGRATION_ID must be manipulated properly. However, the

constant rule is applied when the INTEGRATION_ID is manipulated during the ETL load. The syndicated data during the first bulk load has the 'original in-coming INTEGRATION_ID' || '-1'; the 26th bulk load has the 'original in-coming INTEGRATION_ID' || '-26'.

14.6.1 Known Issues with the Syndicated Data Fact Tables

The following are known issues with creation of aggregate measure columns in the Syndicated Data fact tables.

- With large volumes of syndicated data, the creation of aggregate measures in the Oracle Business Analytics Warehouse can take four times the amount of time needed to load the fact table. This may be unacceptable under circumstances when there are large amounts of syndicated data.
  - Incremental Updates in the Oracle Business Analytics Warehouse LS Dimension Tables.
  - MAT aggregate measures are handled by the ETL process and not by metadata. All other aggregate measures are handled by metadata.

14.7 Incremental Updates in the Oracle Business Analytics Warehouse LS Dimension Tables

This issue is specific to Oracle Business Analytics Warehouse for Life Sciences and does not affect other products. In the LS dimension tables the following incremental updates are supported. Some dimensions have more than one set of attributes.

14.7.1 W_ALIGNMT_DH

Base Dimensional Hierarchy Table.

Attribute. Zip code, Brick, or Account/Contact Hierarchy depends on what alignment item type is selected.

Note: In release 7.9, all Zip Code, Brick and Account/Contact hierarchy alignment can be loaded.

Description. Contains the current version of alignment only.

14.7.2 W_ALIGNVER_DH

Slowly Changing Dimension Type.

Attribute. Historical Zip code, Brick, or Account/Contact Hierarchy depends on what alignment item type is selected.

Note: In release 7.9, all Zip Code, Brick and Account/Contact hierarchy alignment can be loaded.

Description. Tracks historical alignments. Note the following:

- Depending on selected alignment type in AlignmentType.csv, only the selected alignment rules are extracted and loaded into the Oracle Business Analytics

Warehouse. You can change the alignment type only when you run a full refresh ETL.

> **Note:** In release 7.9, all Zip Code, Brick and Account/Contact alignment types can be extracted and loaded.

  - When the alignment rule is loaded into the Oracle Business Analytics Warehouse, the current alignment rule is stored in the W_ALIGNMT_DH base table and the existing historical alignment rules are stored in W_ALIGNVER_DH table.

- The creation of Alignment Versions is controlled though the setting of the Alignment Version flag in the DAC console. In Pharma Analytics 7.9 this flag is now set in the Source System Parameters tab in the DAC Design view. The parameter name is '$$ALIGNMENT_VER' and takes a value of either Y or N.
  - If the $$ALIGNMENT_VER parameter is set to N, then the ETL makes changes to existing alignments if there is a change on an assignment rule.
  - If the $$ALIGNMENT_VER parameter is set to Y, it creates a new alignment version. Any existing alignment moves to the W_ALIGNVER_DH table as history. The W_ALIGNVER_DH table is used only when the $$ALIGNMENT_VER parameter is set to Y.

The dimension has these characteristics:

- The first alignment after Full load is 1.
- Incremental Updates in the Oracle Business Analytics Warehouse LS Dimension Tables
- A new version is created when a new assignment rule (new position or position relationship) or an assignment criteria is modified (change in postal code, brick, contact, or account) if the Alignment Version flag is set to Y.
- Assignment criteria:
  - Contact ZIP Code or Account ZIP Code cannot use ZIP Code ranges. Each ZIP Code assigned to a territory must be on a separate row, so the same value must be entered for both ZIP Code Low and ZIP Code High. For example, if ZIP Code 09654 is assigned to a territory, the value for both ZIP Code High and ZIP Code Low should be 09654. Also, the same ZIP Code should not be assigned twice to the same territory and a ZIP Code should be unique to a territory when using assignment criteria for Contact ZIP Code or Account ZIP Code.
  - Contact Brick or Account Brick require unique bricks assigned to a territory.
  - Every new version increases the counter by one and is a sequential number. Any pre-existing history version alignment data moves to W_ALIGNVER_DH historical alignment table whenever a new version is created. That is, the W_ALIGNMT_DH table always maintains the latest version while the W_ALIGNVER_DH table maintains the rest of the historical version if the $$ALIGNMENT_VER parameter is set to Y. So, when the historical alignment dimension table gets too large, your administrator can create a SQL query to delete all attributes for version numbers from the W_ALIGNVER_DH table.
  - Effective date of the alignment is assignment activation date.

15 Configuring Oracle Project Analytics

This section describes how to configure Oracle Project Analytics. It contains the following topics:

15.1 Overview of Oracle Project Analytics

Oracle Project Analytics offers organizations a comprehensive analytics solution that delivers pervasive insight into forecast, budgets, cost, revenue, billing, profitability, and other aspects of project management to help effectively track project life cycle status. It provides consolidated and timely information that is personalized, relevant, and actionable to improve performance and profitability. Oracle Project Analytics is also integrated with other applications in the Oracle BI Applications family to deliver cross functional analysis, such as AR and AP invoice aging analysis and procurement transactions by project.

Oracle Project Analytics provides role-based reporting and analysis for various roles involved in the project life cycle. Typical roles include Project Executive, Project Manager, Project Cost Engineer/Analyst, Billing Specialist, Project Accountant and Contract Administrator.

Executives can closely monitor the organization's performance and the performance of the projects that the organization is responsible for by looking into a particular program and project and verifying how the period, accumulated, or estimated-at-completion cost is doing compared to budget and forecast. Cost variances and trends can be analyzed so that prompt actions can be taken to get projects on track or make any necessary changes in estimates, minimizing undesired results and reactive measures.

Oracle Project Analytics shows past, present, and future performance, and includes estimated metrics at project completion. Further analysis can be done on each project by drilling down to detailed information including profitability and cost at the task level.

Project managers can view the projects that they are responsible for, compare key metrics between projects, and analyze the details of a particular project such as cost distribution by task, resource, and person. Oracle Project Analytics provides a comprehensive, high-level view of accumulated and trending information for a single project or group of projects, as well as detailed information, such as budget accuracy and details by project and financial resource. Project managers can view cost and revenue by task, expenditure category, or resource type; and by project or resource. The level of analysis can be as granular as cost, revenue, or budget transaction.

Oracle Project Analytics provides out-of-the-box adapters for Oracle EBS 11.5.10 (Family Pack M) and R12, and PeopleSoft 8.9 and 9.0. It also provides universal adapters to extract and load data from legacy sources such as homegrown systems or from sources that have no prepackaged source-specific ETL adapters.

Oracle Project Analytics application comprises the following Subject Areas:

- **Funding.** A detailed Subject Area that provides the ability to track Agreement Amount, Funding Amount, Baselined Amount, and all changes in funding throughout the life cycle of the project. In addition, it provides the ability to do comparative analysis of Agreement Amount, Funding Amount, Invoice Amount, and the remaining funding amount across projects, tasks, customers, organizations, and associated hierarchies.
- **Budgets.** A detailed Subject Area that provides the ability to report on Cost Revenue, Margin for Budgets, and Budget changes including tracking original and current budgets across projects, tasks, organizations, resources, periods and associated hierarchies at budget line level. The product supports two methods for distributing Budgeted amounts across a budget line time period:
  - Allocating the whole amount to the period where the Budget Line Start date falls. This is supported by the Logical Fact "Fact - Project Budget."
  - Using a linear distribution to split the budget amount among the budget line period. This is supported by the logical fact "Fact - Project Budget - Linear Spread." For more information about this fact, see Section 15.1.1, "Linear Spread Metrics for Project Budget Fact."
- **Forecast.** A detailed Subject Area that provides the ability to report on Cost, Revenue and Margin for Forecasts, and Forecast changes. Forecast change analysis includes tracking original and current forecasts across projects, tasks, organizations, resources, periods and associated hierarchies. It provides the ability to track the metrics that indicate the past, present and future performance of cost, revenue, and margin.
- **Cost.** A detailed Subject Area that provides the ability to report on Cost (Burdened Cost), Raw Cost, Burden Cost for the past and current periods including inception-to-date, year-to-date comparisons across projects, tasks, organizations, resources, suppliers, and associated hierarchies. It provides the ability to track the cost at cost distribution level.
- **Revenue.** A detailed Subject Area that provides the ability to report on Revenue transactions for the past, and current periods including inception-to-date, year-to-date comparisons across projects, tasks, organizations, resources, and associated hierarchies. It provides the ability to track the revenue at Revenue distribution level.
- **Billing.** A detailed Subject Area that provides the ability to report on Billing Amount, Retention Amount, Unearned Amount, and Unbilled Receivables

Amounts across the projects, tasks, organizations, resources, and associated hierarchies. It provides the ability to track the invoice amount at invoice (draft invoice) line level only. Note: Invoice tax amount is not captured in this release.

- **Performance.** A consolidated Subject Area that includes combined information from Budgets, Forecasts, Cost, Revenue, and provides the ability to do performance by comparing the actual (cost, revenue, margin and margin percentage) with budgets, and forecasts across projects, tasks, organizations, resources, and associated hierarchies.
- **Project Commitments.** For Oracle EBS source systems only. A detailed Subject Area that provides the ability to report on project commitments, which include total raw and burdened amounts for requisitions, purchase orders, and supplier invoices for organizations, projects, tasks, resources, suppliers, and associated hierarchies. The subject area provides the ability to track commitments at the commitment document level.

15.1.1 Linear Spread Metrics for Project Budget Fact

In the Oracle EBS Project application, the Budget Line amount is split equally between all the days (including weekend days) in the budget line duration when calculating the Cost Amount, Cost ITD Amount, or Cost PTD Amount for a certain Period. By default, Oracle Project Analytics allocates the full Budget Line Amount to the Period containing the Budget Line Start Date. All the necessary metadata is provided in the RPD to support the calculation method adopted by the Oracle Project application that uses Linear Spreading to calculate PTD and ITD metrics.

The "Fact – Project Budget - Linear Spread" logical table contains Budget metrics built using a Linear Spreading logic in the Business Model and Mapping layer. However, the new metrics are not exposed in the Presentation layer. You can either replace the existing Budget metrics in the Presentation layer with the new ones (the same metric names are used in the old and new logical Budget fact tables), or create a new subject area to expose the new metrics. The first option enables you to reuse the existing reports with Budget metrics without any modification.

15.2 Configuration Required Before a Full Load of Project Analytics

This section contains configuration steps that you need to perform on Oracle Project Analytics before you do a full data load. It contains the following topics:

- Section 15.2.1, "Configuration Steps for Project Analytics for All Source Systems"
- Section 15.2.2, "Configuration Steps for Project Analytics for Oracle EBS"
- Section 15.2.3, "Configuration Steps for Project Analytics for PeopleSoft"
- Section 15.2.4, "Configuration Steps for Project Analytics for Universal"

15.2.1 Configuration Steps for Project Analytics for All Source Systems

This section contains configuration steps before you do a full load that apply to all source systems. It contains the following topics:

- Section 15.2.1.1, "Configuring Dimension Hierarchies for Project Analytics"
- Section 15.2.1.2, "Configuring DAC Parameters for Project Analytics"
- Section 15.2.1.3, "Configuring the Project Cost Aggregate Table"

Note: For configuration steps that apply to all Oracle BI Application modules, see Chapter 3, "Configuring Common Areas and Dimensions."

15.2.1.1 Configuring Dimension Hierarchies for Project Analytics

Out of the box, Oracle BI Applications supports the following hierarchies between the Project Analytics dimensions:

- Project to Task
- Project Organization to Project Manager to Project
- Expenditure Class to Expenditure Category

You can set any type of hierarchy drill-downs between dimensions in the Oracle BI Applications analytics repository (RPD).

To configure dimension hierarchies:

1. Select the dimension that you want to drill down from.
2. Go to the Hierarchy in the RPD that corresponds to this dimension.
3. Go to the Detail Level of this hierarchy and double-click it.
4. Go to the Preferred Drill Path tab.
5. Click the Add button.
6. In the new window, choose the hierarchy and level from the Core Business Model to which you want a user to drill down when a user clicks a record from the current dimension.

Note: The Resource Breakdown Structure (RBS) available in Oracle EBS has not been added to the data warehouse because of the flexibility of setting hierarchies and drill down between dimensions in the RPD.

15.2.1.2 Configuring DAC Parameters for Project Analytics

Table 15–1 lists the metadata in DAC that pertains to Oracle Project Analytics for Oracle EBS and PeopleSoft.

Table 15–1 DAC Metadata for Oracle Project Analytics for Oracle EBS and PeopleSoft

DAC Metadata Type	Metadata Name
Supported Container	■ Oracle 11.5.10 ■ Oracle R12 ■ PeopleSoft 8.9 ■ PeopleSoft 9.0 ■ Universal
Subject Areas	Projects
Task Groups	■ TASK_GROUP_Extract_Project_Revenue ■ TASK_GROUP_Load_ProjectCostFacts

Table 15–1 (Cont.) DAC Metadata for Oracle Project Analytics for Oracle EBS and

DAC Metadata Type	Metadata Name
Execution Plans	▪ Projects - Oracle 11.5.10 ▪ Projects - Oracle R12 ▪ Projects - PeopleSoft 8.9 ▪ Projects - PeopleSoft 9.0 ▪ Projects - Universal
Configuration Tag	▪ Project Dimensions ▪ Multiple Calendar Support

Note that in DAC, the two configuration tags, Project Dimensions and Multiple Calendar Support, are specific to Project Analytics. They are used to bring in all the correct tasks for populating the Project Analytics tables in the data warehouse.

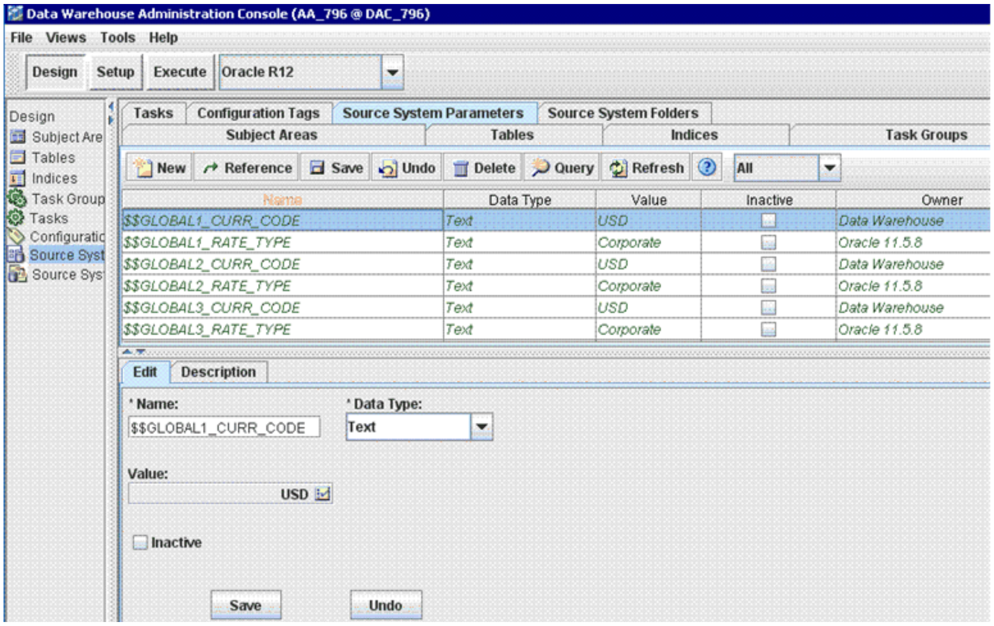
Before you run the initial ETL run, you need to configure the COST_TIME_GRAIN parameter for the time aggregation level in the Project Cost Lines aggregate fact table.

To configure the COST_TIME_GRAIN parameter:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab.
3. Select the Cost_Time_Grain parameter and then in the Edit tab, select the appropriate value in the Value field. Valid values include PERIOD, QUARTER, or YEAR.

To configure the Global Currency parameters:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Source System Parameters tab.
3. Select the global currency code parameters and modify them as appropriate.


Data Warehouse Administration Console (AA_796 @ DAC_796)
File Views Tools Help
Design Setup Execute
Oracle R12
Design
Subject Are
Tables
Indices
Task Group
Tasks
Configuratic
Source Syst
Source Sys
Tasks Configuration Tags Source System Parameters Source System Folders
Subject Areas Tables Indices Task Groups
New Reference Save Undo Delete Query Refresh
All
Edit Description
Save Undo

Name	Data Type	Value	Inactive	Owner
$$GLOBAL1_CURR_CODE	Text	USD	□	Data Warehouse
$$GLOBAL1_RATE_TYPE	Text	Corporate	□	Oracle 11.5.8
$$GLOBAL2_CURR_CODE	Text	USD	□	Data Warehouse
$$GLOBAL2_RATE_TYPE	Text	Corporate	□	Oracle 11.5.8
$$GLOBAL3_CURR_CODE	Text	USD	□	Data Warehouse
$$GLOBAL3_RATE_TYPE	Text	Corporate	□	Oracle 11.5.8

* Name: $$GLOBAL1_CURR_CODE

* Data Type: Text

Value: USD

□ Inactive

15.2.1.3 Configuring the Project Cost Aggregate Table

The Project Cost aggregate table (W_PROJ_COST_A) is used to capture information about the project cost distributions for the expenditure items. You need to configure the Project Cost Lines aggregate table before the initial ETL run and subsequent incremental ETL.

Before the initial ETL run, you need to configure the COST_TIME_GRAIN parameter in DAC (refer to the section above for more information) for the time aggregation level in the Project Cost Lines aggregate fact table.

By default, the COST_TIME_GRAIN parameter has a value of QUARTER. The possible values for the COST_TIME_GRAIN parameter are:

- PERIOD
- QUARTER
- YEAR

The Project Cost Lines aggregate table is fully loaded from the base table in the initial ETL run. The table can grow to millions of records. Therefore, the Project Cost aggregate table is not fully reloaded from the base table after each incremental ETL run. The Oracle Business Analytics Warehouse minimizes the incremental aggregation effort by modifying the aggregate table incrementally as the base table is updated. The process is as follows:

1. Oracle Business Analytics Warehouse finds the records to be updated in the base table since the last ETL run, and loads them into the W_PROJ_COST_LINE_TMP table. The measures in these records are multiplied by (-1). The mapping responsible for this task is SIL_ProjectCostLinesFact_Derive_PreLoadImage.
2. Oracle Business Analytics Warehouse finds the inserted or updated records in the base table since the last ETL run, and loads them into the W_PROJ_COST_LINE_TMP table, without changing their sign. The mapping responsible for this task is SIL_ProjectCostLinesFact_Derive_PreLoadImage, which is run before PLP_ProjectCostLinesFact_Derive_PostLoadImage updates or inserts records in the base table.
3. Oracle Business Analytics Warehouse aggregates the W_PROJ_COST_LINE_TMP table and load to W_PROJ_COST_A_TMP, which has the same granularity as the W_PROJ_COST_A table.
4. The PLP_ProjectCostLinesAggregate_Derive mapping looks up the W_PROJ_COST_A aggregate table to update existing buckets or insert new buckets in the aggregate table (the mapping is PLP_ProjectCostLinesAggregate_Load).

For instructions on how to update the RPD metadata after changing the aggregation period, see technical note 1088171.1 on My Oracle Support.

15.2.2 Configuration Steps for Project Analytics for Oracle EBS

This section contains configuration steps before you do a full data load that apply to Oracle EBS. It contains the following topics:

- Section 15.2.2.1, "About Configuring Domain Values and CSV Worksheet Files for Oracle Project Analytics"
- Section 15.2.2.2, "How to Configure the DomainValues_Project_Type_Class_code_oraVersion.csv"
- Section 15.2.2.3, "How to Configure the domainValues_Project_Status_oraVersion.csv"

- Section 15.2.2.4, "How to Configure the domainValues_Project_ServiceType_oraVersion.csv"
- Section 15.2.2.5, "How to Configure the domainValues_Project_ResourceClass_oraVersion.csv"
- Section 15.2.2.6, "How to Configure the domainValues_Project_InvoiceRevenue_Status_oraVersion.csv"
- Section 15.2.2.7, "How to Configure the domainValues_Project_Invoice_Line_Type_oraVersion.csv"
- Section 15.2.2.8, "How to Configure the domainValues_Project_Funding_Level_code_oraVersion.csv"
- Section 15.2.2.9, "How to Configure the domainValues_Project_Funding_Category_oraVersion.csv"
- Section 15.2.2.10, "How to Configure the domainValues_Project_Cost_Transfer_Status_oraVersion.csv"
- Section 15.2.2.11, "How to Configure the domainValues_Project_Budget_Time_Phased_Code_oraVersion.csv"
- Section 15.2.2.12, "About the Project Commitments Subject Area for Oracle EBS"
- Section 15.2.2.13, "How to Configure the domainValues_Project_CommitmentType_oraVersion.csv"
- Section 15.2.2.14, "About Configuring the Project Dimension for Oracle EBS"
- Section 15.2.2.15, "About Configuring the Task Dimension for Oracle EBS"
- Section 15.2.2.16, "About Configuring the Financial Resource Dimension"
- Section 15.2.2.17, "About Configuring the Project Classification Dimension"
- Section 15.2.2.18, "About Configuring the Work Type Dimension"
- Section 15.2.2.19, "About Configuring the Job Dimension"
- Section 15.2.2.20, "About Operating Unit Based Security with Oracle EBS"
- Section 15.2.2.21, "Configuring a Custom Relationship Value for Project Customer"
- Section 15.2.2.22, "About Configuring Budget Fact for Oracle EBS"
- Section 15.2.2.23, "About Configuring Forecast Fact for Oracle EBS"
- Section 15.2.2.24, "About Configuring Funding Fact for Oracle EBS"
- Section 15.2.2.25, "About Configuring Cost Fact for Oracle EBS"
- Section 15.2.2.26, "Configuring Revenue Fact for Oracle EBS"
- Section 15.2.2.27, "Configuring Invoice Fact for Oracle EBS"
- Section 15.2.2.28, "How to Tune Performance for Project Analytics for Oracle EBS"

15.2.2.1 About Configuring Domain Values and CSV Worksheet Files for Oracle Project Analytics

You configure Oracle Project Analytics by mapping domain values to columns in the CSV files located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

For more information about configuring domain values with CSV worksheet files, see Chapter 15.2.2.1, "About Configuring Domain Values and CSV Worksheet Files for Oracle Project Analytics."

For more information about domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

> **Note:** Incorrect mappings result in inaccurate calculations of Oracle Business Intelligence metrics.

Table 15–2 lists the CSV worksheet files and the domain values for Oracle Project Analytics in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

Table 15–2 CSV Worksheet Files and Domain Values for Oracle Project Analytics

Domain Values	Domain Value Table Column	Description	Session
domainValues_Project_Type_Class_code_ora*Version*.csv	W_PROJECT_D.W_PROJECT_TYPE_CLASS_CODE	Lists the Project Type Class codes and their corresponding domain values of Project Type Class Code for the Oracle 11i/12 Application	SDE_ORA_Project
domainValues_Project_Status_ora*Version*.csv	W_STATUS_D.W_STATUS_CODE	Lists the Project Status codes and their corresponding domain values of Project Status Code for the Oracle 11i/12 Application	SDE_ORA_StatusDimension_ProjectStatus
domainValues_Project_ServiceType_ora*Version*.csv	W_XACT_TYPE_D.W_XACT_TYPE_CODE	Lists the Service Type codes and their corresponding domain values for the Oracle 11i/12 Application	SDE_ORA_TransactionTypeDimension_ServiceType
domainValues_Project_Security_Level_code_ora*Version*.csv	W_PROJECT_D.W_SECURITY_LEVEL	Lists the Project Security Level codes and their corresponding domain values for the Oracle 11i/12 Application	SDE_ORA_Project
domainValues_Project_ResourceClass_ora*Version*.csv	W_XACT_TYPE_D.W_XACT_TYPE_CODE	Lists the Resource Class codes and their corresponding domain values for the Oracle 11i/12 Application	SDE_ORA_TransactionTypeDimension_ResourceClass
domainValues_Project_InvoiceRevenue_Status_ora*Version*.csv	W_STATUS_D.W_STATUS_CODE	Lists the Invoice/Revenue Status codes and their corresponding domain values for the Oracle 11i/12 Application	SDE_ORA_StatusDimension_Project_InvoiceStatus, SDE_ORA_StatusDimension_Project_RevenueStatus
domainValues_Project_Invoice_Line_Type_*Version*.csv	W_XACT_TYPE_D.W_XACT_TYPE_CODE	Lists the Invoice Line Type codes and their corresponding domain values for the Oracle 11i/12 Application	SDE_ORA_TransactionTypeDimension_Invoice_Line_Type
domainValues_Project_Funding_Level_code_ora*Version*.csv	W_PROJECT_D.W_PROJECT_FUNDING_LEVEL_CODE	Lists the Project Funding Level codes and their corresponding domain values for the Oracle 11i/12 Application	SDE_ORA_Project

Table 15–2 (Cont.) CSV Worksheet Files and Domain Values for Oracle Project Analytics

Domain Values	Domain Value Table Column	Description	Session
domainValues_Project_Funding_Category_ora*Version*.csv	W_XACT_TYPE_D.W_XACT_TYPE_CODE	Lists the Funding Category codes and their corresponding domain values for the Oracle 11i/12 Application	SDE_ORA_TransactionTypeDimension_FundingCategory
domainValues_Project_Cost_Transfer_Status_ora*Version*.csv	W_STATUS_D.W_STATUS_CODE	Lists the Cost Transfer Status codes and their corresponding domain values for the Oracle 11i/12 Application	SDE_ORA_StatusDimension_Project_TransferStatus
domainValues_Project_Budget_Time_Phased_Code_ora*Version*.csv	W_BUDGET_D.W_TIME_PHASED_CODE	Lists the Budget Time Phased codes and their corresponding domain values for the Oracle 11i/12 Application	SDE_ORA_BudgetDimension_BudgetType, SDE_ORA_BudgetDimension_FinPlanType

15.2.2.2 How to Configure the DomainValues_Project_Type_Class_code_oraVersion.csv

This section explains how to configure the domainValues_Project_Type_Class_code_ora*Version*.csv.

1. Identify the Project Types in your Oracle EBS source system by using the following SQL:

   ```
   SELECT DISTINCT
   PROJECT_TYPE_CLASS_CODE
   FROM PA_PROJECT_TYPES_ALL
   ```

2. Using a text editor, open the domainValues_Project_Type_Class_code_ora*Version*.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the PROJECT_TYPE_CLASS_CODE to the PROJECT_TYPE_CLASS_CODE column in the file. The data must be copied starting from the 6th line. Use commas to separate the entries.
4. Map each Project Type Class Code (PROJECT_TYPE_CLASS_CODE) to one domain value. For more information about Project Type domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

15.2.2.3 How to Configure the domainValues_Project_Status_oraVersion.csv

This section explains how to configure the domainValues_Project_Status_ora*Version*.csv.

1. Identify the Project Statuses in your Oracle EBS source system by using the following SQL:

   ```
   SELECT PROJECT_STATUS_CODE, PROJECT_STATUS_NAME
   FROM PA_PROJECT_STATUSES WHERE STATUS_TYPE = 'PROJECT'
   ```

2. Using a text editor, open the domainValues_Project_Status_ora*Version*.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

3. Copy the PROJECT_ STATUS _CODE and PROJECT_ STATUS NAME to the STATUS_CODE and STATUS_NAME columns in the file respectively. The data must be copied starting from the 6th line. Use commas to separate the entries.
4. Map each Project Status Code (PROJECT_STATUS_CODE) to one domain value. For more information about Project Status domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

15.2.2.4 How to Configure the domainValues_Project_ServiceType_oraVersion.csv

This section explains how to configure the domainValues_Project_ServiceType_ ora*Version*.csv.

1. Identify the Project Service Types in your Oracle EBS source system by using the following SQL:

   ```
   SELECT LOOKUP_CODE FROM FND_LOOKUP_VALUES WHERE LOOKUP_TYPE = 'SERVICE TYPE'
   ```

2. Using a text editor, open the domainValues_Project_ServiceType_ora*Version*.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_ HOME*\server\infa_shared\LkpFiles).
3. Copy the LOOKUP_CODE to the XACT_TYPE_CODE column in the file. The data must be copied starting from the 6th line. Use commas to separate the entries.
4. Map each Lookup Code (XACT_TYPE_CODE) to one domain value. For more information about Project Service domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

15.2.2.5 How to Configure the domainValues_Project_ResourceClass_ oraVersion.csv

This section explains how to configure the domainValues_Project_ResourceClass_ ora*Version*.csv.

1. Identify the Project Resource Classes in your Oracle EBS source system by using the following SQL:

   ```
   SELECT PB.RESOURCE_CLASS_CODE,PT.NAME,PT.DESCRIPTION FROM PA_RESOURCE_CLASSES_B
   PB,PA_RESOURCE_CLASSES_TL PT
   WHERE PB.RESOURCE_CLASS_ID = PT.RESOURCE_CLASS_ID
   AND PT.LANGUAGE = 'US'
   ```

2. Using a text editor, open the domainValues_Project_ResourceClass_ora*Version*.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_ HOME*\server\infa_shared\LkpFiles).
3. Copy the RESOURCE_CLASS_CODE, NAME, and DESCRIPTION to the XACT_ TYPE_CODE, XACT_TYPE_NAME, XACT_TYPE_DESC columns in the file respectively. The data must be copied starting from the 6th line. Use commas to separate the entries.
4. Map each Resource Class Code (XACT_TYPE_CODE) to one domain value. For more information about Resource Class domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

15.2.2.6 How to Configure the domainValues_Project_InvoiceRevenue_Status_oraVersion.csv

This section explains how to configure the domainValues_Project_InvoiceRevenue_Status_ora*Version*.csv.

1. Identify the Invoice/Revenue Statuses in your Oracle EBS source system by using the following SQL:

   ```
   SELECT LOOKUP_CODE,MEANING FROM FND_LOOKUP_VALUES WHERE LOOKUP_TYPE =
   'INVOICE/REVENUE STATUS' AND LANGUAGE = 'US'
   ```

2. Open the domainValues_Project_InvoiceRevenue_Status_ora*Version*.csv file using a text editor in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy LOOKUP_CODE and MEANING to the STATUS_CODE and STATUS_NAME columns in the file respectively. The data must be copied starting from the 6th line. Use commas to separate the entries.
4. Map each STATUS_CODE to one domain value. For more information about Invoice Revenue domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

15.2.2.7 How to Configure the domainValues_Project_Invoice_Line_Type_oraVersion.csv

This section explains how to configure the domainValues_Project_ Invoice_Line_Type _ora*Version*.csv.

1. Identify the Invoice Line Types in your Oracle EBS source system by using the following SQL:

   ```
   SELECT LOOKUP_CODE,MEANING,DESCRIPTION FROM FND_LOOKUP_VALUES WHERE LOOKUP_TYPE
   LIKE 'INVOICE LINE TYPE'
   AND LANGUAGE = 'US'
   ```

2. Using a text editor, open the domainValues_Project_Invoice_Line_Type_ora*Version*.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy LOOKUP_CODE, MEANING, and DESCRIPTION to the XACT_TYPE_CODE, XACT_TYPE_NAME, and XACT_TYPE_DESC columns in the file respectively. The data must be copied starting from the 6th line. Use commas to separate the entries.
4. Map each XACT_TYPE_CODE to one domain value. For more information about Invoice Line Type domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

15.2.2.8 How to Configure the domainValues_Project_Funding_Level_code_oraVersion.csv

This section explains how to configure the domainValues_Project_Funding_Level_code_ora*Version*.csv.

1. Identify the Funding Levels in your Oracle EBS source system by using the following SQL:

```
SELECT DISTINCT PROJECT_LEVEL_FUNDING_FLAG FROM PA_PROJECTS_ALL
```

2. Using a text editor, open the domainValues_Project_Funding_Level_code_ora*Version*.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the PROJECT_FUNDING_LEVEL_FLAG to the PROJECT_FUNDING_LEVEL_CODE column in the file. The data must be copied starting from the 6th line. Use commas to separate the entries.
4. Map each PROJECT_FUNDING_LEVEL_CODE to one domain value. For more information about Project Funding domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

15.2.2.9 How to Configure the domainValues_Project_Funding_Category_oraVersion.csv

This section explains how to configure the domainValues_Project_Funding_Category_ora*Version*.csv.

1. Identify the Funding Category Types in your Oracle EBS source system by using the following SQL:

```
SELECT LOOKUP_CODE,MEANING,DESCRIPTION FROM FND_LOOKUP_VALUES
WHERE LOOKUP_TYPE = 'FUNDING CATEGORY TYPE'
AND LANGUAGE = 'US'
```

2. Using a text editor, open the domainValues_Project_Funding_Category_ora*Version*.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the LOOKUP_CODE, MEANING, and DESCRIPTION to the XACT_TYPE_CODE, XACT_TYPE_NAME, and XACT_TYPE_DESC columns in the file respectively. The data must be copied starting from the 6th line. Use commas to separate the entries.
4. Map each XACT_TYPE_CODE to one domain value. For more information about Project Funding domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

15.2.2.10 How to Configure the domainValues_Project_Cost_Transfer_Status_oraVersion.csv

This section explains how to configure the domainValues_Project_Cost_Transfer_Status_ora*Version*.csv.

1. Identify the Project Transfer Statuses in your Oracle EBS source system by using the following SQL:

```
SELECT LOOKUP_CODE,MEANING FROM FND_LOOKUP_VALUES WHERE LOOKUP_TYPE = 'TRANSFER
STATUS' AND LANGUAGE = 'US'
```

2. Using a text editor, open the domainValues_Project_Cost_Transfer_Status_ora*Version*.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

3. Copy the LOOKUP_CODE and MEANING to the STATUS_CODE and STATUS_NAME columns in the file respectively. The data must be copied starting from the 6th line. Use commas to separate the entries.
4. Map each STATUS_CODE to one domain value. For more information about Status Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

15.2.2.11 How to Configure the domainValues_Project_Budget_Time_Phased_Code_oraVersion.csv

This section explains how to configure the domainValues_Project_Budget_Time_Phased_Code_ora*Version*.csv.

1. Identify the Budget Time Phase Codes in your Oracle EBS source system by using the following SQL:

   ```
   SELECT LOOKUP_CODE FROM FND_LOOKUP_VALUES WHERE LOOKUP_TYPE = 'BUDGET TIME
   PHASED TYPE' AND LANGUAGE = 'US'
   ```

2. Using a text editor, open the domainValues_Project_Budget_Time_Phased_Code_ora*Version*.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the LOOKUP_CODE to the TIME_PHASED_CODE column in the file. The data must be copied starting from the 6th line. Use commas to separate the entries.
4. Map each TIME_PHASED_CODE to one domain value.
5. For more information about Budget Time Phase Codes, see *Oracle Business Analytics Warehouse Data Model Reference*.
6. Save and close the file.

15.2.2.12 About the Project Commitments Subject Area for Oracle EBS

Oracle Project Analytics for Oracle EBS includes a Project Commitments subject area that provides the ability to report on project commitments. Project commitments include total raw and burdened amounts for requisitions, purchase orders, and supplier invoices for organizations, projects, tasks, resources, suppliers, and associated hierarchies. The subject area provides the ability to track commitments at the commitment document level.

The Oracle Business Analytics Warehouse includes a star schema to support the Project Commitments subject area. This star contains metrics to report on total commitments and its components, which includes quantity and amount (raw and burdened) for requisitions, purchase orders, and supplier invoices.

The W_PROJ_COMMITMENT_F fact table at the center of the star schema stores the latest commitment data, sourced from the transactional source PA_COMMITMENT_TXNS.

15.2.2.13 How to Configure the domainValues_Project_CommitmentType_oraVersion.csv

To use the project commitments subject area, you need to configure the domainValues_Project_CommitmentType_ora*Version*.csv file with the commitment types that your organization uses in the Oracle EBS implementation.

Table 15–3 describes the CSV worksheet file and domain values for the Project Commitments subject area.

Table 15–3 CSV Worksheet File and Domain Values for the Project Commitments Subject Area

Domain Values	Domain Value Table Column	Description	Session
domainValues_ Project_ CommitmentType_ ora*Version*.csv	W_PROJ_ COMMITMENT_F.W_ COMMITMENT_TYPE	Lists the Commitment Type codes and their corresponding domain values of Commitment Type for the Oracle 11i/12 Application	SDE_ORA_ ProjectCommitmentFact

To configure the domainValues_Project_CommitmentType_ora*Version*.csv file:

1. Identify the Project Commitment Types in your Oracle EBS source system by using the following SQL:

   ```
   SELECT LOOKUP_CODE FROM FND_LOOKUP_VALUES WHERE LOOKUP_TYPE = 'COMMITMENT LINE
   TYPE'
   ```

2. Using a text editor, open the domainValues_Project_CommitmentType_ ora*Version*.csv file located in the $PMServer\LkpFiles folder.
3. Copy the LOOKUP_CODE to the COMMITMENT_TYPE column in the file. The data must be copied starting from the 6th line. Use commas to separate the entries.
4. Map each Lookup Code (COMMITMENT_TYPE) to one domain value and a corresponding meaning. For more information about Project Commitment Type domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

15.2.2.14 About Configuring the Project Dimension for Oracle EBS

The Project dimension (W_PROJECT_D) data is sourced from the PA_PROJECTS_ALL table in Oracle EBS. Other important project attributes such as Project Type, Project Manager, Country, State, City, Project Organization, Project Status, Project Operating Unit ID, Project Primary Customer ID, and Project Primary Customer Account ID are sourced from the following Oracle EBS tables:

- PA_LOCATIONS
- HR_ORGANIZATIONS_ALL
- PA_PROJECT_TYPES
- PA_PROJECT_STATUSES
- PA_IMPLEMENTATIONS_ALL

Out of the box, Oracle BI Applications supports Projects and not Project Templates. The following filter prevents Project Template data from appearing:

```
PA_PROJECTS_ALL.TEMPLATE_FLAG = N
```

If you need to source Project Template information, you can remove this filter condition. However, Project Templates cannot be used for Billing, Costing, or Revenue.

All analysis in pre-built reports is based on Project Operating Unit, which is the operating unit of a project.

The Project dimension is also referenced in some Financial Analytics and Supply Chain Analytics facts. For more information about the integration with other Oracle BI Analytics solutions, Section 15.4.1, "Integrations with Project Analytics for Oracle EBS."

15.2.2.15 About Configuring the Task Dimension for Oracle EBS

Task dimension data is sourced from the task table (PA_TASKS) in Oracle EBS, as well as from other task-related OLTP tables such as:

- PA_PROJ_ELEMENTS
- PA_PROJ_ELEMENT_VERSIONS
- PA_PROJ_ELEM_VER_STRUCTURE
- PA_PROJ_ELEM_VER_SCHEDULE

Attributes such as WBS_NUMBER, PRIORITY_CODE, SCHEDULE_START_DATE, and SCHEDULE_END_DATE are sourced from these tables.

Oracle BI Applications support only the latest version of the Financial Structure by using the following filter conditions:

```
PA_PROJ_ELEM_VER_STRUCTURE.STATUS_CODE = 'STRUCTURE_PUBLISHED'
AND PA_PROJ_ELEM_VER_STRUCTURE.LATEST_EFF_PUBLISHED_FLAG = 'Y'
```

The W_TASK_DH hierarchy table stores the flattened hierarchy for every task in W_TASK_D. It is at the same grain as W_TASK_D and is modeled as a Type I dimension. All tasks in the hierarchy support these columns:

- TASK_NAME
- TASK_NUMBER
- WBS_LEVEL
- WBS_NUMBER

Because both tables, W_TASK_D and W_TASK_DH, are at the same grain, fact tables do not have a separate foreign key to join with this table; instead, the join is on the Task Foreign Key.

Out of the box, Oracle BI Applications support 20 levels in the flattened hierarchy. The levels are Base, 1, 2, and so forth up to 18, and Top. The base level represents the hierarchy record, and Top level is the Top hierarchy under the Project. If your financial structure contains more than 20 levels, you can extend the number of levels in the schema and ETL to support all levels.

To extend levels, you need to add all the change capture columns (TASK_NAME, TASK_NUMBER, WBS_LEVEL and WBS_NUMBER) for every new level that you want in the W_TASK_DH table. In the SILOS folder, update the following mappings for the ETL:

- SIL_Project_TaskDimensionHierarchy
- SIL_Project_TaskDimensionHierarchy_Full
- SIL_Project_TaskDimensionHierarchy_IdentifyBaseModified
- SIL_Project_TaskDimensionHierarchy_IdentifyModifiedRecords

In addition, you need to update the DAC schema by importing the new columns from the physical schema. You must update the following objects in the metadata repository:

- W_TASK_DH table in the physical layer

- Dim - Task Hierarchy Logical Table and Task Hierarchy Dimension in the logical layer
- All the Task Hierarchy Presentation tables in the Presentation Area

15.2.2.16 About Configuring the Financial Resource Dimension

Revenue and Cost are often analyzed side by side in the same report. The data for Revenue is typically based on Expenditure Type and Event Type. The data for Cost is typically based on Expenditure Type. Therefore, both Expenditure Types and Event Types are stored in the same dimension.

The same set of columns are overloaded so that all the Revenue data by Expenditure and Event types can be seen in the same report.

The Expenditure Type data is sourced from the PA_EXPENDITURE_TYPES table. The EXPENDITURE_TYPE, EXPENDITURE_CATEGORY_CODE, and REVENUE_CATEGORY_CODE columns in the OLTP populate the RESOURCE_TYPE_CODE, EXPENDITURE_CATEGORY_CODE and REVENUE_CATEGORY_CODE columns respectively in the dimension table.

The Event Type data is sourced from the PA_EVENT_TYPES table. The EVENT_TYPE, EVENT_TYPE_CLASSIFICATION, and REVENUE_CATEGORY_CODE columns in the OLTP populate the RESOURCE_TYPE_CODE, EVENT_TYPE_CLASS_CODE, and REVENUE_CATEGORY_CODE columns respectively in the dimension table.

15.2.2.17 About Configuring the Project Classification Dimension

Every project can be optionally classified into different categories. Within these categories, a project can be further categorized into different classification codes. Depending on how these classification categories are defined in the application, for some categories, a project can be classified with more than one classification code.

The Project Classification Table (W_PROJ_CLASSIFICATION_D) is at the grain of Project, Classification Category and Classification Code. The Project facts do not have an explicit foreign key for joining with Project Classification Dimension; instead the join is on the Project Foreign Key. As specifying a Classification Category is optional for a project, so the logical join in the metadata repository between the Facts and Project Classification Dimension has been set as right outer join to avoid losing records in case the project has not been classified.

> **Note:** A particular classification code might exist for more than one classification category. Therefore, to avoid double counting, it is important that a classification category is fixed in a report that has classification code as one of the reporting attributes. If a Project belongs to more than one Classification Category under the same Classification, the Project metrics (Cost, Revenue, and so forth) will be double counted.

15.2.2.18 About Configuring the Work Type Dimension

The W_WORK_TYPE_D dimension stores the implementation defined work types to be performed on various projects. For ORACLE EBS sources, the "work type id" can be defined at the Project level, Task level, or Expenditure Item level. While populating the Cost & Revenue Facts, the work type id is defined at the Task level if it is null at the Expenditure Item level.

15.2.2.19 About Configuring the Job Dimension

The Job dimension is maintained in the Human Resources Analytics module.

For information about how to configure this dimension, see Chapter 7.2.3.2, "How to Configure the Key Flexfields for the Job Dimension."

15.2.2.20 About Operating Unit Based Security with Oracle EBS

All fact tables in Project Analytics are secured by Operating Unit (or Business Unit) security. This security filter uses the OU_ORG session variable, which is initialized by the list of Operating Units the user has access to in the OLTP system. An additional join is added to each fact report, so only transaction data for Projects that are owned by the Project Operating Units that the user is allowed to access are shown in the report. Dimension reports are not secured by filters.

For more information about adding security to dimensions and implementation steps for operating unit based security with Oracle EBS, see *Oracle Business Intelligence Applications Security Guide*.

15.2.2.21 Configuring a Custom Relationship Value for Project Customer

By default, Oracle EBS only has the 'PRIMARY' relationship code in the PA_PROJECT_CUSTOMERS table. Therefore, the value is included in the Informatica lookup used in the source extract mapping for the Project dimension to get the customer for a project. Customers can define an additional value such as 'OVERRIDE CUSTOMER' as the relationship value. In this case, the lookup must be edited to include any additional values.

To edit the lookup:

1. Open the PowerCenter Designer and connect to your Informatica repository where you want to effect these changes.
2. Open the SDE_ORA_11510_Adaptor or SDE_ORA_R12_Adaptor folder of the Informatica repository.
3. Open the mapping SDE_ORA_ProjectDimension and open the mplt_SA_ORA_ProjectDimension mapplet in it.
4. Check out the mplt_SA_ORA_ProjectDimension mapplet.
5. Open the lookup LKP_PROJ_CUST, select the Properties tab, and then open the SQL query.
6. Remove the existing SQL and add the following sample SQL where it is assumed the values are 'PRIMARY' and 'OVERRIDE CUSTOMER'. Modify it according to your configuration. If you want it to be independent of any relationships, then just remove the filters on PROJECT_RELATIONSHIP_CODE.

```
SELECT MAX(PA_PROJECT_CUSTOMERS.CUSTOMER_ID) AS CUSTOMER_ID,
PA_PROJECT_CUSTOMERS.PROJECT_ID AS PROJECT_ID
FROM PA_PROJECT_CUSTOMERS
WHERE UPPER(PA_PROJECT_CUSTOMERS.PROJECT_RELATIONSHIP_CODE) in
 ('PRIMARY','OVERRIDE CUSTOMER')
AND PA_PROJECT_CUSTOMERS.CUSTOMER_BILL_SPLIT =
  (SELECT MAX(CUSTOMER_BILL_SPLIT)
  FROM PA_PROJECT_CUSTOMERS A
  WHERE A.PROJECT_ID = PA_PROJECT_CUSTOMERS.PROJECT_ID
  and UPPER(a.PROJECT_RELATIONSHIP_CODE) in ('PRIMARY','OVERRIDE CUSTOMER')
  )
GROUP BY PA_PROJECT_CUSTOMERS.PROJECT_ID
```

7. Validate the lookup SQL and press OK twice, and then save the repository and check in the mapplet. Review the mapping to ensure it is valid.
8. Open PowerCenter Workflow Manager and connect to one of these repository folders: SDE_ORA_11510_Adaptor or SDE_ORA_R12_Adaptor.
   a. Check out the SDE_ORA_Project session in the Task Developer tab.
   b. Validate, save, and check in the session.
   c. Save your repository.

15.2.2.22 About Configuring Budget Fact for Oracle EBS

Baselined Budgets are extracted into the Budget Fact (W_PROJ_BUDGET_F) table, the grain of which is Budget Line. Because only the baselined budgets are extracted, the records in this table are not updated after they are loaded into the data warehouse; only new records are inserted during the incremental ETL run. Budgets are stored in the Budget dimension (W_BUDGET_D).

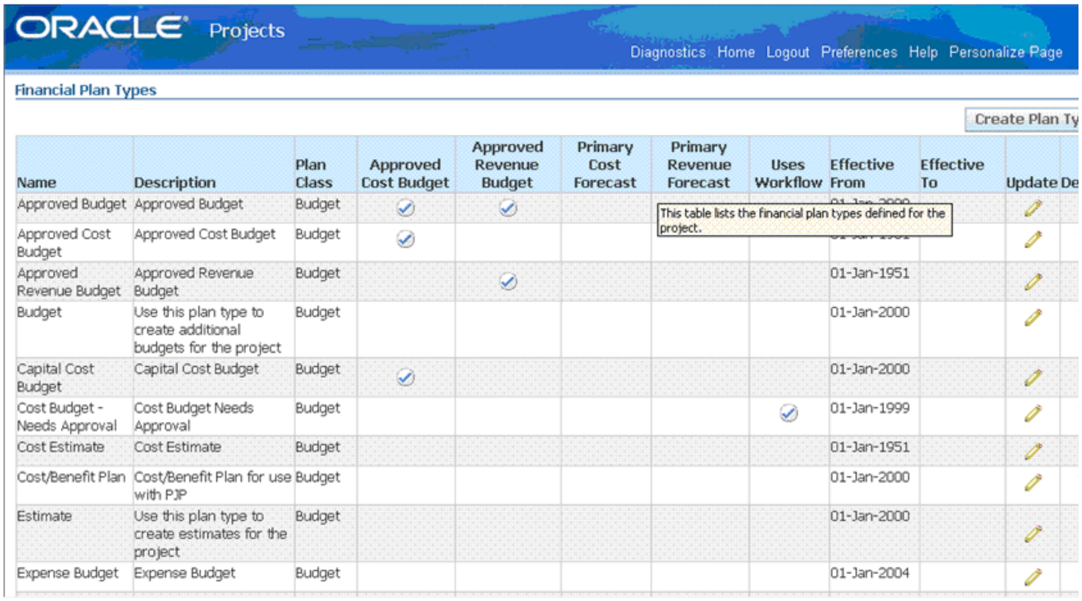
> **Note:** For Oracle EBS, Transaction Currency is the Document Currency for this fact.

Defining Filters on Budget Metrics A user can create multiple budgets for a single Project and multiple versions for the same budget type. Therefore, all exposed metrics are filtered by the following filters:

- **Approved Budget Type.** A project can have only one Cost Budget with a budget type as "Approved Cost Budget" and one Revenue Budget with a budget type as "Approved Revenue Budget." Therefore, all Cost Budget metrics are filtered by the Approved Cost Budget and Approved Revenue Budget flags to ensure that the metrics include data from one budget only.
- **Current or Original Budget.** Each Project forecast can have multiple versions. The Current Version may not be the same as the Original version. Therefore, to show only one Forecast version at a time, there are separate metrics for Current version and Original version. These flags are set automatically in OLTP when the forecast is baselined, but users can update them manually.

The user can still see the metrics for any other budget type or version by bringing the non-filtered metrics from Fact - Project Budget fact table into the Presentation area. But to avoid duplicate data, the report must have a filter on "Dim - Budget.Budget Type" and "Dim - Budget.Budget Version".

Before running the ETL for the first time, go to the Financial Plan Type page in the HTML application and set your Approved Cost Budget Type and your Approved Revenue Budget Types. Also, you must select the Approved Cost Budget and Approved Revenue Budget types.



Financial Plan Types

Name	Description	Plan Class	Approved Cost Budget	Approved Revenue Budget	Primary Cost Forecast	Primary Revenue Forecast	Uses Workflow	Effective From	Effective To	Update	De
Approved Budget	Approved Budget	Budget	✓	✓				01-Jan-2000		✎	
Approved Cost Budget	Approved Cost Budget	Budget	✓					01-Jan-1951		✎	
Approved Revenue Budget	Approved Revenue Budget	Budget		✓				01-Jan-1951		✎	
Budget	Use this plan type to create additional budgets for the project	Budget						01-Jan-2000		✎	
Capital Cost Budget	Capital Cost Budget	Budget	✓					01-Jan-2000		✎	
Cost Budget - Needs Approval	Cost Budget Needs Approval	Budget					✓	01-Jan-1999		✎	
Cost Estimate	Cost Estimate	Budget						01-Jan-1951		✎	
Cost/Benefit Plan	Cost/Benefit Plan for use with PJP	Budget						01-Jan-2000		✎	
Estimate	Use this plan type to create estimates for the project	Budget						01-Jan-2000		✎	
Expense Budget	Expense Budget	Budget						01-Jan-2004		✎	

This table lists the financial plan types defined for the project.

Budgets Created in Forms Client For budgets entered through the Form client, the PA_BUDGET_ TYPES.PLAN_TYPE column is not populated for the two predefined budget types, AC and AR. Therefore, the following ETL logic is incorporated in mplt_ BC_ORA_BudgetVersion_BudgetType in the SDE_ORA_Budget Dimension_ BudgetType mapping:

```
IIF(ISNULL(INP_PLAN_
TYPE),DECODE(NAME,'AC','BUDGET','AR','BUDGET','FC','FORECAST','FR',
'FORECAST',INP_PLAN_TYPE) , INP_PLAN_TYPE)
```

Budget Fact Canonical Date The Budget Fact contains the following two sets of Accounting Date and Period WIDs:

- PROJ_ACCT_START_DT_WID, PROJ_ACCT_END_DT_WID, and PROJ_ PERIOD_WID

  PROJ_ACCT_START_DT_WID and PROJ_ACCT_END_DT_WID are populated using START_DATE and END_DATE of budget line only for budgets that are time-phased using the Project Accounting (PA) Calendar.

- GL_ACCT_START_DT_WID, GL_ACCT_END_DT_WID, and GL_PERIOD_WID

  The GL_ACCT_START_DT_WID and GL_ACCT_END_DT_WID are populated using the START_DATE and END_DATE of budget line for budgets that are time-phased by the General Ledger (GL) Calendar.

  For budgets defined with Time Phase equal 'P'(PA), 'N'(No Time Phase) or 'R'(Date Range), the GL_ACCT_START_DT_WID and GL_PERIOD_WID are resolved using the START_DATE of the budget line by choosing the period containing that date in the GL Calendar (pinned by the GL_MCAL_CAL_WID).

  This approach assumes that for time phase 'P','N', and 'R', there is a period containing the START_DATE for the given GL Calendar in the OLTP database.

For Forms -based budgets, even though the application does not allow creating budget lines in a different currency than the Project Functional currency, the currency from Project Functional Currency is used for the default value of the Document Currency field. This enables Budget Amounts to be analyzed in the Global Currencies. For example, Doc Raw Cost Amount is populated as:

```
IIF(ISNULL(INP_DOC_RAW_COST),IIF((TXN_CURRENCY_CODE = PROJFUNC_CURRENCY_CODE),INP_
LOC_RAW_COST,NULL),INP_DOC_RAW_COST)
```

15.2.2.23 About Configuring Forecast Fact for Oracle EBS

The Forecast fact table is based on PA_BUDGET_LINES. A filter is applied to the Budget Version table to extract only baselined Forecasts for the Forecast fact. The grain of this table is a Forecast line. The ETL extracts only baselined forecasts, so the records in this table are not updated after they are loaded to the data warehouse; only new records are inserted during an incremental run. Forecasts are stored in the Budget dimension (W_BUDGET_D) as well.

> **Note:** For Oracle EBS, Transaction Currency is the Document Currency for this fact.

Defining Filters on Forecast Metrics Users can create multiple forecasts for a single Project and multiple versions for the same forecast type. Therefore, Oracle BI Applications filter all exposed metrics using the following filters:

- **Primary Forecast Type.** One project can have only one Cost Forecast with a forecast type of "Primary Cost Forecast" and one Revenue Forecast with a Forecast type of "Primary Revenue Forecast." Therefore, all Cost and Revenue Forecast metrics are filtered on two flags, Primary Cost Forecast and Primary Revenue Forecast, to make sure we are showing data for only one forecast.
- **Current or Original Forecast.** One Project forecast can have multiple versions. To show only one forecast version at a time, every metric for the Current Version and the Current Original Version is shown. These flags are set automatically in OLTP when the forecast is baselined, but users can update them manually.

Users can still view metrics for any other forecast type or version by bringing the non-filtered metrics from the Fact - Project Forecast fact table into the Presentation area. But to avoid duplicate data, the report must have a filter on Dim - Budget.Budget Type and Dim - Budget.Budget Version.

Before running the ETL for the first time, access the Financial Plan Type page in the HTML client, and select your Primary Cost forecast and Primary Revenue forecast types.

Forecasts Created in Forms Client For Forecasts entered through the Form client, the PA_BUDGET_ TYPES.PLAN_TYPE column is not populated for the two predefined budget types, 'FC' and 'FR'. Therefore, the following ETL logic is incorporated in mplt_BC_ORA_BudgetVersion_BudgetType in the SDE_ORA_Budget Dimension_BudgetType mapping:

```
IIF(ISNULL(INP_PLAN_TYPE),DECODE(NAME,'AC','BUDGET','AR',
'BUDGET','FC','FORECAST','FR', 'FORECAST',INP_PLAN_TYPE) , INP_PLAN_TYPE)
```

For 'FC' and 'FR' types of Forecast versions created in the Forms client, the PRIMARY_ COST_FORECAST _FLAG and PRIMARY_REV_FORECAST_FLAG are not populated in PA_BUDGET_VERSIONS. Therefore, the following ETL logic is incorporated in mplt_BC_ORA_BudgetVersion_BudgetType in the SDE_ORA_Budget Dimension_ BudgetType mapping:

```
IIF(ISNULL(INP_PRIMARY_COST_FORECAST_FLAG),IIF(NAME='FC','Y'),INP_PRIMARY_COST_
FORECAST_FLAG)
IIF(ISNULL(INP_PRIMARY_REV_FORECAST_FLAG),IIF(NAME='FR','Y'),INP_PRIMARY_REV_
FORECAST_FLAG)
```

For Forms based forecasts, even though the application does not allow the creation of forecast lines in a different currency than the Project Functional currency, we are defaulting the Project Functional Currency in the Document Currency field, so that the Forecast Amounts can also be analyzed in the Global Currencies. For example Doc EAC Raw Cost Amount is populated as:

```
IIF(ISNULL(INP_DOC_EAC_RAW_COST),IIF((DOC_CURRENCY_CODE = LOC_CURRENCY_CODE),INP_
LOC_EAC_RAW_COST,NULL),INP_DOC_EAC_RAW_COST)
```

Forecast Fact Canonical Date The Forecast fact has the following two sets of Accounting Date and Period WIDs:

- PROJ_ACCT_START_DT_WID, PROJ_ACCT_END_DT_WID & PROJ_PERIOD_WID

  PROJ_ACCT_START_DT_WID and PROJ_ACCT_END_DT_WID are populated using START_DATE and END_DATE of forecast line only for Forecasts that are time phased using the Project Accounting (PA) Calendar.

- GL_ACCT_START_DT_WID, GL_ACCT_END_DT_WID & GL_PERIOD_WID

  The GL_ACCT_START_DT_WID and GL_ACCT_END_DT_WID are populated using START_DATE and END_DATE of forecast line for Forecasts time phased by the General Ledger (GL) Calendar.

  For Forecasts with a Time Phase equal to 'P' (PA), 'N' (No Time Phase), or 'R' (Date Range), the GL_ACCT_START_DT_WID and GL_PERIOD_WID are resolved using the START_DATE of the forecast line by choosing the Period containing that date in the corresponding GL Calendar.

  This approach assumes that for time phase equal 'P', 'N' or 'R', there will always be a period containing the START_DATE for the given GL Calendar in OLTP database.

15.2.2.24 About Configuring Funding Fact for Oracle EBS

Funding is based on Funding Line, which represents allocations made to a project or task. The line level funding information is held in the Funding Line fact (W_PROJ_FUNDING_ LINE_F), which is based on PA_PROJECT_FUNDINGS table in the Billing Module of Oracle EBS.

Also, data is extracted from the Summary Funding table (PA_SUMMARY_PROJECT_FUNDINGS) to retrieve additional metrics like Unbaselined Amount, Baselined Amount, Invoiced Amount, Revenue Accrued; which are not available in the Funding line Fact; these would be available in Funding Header Fact (W_PROJ_FUNDING_HDR_F). Before running any ETL job, you need to run the following process in Oracle EBS to update this table: PRC: Refresh Project Summary Amounts.

> **Note:** For Oracle EBS, Funding Currency is the Document Currency for this fact.

You need to use the following CSV files to configure the Project Funding domain values:

- **domainValues_Project_Funding_Category_ora*Version*.csv.** Used for categorizing funding allocation types. If you change the domain values, you must also change corresponding metric definitions in the RPD. For more information about how to configure this file, see Section 15.2.2.9, "How to Configure the domainValues_Project_Funding_Category_oraVersion.csv."

- **domainValues_Project_Funding_Level_code_ora*Version*.csv.** This flat file is used to indicate whether a funding line is for a Task or a Project. It is not used in any out of the box metric definition. For information about how to configure this file, see Section 15.2.2.8, "How to Configure the domainValues_Project_Funding_Level_code_oraVersion.csv."

Funding Fact Canonical Date GL Date is not populated in the OLTP application. So in the data warehouse, the GL Date for Oracle EBS is based on the Funding Allocation Date, using the GL Calendar of the Project OU. This enables cross-functional analysis on GL Calendar. For example, cross analysis of funding and billing by Fiscal Year is not possible if there is no GL Date in the Funding fact. Customers who do not want to perform analysis based on GL Calendar can instead base it on Enterprise Calendar.

> **Note:** The GL date (Funding Allocation Date) is the canonical date for this table and is also used for global exchange rate calculation.

15.2.2.25 About Configuring Cost Fact for Oracle EBS

Actual Costs are extracted from the Cost Distribution Lines table in the Project Costing module in Oracle EBS and loaded into the Cost Line Fact (W_PROJ_COST_LINE_F) table.

For Oracle EBS, Transaction Currency is the Document Currency for this fact.

> **Note:** Oracle EBS concurrent programs (such as PRC: Distribute Labor Costs and PRC: Distribute Usage and Miscellaneous Costs) for distributing Cost should be run before running the ETL to load the data warehouse. If the Cost Distribution program is not run before every incremental ETL run, the data in Cost Distribution Fact will not be synchronized with the actual expenditures in the Expenditure Fact table.

Expenditure Fact The Expenditure Fact (W_PROJ_EXP_LINE_F) is based on PA_EXPENDITURE_ITEMS_ALL. It shows the actual expenditure data before distribution. This fact should be used by customers who do not distribute their Expenditure on a daily basis, but who have some users who need to see a frequently updated view of Expenditure data.

> **Note:** The GL Date is assigned to the Cost Distribution Line only (during Cost distribution) and not to the Expenditure Item records. Therefore, the Expenditure data can only be analyzed by the Enterprise Calendar dimension and not by the GL calendar. Also, the Expenditure data cannot be analyzed by the GL Account because the GL account is associated only when the data is distributed.

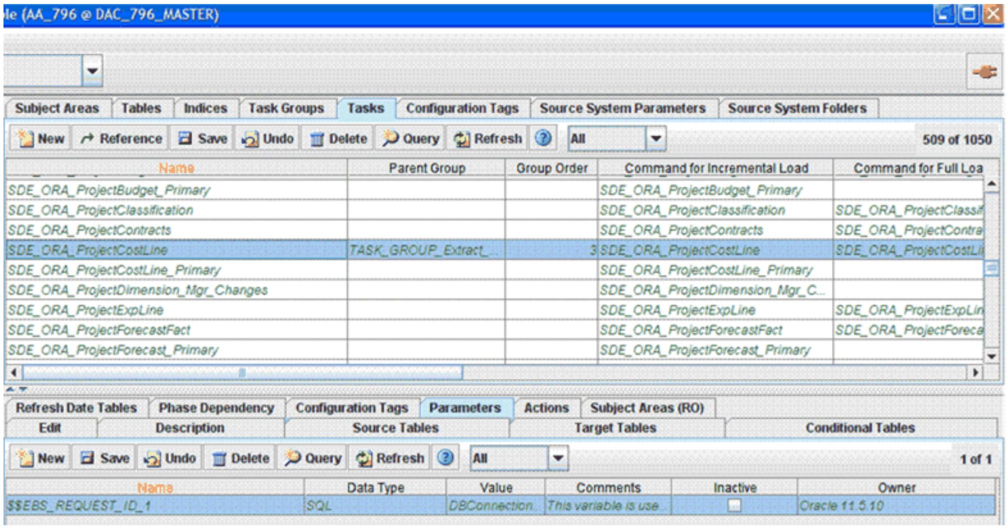
Cost Fact Canonical Date The Canonical Date dimension for the Cost fact is based on the RCVR_GL_DATE from Distribution Line table, whereas the Canonical Date dimension for the Expenditure fact is based on the EXPENDITURE_DATE from the Expenditure Items table.

The multi calendar date dimension contains calendars for multiple organizations. It is essential that all records in a report analyzing data by the Fiscal Calendar (Dim - Fiscal Calendar) point to the same calendar. For this reason, all reports in the dashboard are filtered on the Project Business Unit. To make all Cost records in a Project Business Unit point to the same calendar, the RCVR_GL_DATE and RCVR_PA_DATE columns are used to populate the GL_ACCOUNTING_DT_WID and PROJ_ACCOUNTING_DT_WID columns in the fact table respectively. Expenditure OU view (in Cost Fact) can be built using Enterprise Calendar as well.

About Domain Values for Cost Fact The Project Cost Transfer Status has been modeled as a domain value and can be configured using the domainValues_Project _Cost_Transfer_Status_ora11i.csv. For information, see Section 15.2.2.10, "How to Configure the domainValues_Project_Cost_Transfer_Status_oraVersion.csv."

Incremental Logic for Cost Fact The incremental extract logic for the Cost fact table depends on the 'REQUEST_ID' field of the Cost Distribution Lines table. The W_PROJ_ETL_PS parameter table facilitates this logic. Using a separate Informatica process, the maximum Request Id in the source table at the time of the ETL run is stored in this table, which is subsequently used to populate the SDE task (SDE_ORA_ProjectCostLine) level DAC variable $$EBS_REQUEST_ID_1. It is initialized using the following query:

```
SELECT COALESCE(PRE_REQUEST_ID,0) FROM W_PROJ_ETL_PS WHERE TBL_NAME = 'PA_COST_
DISTRIBUTION_LINES_ALL'
```



Name	Parent Group	Group Order	Command for Incremental Load	Command for Full Loa
SDE_ORA_ProjectBudget_Primary			SDE_ORA_ProjectBudget_Primary	
SDE_ORA_ProjectClassification			SDE_ORA_ProjectClassification	SDE_ORA_ProjectClassif
SDE_ORA_ProjectContracts			SDE_ORA_ProjectContracts	SDE_ORA_ProjectContra
SDE_ORA_ProjectCostLine	TASK_GROUP_Extract_...	3	SDE_ORA_ProjectCostLine	SDE_ORA_ProjectCostLi
SDE_ORA_ProjectCostLine_Primary			SDE_ORA_ProjectCostLine_Primary	
SDE_ORA_ProjectDimension_Mgr_Changes			SDE_ORA_ProjectDimension_Mgr_C...	
SDE_ORA_ProjectExpLine			SDE_ORA_ProjectExpLine	SDE_ORA_ProjectExpLin
SDE_ORA_ProjectForecastFact			SDE_ORA_ProjectForecastFact	SDE_ORA_ProjectForeca
SDE_ORA_ProjectForecast_Primary			SDE_ORA_ProjectForecast_Primary	

Name	Data Type	Value	Comments	Inactive	Owner
$$EBS_REQUEST_ID_1	SQL	DBConnection...	This variable is use...	□	Oracle 11.5.10

Note: If you are missing some Cost records in W_PROJ_COST_LINE_F after an incremental update, download patch 9896800 from My Oracle Support. The Tech Note included with the patch explains the scenarios where this can happen, and the proposed solution.

Cost Aggregate Table The Cost aggregate table facilitates high performance analysis. For more information about configuring this table, see Section 15.2.1.3, "Configuring the Project Cost Aggregate Table."

15.2.2.26 Configuring Revenue Fact for Oracle EBS

Actual Revenue Line records are extracted from the Revenue/Event Distribution Lines tables (PA_CUST_REV_DISTRIB_LINES_ALL and PA_CUST_EVENT_DIST_ALL) in the Project Costing module in Oracle EBS and are loaded into the Revenue Line Fact (W_PROJ_REVENUE_LINE_F) table.

For Oracle EBS, Revenue Transaction Currency Code is the Document Currency Code for this fact.

> **Note:** Oracle EBS concurrent programs (such as PRC: Generate Draft Revenue for a Single Project or PRC: Generate Draft Revenue for a Range of Projects) for distributing revenue should be run before the ETL is run to load the data warehouse.

For the Revenue Header Fact (W_PROJ_REVENUE_HDR_F), the primary source is the PA_DRAFT_REVENUES table. Revenue line metrics, such as Bill and Revenue amounts, are aggregated in this table as well.

Revenue Fact Canonical Date The Canonical Date dimension is based on the GL_DATE from the Draft Revenues table.

Revenue Facts Staging Table This is a common staging table that loads both the header and the line level revenue fact tables.

Revenue Fact Multicurrency Support Some metrics such as Unearned Revenue, Unbilled Receivables, Realized Gains, and Realized Losses are only available in Local Currency and Global Currencies. There are three columns in w_proj-revenue_line_f and w_proj-revenue_hdr_f respectively for revenue amounts in global currencies configured in DAC.

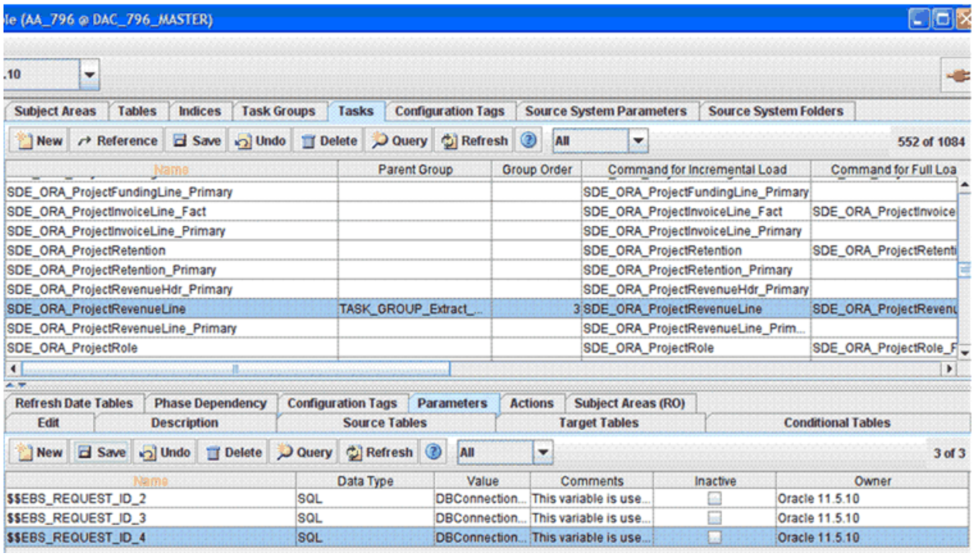
Revenue Fact Domain Values The project revenue status has been modeled as a domain value and can be configured using the domainValues_Project _ InvoiceRevenue _Status_ora11i.csv. For information about how to configure this file, see Section 15.2.2.6, "How to Configure the domainValues_Project_InvoiceRevenue_Status_oraVersion.csv."

Incremental Logic for Revenue Fact The incremental extract logic for the Revenue fact table depends on the REQUEST_ID field of the Revenue Distribution Lines table. The W_PROJ_ETL_PS parameter facilitates this logic, and through a separate Informatica process, the maximum Request Id in the source table at the time of the ETL run is stored in this table, which is subsequently used to populate the following variables for the SDE_ORA_ProjectRevenueLine task in DAC:

- $$EBS_REQUEST_ID_2
- $$EBS_REQUEST_ID_3
- $$EBS_REQUEST_ID_4

They are initialized using the following queries:

- `SELECT COALESCE(PRE_REQUEST_ID,0) FROM W_PROJ_ETL_PS WHERE TBL_NAME ='PA_CUST_EVENT_RDL_ALL'`
- `SELECT COALESCE(PRE_REQUEST_ID,0) FROM W_PROJ_ETL_PS WHERE TBL_NAME ='PA_CUST_REV_DIST_LINES_ALL'`
- `SELECT COALESCE(PRE_REQUEST_ID,0) FROM W_PROJ_ETL_PS WHERE TBL_NAME ='PA_DRAFT_REVENUES_ALL'`



Name	Parent Group	Group Order	Command for Incremental Load	Command for Full Loa
SDE_ORA_ProjectFundingLine_Primary			SDE_ORA_ProjectFundingLine_Primary	
SDE_ORA_ProjectInvoiceLine_Fact			SDE_ORA_ProjectInvoiceLine_Fact	SDE_ORA_ProjectInvoice
SDE_ORA_ProjectInvoiceLine_Primary			SDE_ORA_ProjectInvoiceLine_Primary	
SDE_ORA_ProjectRetention			SDE_ORA_ProjectRetention	SDE_ORA_ProjectRetenti
SDE_ORA_ProjectRetention_Primary			SDE_ORA_ProjectRetention_Primary	
SDE_ORA_ProjectRevenueHdr_Primary			SDE_ORA_ProjectRevenueHdr_Primary	
SDE_ORA_ProjectRevenueLine	TASK_GROUP_Extract_...	3	SDE_ORA_ProjectRevenueLine	SDE_ORA_ProjectRevenu
SDE_ORA_ProjectRevenueLine_Primary			SDE_ORA_ProjectRevenueLine_Prim...	
SDE_ORA_ProjectRole			SDE_ORA_ProjectRole	SDE_ORA_ProjectRole_F

Name	Data Type	Value	Comments	Inactive	Owner
$$EBS_REQUEST_ID_2	SQL	DBConnection...	This variable is use...	□	Oracle 11.5.10
$$EBS_REQUEST_ID_3	SQL	DBConnection...	This variable is use...	□	Oracle 11.5.10
$$EBS_REQUEST_ID_4	SQL	DBConnection...	This variable is use...	□	Oracle 11.5.10

Note: If you are missing some Revenue records in W_PROJ_REVENUE_LINE_F after an incremental update, download patch 9896800 from My Oracle Support. The Tech Note included with the patch explains the scenarios where this can happen, and the proposed solution.

15.2.2.27 Configuring Invoice Fact for Oracle EBS

Line level invoice information is extracted from the Invoice Line table (PA_DRAFT_INVOICE_ITEMS) in the Billing Module of Oracle EBS and loaded into Invoice Line Fact (W_PROJ_INVOICE_LINE_F). All invoices at any stage of the invoice generation process, such as creation, approval, release, transfer, and so forth, are loaded into this table so that customers can see a full view of the invoice generation process. Some of the information available in the Invoice Header Table (PA_DRAFT_INVOICES_ALL) such as GL Date and PA Date; and flags such as Write-Off Flag, Concession Flag, Cancelled Flag, and Retention Invoice Flag in Oracle EBS, have also been denormalized into Invoice Line Fact.

For Oracle EBS, Invoice Currency is the Document Currency for this fact.

Note: The Oracle EBS concurrent programs, such as PRC: Generate Draft Invoices for a Single Project or PRC: Generate Draft Invoices for a Range of Projects, for generating draft invoices, or PRC: Interface streamline Process, for transferring invoice to Receivables, should be run before the ETL is run to load the warehouse.

Invoice Fact Canonical Date The Canonical Date for this fact table is the GL Date, which is available in the PA_DRAFT_INVOICES_ALL table. This date is also used for global exchange rate calculation.

About Invoice Fact Domain Values The invoice status has been modeled as a domain value and can be configured using the domainValues_Project_ InvoiceRevenue_Status_ora11i.csv file. For information about how to configure this file, see Section 15.2.2.6, "How to Configure the domainValues_Project_InvoiceRevenue_Status_oraVersion.csv."

The invoice line type has also been modeled as a domain value and can be configured using the domainValues_Project_ Invoice_Line_ Type_ora11i.csv file. For more information about how to configure this file, see Section 15.2.2.7, "How to Configure the domainValues_Project_Invoice_Line_Type_oraVersion.csv."

Retention Fact The Retention fact table (W_PROJ_RETENTION_F) holds aggregated information about Retention Billing Metrics. It is based on PA_SUMMARY _ PROJECT_ RETN table in Oracle EBS. Before running any ETL job, you need to run the PRC: Refresh Project Summary Amounts process in Oracle EBS to update this table. Otherwise, the data in the Invoice Line fact table and aggregated Retention fact table will be inconsistent.

Out of the box, the Logical Table Source based on this table in the metadata repository (RPD) is disabled. If you want to use this table, you need to activate it by opening the RPD using the Oracle BI Administration Tool, and then going to Fact_W_PROJ_ RETENTION_F_Project_Retention LTS under Fact - Project Billing in the Business Model and Mapping layer and clearing the **Disabled** option.

The corresponding ETL tasks have not been deactivated out of the box in DAC. Therefore, if you do not use this fact table, access the Oracle 11510 and Oracle R12 containers in DAC and deactivate the SDE_ORA_ProjectRetention task. Also, access the DataWarehouse container in DAC and deactivate the following tasks:

- SIL_ProjectRetention_Fact
- SIL_ProjectRetentionFact_IdentifyDelete
- SIL_ProjectRetentionFact_SoftDelete

15.2.2.28 How to Tune Performance for Project Analytics for Oracle EBS

You can increase the ETL performance for Project Analytics for Oracle EBS by using hash joins in the Oracle database optimizer. In Informatica, configure the $$HINT1 parameter for each of the mappings listed in Table 15–4.

> **Note:** This configuration applies to Oracle databases only. Out of the box, this parameter is blank.

Table 15–4 Parameters for Increasing Performance for Project Analytics for Oracle EBS

Mapping	$$HINT1 Parameter
SDE_ORA_ProjectCostLine_ Fact	`/*+ USE_HASH(pa_cost_distribution_lines_all pa_expenditure_items_all pa_expenditures_all pa_implementations_all gl_sets_of_books pa_ project_assignments pa_lookups pa_projects_ all pa_project_types_all pa_expenditure_ types) */`
SDE_ORA_ProjectInvoiceLine_ Fact	`/*+ USE_HASH(pa_draft_invoice_items pa_tasks pa_draft_invoices_all pa_projects_all pa_ agreements_all pa_lookups) */`
SDE_ORA_ ProjectFundingHeader_Fact	`/*+USE_HASH(PA_PROJECTS_ALL PA_TASKS PA_ AGREEMENTS_ALL PA_SUMMARY_PROJECT_FUNDINGS) */`

15.2.3 Configuration Steps for Project Analytics for PeopleSoft

This section contains configuration steps before you do a full data load that apply to PeopleSoft. It contains the following topics:

15.2.3.1 About Persisted Stage for Oracle Project Analytics for PeopleSoft

Oracle Project Analytics for PeopleSoft is delivered using a persisted stage strategy for the incremental load process for the following PeopleSoft Project Analytics areas:

- Budget
- Revenue
- Cost

The PeopleSoft source system tables for these analytics areas do not reliably update the date columns. This prevents the use of the standard incremental load process.

The persisted stage strategy enables incremental loads of data from these tables. Each time the persisted stage workflows are run, a full extract from the OLTP to the OLAP is executed. Each time a record is added to the OLAP persisted stage table or on any full load to the OLAP persisted stage table, dates are updated in the persisted stage table to the current system date. If a record already exists in the OLAP persisted stage table and has been updated in the OLTP, the dates are also updated to the current system date. Once the persisted stage tables are populated, ETLs are then run to extract only new and updated records from the persisted stage tables to the fact staging tables.

The benefit of using persisted stage for incremental loads is that the system only updates records in the fact staging tables that have changed since the previous ETL run. The persisted stage tables require additional space in the database. Therefore, Oracle gives you the option to configure the ETL process to run in a non-persisted stage (full extract and full load for every ETL run) instead of persisted stage.

You can configure persisted stage or non-persisted stage for each individual area: Budget, Cost, or Revenue. However, you cannot configure any of these areas for both persisted stage and non-persisted stage.

To configure non-persisted stage for Budget, Revenue, or Cost:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. On the Subject Areas tab, select the Projects Subject Area.
3. Add any of the following Configuration tags, depending on the area that you want to change to non-persisted stage:
   - Project_NonPersistedStage_Budget
   - Project_NonPersistedStage_Cost
   - Project_NonPersistedStage_Revenue
4. For each Configuration tag that you added for non-persisted stage, remove its persisted stage Configuration tag.

   For example, remove any or all of the following tags:
   - Project_PersistedStage_Budget
   - Project_PersistedStage_Cost
   - Project_PersistedStage_Revenue
5. Reassemble the Projects Subject Area.

   You must perform this step to save the updated tags.
6. Rebuild the applicable version of the PeopleSoft Execution Plan, for example Projects - PeopleSoft 8.9.

> **Note:** To revert to persisted stage, reverse the steps in this task by removing Configuration tags for non-persisted stage and adding Configuration tags for persisted stage.

15.2.3.2 About Configuring Domain Values and CSV Worksheet Files for Oracle Project Analytics

If you modify or extend a seeded list of values, you must configure the CSV files for Oracle Project 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 from the seeded values, you need to follow the instructions to configure the domain values and CSV worksheet files.

Table 15–5 Domain Values and CSV Worksheet Files for Oracle Project Analytics and PeopleSoft

Worksheet File Name	Description	Session
domainValues_Resource_Class_XACT_TYPE_D_PSFT.csv	Lists the Transaction types and the corresponding domain values for the Oracle PSFT 89/90 application. For more information about updating the values in this file, see Section 15.2.3.3, "How to Configure Resource Class Domain Values for the Transaction Type Dimension."	SDE_PSFT_TransactionTypeDimension_ResourceClass
domainValues_Project_Invoice_Line_Type_psft.csv	Lists the Transaction Types and the corresponding domain values for the Oracle PSFT 89/90 application. For more information about updating the values in this file, see Section 15.2.3.4, "How to Configure Invoice Line Domain Values for the Transaction Type Dimension."	SDE_PSFT_TransactionTypeDimension_InvoiceLineType
domainValues_Project_Type_Class_code_psft.csv	Lists the Project Type and the corresponding domain values for the Oracle PSFT 89/90 application. For more information about updating the values in this file, see Section 15.2.3.5, "How to Configure Project Type Class Code Domain Values for the Project Dimension."	SDE_PSFT_CodeDimension_Project_Project_Type
domainValues_Project_Status_code_psft.csv	Lists the Project Status and the corresponding domain values for the Oracle PSFT 89/90 application. For more information about updating the values in this file, see Section 15.2.3.6, "How to Configure Project Status Domain Values for the Status Dimension."	SDE_PSFT_StatusDimension_ProjectStatus_Full
domainValues_transfer_status_code_psft.csv	Lists the Transfer Status and the corresponding domain values for the Oracle PSFT 89/90 application. For more information about updating the values in this file, see Section 15.2.3.7, "How to Configure Transfer Status Domain Values for the Status Dimension."	SDE_PSFT_StatusDimension_TransferStatus_Full
domainValues_revenue_status_code_psft.csv	Lists the Revenue Status and the corresponding domain values for the Oracle PSFT 89/90 application. For more information about updating the values in this file, see Section 15.2.3.8, "How to Configure Revenue Status Domain Values for the Status Dimension."	SDE_PSFT_StatusDimension_RevenueStatus_Full
domainValues_invoice_status_code_psft.csv	Lists the Invoice Status and the corresponding domain values for the Oracle PSFT 89/90 application. For more information about updating the values in this file, see Section 15.2.3.9, "How to Configure Invoice Status Domain Values for the Status Dimension."	SDE_PSFT_StatusDimension_InvoiceStatus_Full

15.2.3.3 How to Configure Resource Class Domain Values for the Transaction Type Dimension

This section describes how to configure Resource Class domain values for the Transaction Type dimension.

1. Identify the Resource Class Types in the PeopleSoft source system by using the following SQL:

```
SELECT
  FIELDVALUE,
  XLATSHORTNAME,
  XLATLONGNAME,
  LASTUPDDTTM,
  LASTUPDOPRID
FROM
  PSXLATITEM
 WHERE
  FIELDNAME='RESOURCE_CLASS'
  AND EFF_STATUS = 'A';
```

2. Using a text editor, open the domainValues_Resource_Class_XACT_TYPE_D_PSFT.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   In the text editor view, distinct records are listed in separate lines and the attributes are separated by commas. The records start on the next line after the last updated line.

3. Copy the Lookup Code and Meaning from the SQL results to the SOURCE_TYPE_CODE and SOURCE_TYPE_NAME columns in the CSV file respectively.

   The data must be copied starting from the 6th line. Use commas to separate the entries.

4. Map each FIELDVALUE to the XACT_TYPE_CODE column.

5. Enter the corresponding W_XACT_TYPE_CODE and W_XACT_TYPE_DESC.

   For more information about Transaction Type, Resource Class domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

6. Save and close the file.

15.2.3.4 How to Configure Invoice Line Domain Values for the Transaction Type Dimension

This section describes how to configure Invoice Line domain values for the Transaction Type dimension.

1. Identify the Invoice Line Type in your PeopleSoft source system by using the following SQL:

```
SELECT
  FIELDVALUE,
  XLATSHORTNAME,
  XLATLONGNAME,
  LASTUPDDTTM,
  LASTUPDOPRID
FROM
  PSXLATITEM
 WHERE
  FIELDNAME='LINE_TYPE'
  AND EFF_STATUS = 'A';
```

2. Using a text editor, open the domainValues_Project_Invoice_Line_Type_psft.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

   If the CSV file is opened with a text editor, distinct records are listed in separate lines and the attributes are separated by commas. The records start on the next line after the Last Updated line.

3. Copy the Lookup Code and Meaning from the SQL results to the SOURCE_TYPE_CODE and SOURCE_TYPE_NAME columns in the CSV file respectively.

   The data must be copied starting from the 6th line. Use commas to separate the entries.

4. Map each of these values from the SQL query results to the CSV file:

   - FIELDVALUE to the XACT_TYPE_CODE column
   - XLATSHORTNAME to the XACT_TYPE_NAME column

- XLATLONGNAME to the XACT_TYPE_DESC column

5. Enter the corresponding W_XACT_TYPE_CODE and W_XACT_TYPE_DESC.

   These domain values are user defined.

   For more information about Invoice Line Type, Transaction Type domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

6. Save and close the file.

15.2.3.5 How to Configure Project Type Class Code Domain Values for the Project Dimension

This section describes how to configure Project Type Class Code domain values for the Project dimension.

1. Identify the Project Type Class Codes in your Project Dimension by using the following SQL:

```
SELECT T.PROJECT_TYPE||'~'||T.SETID AS PROJECT_TYPE,
T.DESCR
FROM PS_PROJ_TYPE_TBL T
WHERE T.EFFDT = (SELECT MAX(EFFDT) FROM PS_PROJ_TYPE_TBL T1
 WHERE T1.SETID = T.SETID AND T1.PROJECT_TYPE = T.PROJECT_TYPE
   AND T1.EFFDT <= CURRENT_TIMESTAMP
   GROUP BY T1.SETID, T1.PROJECT_TYPE)
```

> **Note:** PeopleSoft does not have a value directly corresponding to Project Type Class, so you must use this flat file to specify a value to use for this column. This flat file populates both the PROJECT_TYPE_CLASS_CODE and the W_PROJECT_TYPE_CLASS_CODE. The flat file also populates the W_PROJECT_TYPE_CLASS_DESC column.

2. Using a text editor, open the domainValues_Project_Type_Class_code_psft.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).

3. Copy the data from the PROJECT_TYPE column to the PROJECT_TYPE_CLASS_CODE column in the CSV file.

4. Map each Project Type Class to a domain value.

   For more information on Project Type Class Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

5. Map each Project Type Class to a value of Y or N in the CAPITALIZABLE_FLG column. If the Project Type Class is considered capitalizable, enter Y. Otherwise, enter N.

6. Save and close the file.

15.2.3.6 How to Configure Project Status Domain Values for the Status Dimension

This section describes how to configure Project Status domain values for the Status dimension.

1. Identify the Project Type Class Codes in your Project Dimension by using the following SQL:

```
SELECT T.PROJECT_STATUS ||'~'||T.SETID, T.DESCR, T.PROCESSING_STATUS,
 UPPER(P.XLATSHORTNAME) AS PROCESSING_STATUS_DESC
```

```
FROM PS_PROJ_STATUS_TBL T, PSXLATITEM P
WHERE T.PROCESSING_STATUS = FIELDVALUE AND FIELDNAME = 'PROCESSING_STATUS' AND
  T.EFFDT = (SELECT MAX(EFFDT) FROM PS_PROJ_STATUS_TBL X
WHERE X.EFFDT <= SYSDATE AND X.PROJECT_STATUS=T.PROJECT_STATUS AND X.SETID =
T.SETID
  GROUP BY X.SETID, X.PROJECT_STATUS)
AND P.EFFDT = (SELECT MAX(EFFDT) FROM PSXLATITEM X1
WHERE X1.EFFDT <= SYSDATE AND X1.FIELDNAME = 'PROCESSING_STATUS' AND
X1.FIELDVALUE = P.FIELDVALUE AND X1.EFF_STATUS = 'A'
  GROUP BY FIELDNAME)
```

PeopleSoft uses the Project Status for Status Code and the Processing Status for the Substatus Code.

2. Using a text editor, open the domainValues_Project_Status_code_psft.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the data in the PROJECT_STATUS column to STATUS_CODE column in the CSV file.
4. Copy the data from the PROCESSING_STATUS_DESC column to the W_SUBSTATUS and W_SUBSTATUS_DESC columns in the CSV file.
5. Map each Project Status to a domain value.

   For more information on Project Status domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
6. Save and close the file.

15.2.3.7 How to Configure Transfer Status Domain Values for the Status Dimension

This section describes how to configure Transfer Status domain values for the Status dimension.

1. Identify the Transfer Status codes in your Status dimension by using the following SQL:

```
SELECT FIELDVALUE AS TRANSFER_STATUS, EFF_STATUS, XLATSHORTNAME AS TRANSFER_
STATUS_NAME
FROM PSXLATITEM
WHERE FIELDNAME = 'GL_DISTRIB_STATUS' AND
EFFDT <= SYSDATE
```

   PeopleSoft uses the GL Distrib Status for Transfer Status.
2. Using a text editor, open the domainValues_transfer_status_code_psft.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the data in the TRANSFER_STATUS column to the STATUS_CODE column in the CSV file.
4. Map each Transfer Status to a domain value.

   For more information on Transfer Status domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

15.2.3.8 How to Configure Revenue Status Domain Values for the Status Dimension

The source for the Revenue Status records in the Status dimension is the sourceValues_revenue_status_code_psft.csv flat file. These values are derived based on GL_DISTRIB_STATUS and BI_DISTRIB_STATUS, which are system codes. These values are determined based on whether or not the GL Status and the Billing Status are D-Distributed.

To configure Revenue Status domain values for the Status dimension:

1. Using a text editor, open the domainValues_revenue_status_code_psft.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
2. Copy the data from the STATUS_CODE column to the STATUS_CODE column in the CSV file.
3. Map each Revenue Status to a domain value.

   For more information on Revenue Status domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
4. Save and close the file.

15.2.3.9 How to Configure Invoice Status Domain Values for the Status Dimension

This section describes how to configure Invoice Status domain values for the Status dimension.

1. Identify the Invoice Status codes in your Status dimension by using the following SQL:

   ```
   SELECT DISTINCT FIELDVALUE AS INVOICE_STATUS, EFF_STATUS, XLATSHORTNAME AS
   INVOICE_STATUS_DESC FROM PSXLATITEM
   WHERE FIELDNAME = 'BILL_STATUS'
   ```
2. Using a text editor, open the domainValues_invoice_status_code_psft.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
3. Copy the data from the INVOICE_STATUS column to the STATUS_CODE column in the CSV file.
4. Map each Invoice Status to a domain value.

   For more information on Invoice Status domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

15.2.3.10 How to Configure Project Budget

Cost Budget is extracted from Project Costing for all Analysis Types within the project's Cost Budget Analysis Group. All extracted Cost Budgets are loaded into the Budget fact table as Raw Cost unless you perform one or both of the following configurations described in this section:

- Section 15.2.3.10.1, "Identifying Project Budget Burden Costs Based on Analysis Type"
- Section 15.2.3.10.2, "Identifying Project Budget Burden Costs Based on a Source Type, Category, and Subcategory Combination of Values"

In addition, you must perform the steps in this section:

- Section 15.2.3.10.3, "How to Configure Project Budget Analytics"

15.2.3.10.1 Identifying Project Budget Burden Costs Based on Analysis Type The ETL process uses the file_Project_Budget_Burden_Analysis_Type_psft.csv flat file to list all Analysis Types for Project Budget Burden Cost. If the ETL process finds the Analysis Type in this flat file, it will not perform further lookups against other lookup tables to determine Project Budget Burden Cost.

To configure the file_Project_Budget_Burden_Analysis_Type_psft.csv file:

1. Using a text editor, open the file_Project_Budget_Burden_Analysis_Type_psft.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
2. Enter a list of Analysis Types to be considered as Project Budget Burden Costs. The format is XXX,1 where XXX is an Analysis Type. The 1 is used as a return value to indicate that this is a Project Budget Burden Cost.

   The following is an example of classifying Costs with BUR and BRD Analysis Types as Project Budget Burden Costs:

   ```
   BUR,1
   BRD,1
   ```
3. Save and close the file.

15.2.3.10.2 Identifying Project Budget Burden Costs Based on a Source Type, Category, and Subcategory Combination of Values You must configure the following flat files to identify Project Budget Burden Costs based on a Source Type, Category, and Subcategory combination of values:

- file_Project_Cost_Burden_TypeCatSub_config_psft.csv

  The ETL process uses this flat file to designate which columns (Source Type, Category, and Subcategory) are used in the lookup. A DAC parameter determines if this lookup is performed for an implementation.
- file_Project_Cost_Burden_TypeCatSub_psft.csv

  The ETL process uses this flat file to list all Source Type, Category, and Subcategory combination of values to use for Project Budget Burden Cost.

> **Note:** Both Project Budget and Project Cost use these flat files, along with a DAC parameter, to identify Burden Cost. You can customize these files if the requirements differ between Budget and Cost in your implementation.

For information about how to configure Project Cost, see Section 15.2.3.11, "How to Configure Project Cost."

To configure the file_Project_Cost_Burden_TypeCatSub_config_psft.csv file:

1. Using a text editor, open the file_Project_Cost_Burden_TypeCatSub_config_psft.csv file located in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).
2. Enter only one row with RowID of 1. Enter a Y in each column that represents the combination to be evaluated as a Burden Cost. The columns are:
   - Row ID

- Source Type
- Category
- Subcategory

The following example shows how to use combinations of Source Type and Category:

```
1,Y,Y,
```

3. Save and close the file.

To configure the file_Project_Cost_Burden_TypeCatSub_psft.csv file:

1. Using a text editor, open the file_Project_Cost_Burden_TypeCatSub_psft.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
2. Enter a list of Resource Type, Resource Category, and Resource Subcategory combinations to be considered as Burden costs. The format is:

```
XXXXX,XXXXX,XXXXX,1
```

XXXXX is a combination of Resource Type, Resource Category, and Resource Subcategory

The 1 is a return value that indicates that this is a Project Budget Burden Cost. Each combination of lookup values must be specified. Wildcards are not supported.

The following is an example of classifying costs with G&A or FRNG Source Type as Project Budget Burden Costs:

```
G&A,,,1
FRNG,,,1
```

> **Note:** This CSV file is used in conjunction with the file_Project_Cost_Burden_TypeCatSub_config_psft.csv configuration file. In this example, this configuration file would contain the value 1,Y.

Given the example configuration above, Project Costing records with the following Source Type, Category, Subcategory combinations are considered Project Budget Burden Costs:

Source Type	Category	Subcategory
G&A	-	-
FRNG	LUX	TEMP
FRNG	BONUS	-

> **Note:** You must specify each combination of lookup values. The lookup will use columns with a Y in the configuration file.

3. Save and close the file.

15.2.3.10.3 How to Configure Project Budget Analytics This section describes how to configure Project Budget Analytic.s

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.

   For information about how to log into DAC, see *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

2. Display the Source System Parameters tab.
3. Set the following parameters:

Parameter Name	Description
$$BURDEN_ANALYSIS_TYPE	Use this parameter to specify Analysis Types as Burden Cost for the lookup. Valid values are: ■ 1. Enables the implementation to perform this lookup. ■ 0. (Default) Disables this lookup.
$$BURDEN_TYPECATSUB	Use this parameter to specify a combination of Source Type, Category, and Subcategory values as Burden Cost for the lookup. Valid values are: ■ 1. Enables this lookup. ■ 0. (Default) Disables this lookup.

> **Note:** The $$BURDEN_ANALYSIS_TYPE and $$BURDEN_TYPECATSUB parameters are used for both Project Budget and Project Cost analytics. Changing these parameters for Project Budget affects Project Cost.

15.2.3.11 How to Configure Project Cost

Actual Costs are extracted from Project Costing for all Analysis Types within the project's Actual Cost Analysis Group.

All costs extracted will be loaded into the Cost Fact Line table as Raw Cost unless you perform one or both of the following configurations:

- Section 15.2.3.11.1, "Identifying Project Cost Burden Costs based on Analysis Type"
- Section 15.2.3.11.2, "Identifying Project Cost Burden Costs based on a Source Type, Category, and Subcategory Combination of Values"

All extracted Costs are loaded into the Cost Fact Line table without a Resource Class assigned unless you perform one or both of the following configurations:

- Section 15.2.3.11.3, "Assigning Resource Class based on a Source Type, Category, and Subcategory Combination of Values"
- Section 15.2.3.11.4, "Assigning Resource Class Based on a ChartField Combination of Values"

In addition, you must perform the steps in this section:

- Section 15.2.3.11.5, "How to Configure Project Cost Analytics"

15.2.3.11.1 Identifying Project Cost Burden Costs based on Analysis Type The ETL process uses the file_Project_Cost_Burden_Analysis_Type_psft.csv flat file to list all Analysis

Types for Project Cost Burden Cost. If the ETL process finds the Analysis Type in this flat file, it will not perform further lookups against other lookup tables to determine Project Cost Burden Cost.

To identify the Project Cost Burden Costs based on Analysis Type:

1. Using a text editor, open the file_Project_Cost_Burden_Analysis_Type_psft.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
2. Enter a list of Analysis Types to be considered as Burden Costs. The format is XXX,1, where XXX is an Analysis Type. The 1 is a return value that indicates that this is a Burden Cost.

   The following example shows how to classify Costs with BUR and BRD Analysis Types as Burden Costs:

   ```
   BUR,1
   BRD,1
   ```
3. Save and close the file.

15.2.3.11.2 Identifying Project Cost Burden Costs based on a Source Type, Category, and Subcategory Combination of Values You must configure the following flat files to identify Project Cost Burden Costs based on a Source Type, Category, and Subcategory combination of values:

- file_Project_Cost_Burden_TypeCatSub_config_psft.csv

  Use this flat file to specify the columns (Source Type, Category, and Subcategory) to use in the lookup. A DAC parameter determines if this lookup should be performed for an implementation.
- file_Project_Cost_Burden_TypeCatSub_psft.csv

  Use this flat file to list all Source Type, Category, and Subcategory combination of values to use as Project Cost Burden Cost.

> **Note:** Both Project Budget and Project Cost use these flat files, along with a DAC parameter, to identify Burden Costs. You can customize these files if the requirements differ between Project Budget and Project Cost in your implementation.

For information about how to configure Project Budget, see Section 15.2.3.10, "How to Configure Project Budget."

To configure the file_Project_Cost_Burden_TypeCatSub_config_psft.csv file:

1. Using a text editor, open the file file_Project_Cost_Burden_TypeCatSub_config_psft.csv file located in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).
2. Enter only one row with RowID of 1. Enter a Y in each column that represents the combination to be evaluated as a Project Cost Burden Cost. The columns are:
   - Row ID
   - Source Type
   - Category
   - Subcategory

The following is an example of using a combination of Source Type and Category:

```
1,Y,Y
```

3. Save and close the file.

To configure the file_Project_Cost_Burden_TypeCatSub_psft.csv file:

1. Using a text editor, open the file_Project_Cost_Burden_TypeCatSub_psft.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
2. Enter a list of Resource Type, Resource Category, and Resource Subcategory combinations to be considered as Project Cost Burden Costs. The format is:

   ```
   XXXXX,XXXXX,XXXXX,1
   ```

   XXXXX represents each combination of Resource Type, Resource Category, and Resource Subcategory.

   The 1 is a return value that indicates that this is a Burden Cost. Each combination of lookup values must be specified. Wildcards are not supported.

   The following example shows how to classify Costs with G&A or FRNG Source Type as Project Cost Burden Costs:

   - G&A,,,1
   - FRNG,,,1

   Note: This CSV file is used in conjunction with the file_Project_Cost_Burden_TypeCatSub_config_psft.csv configuration file. In this example, this configuration file would contain the value 1,Y.

   In the example above, Project Costing records with the following Source Type, Category, and Subcategory combinations would be considered Project Cost Burden Costs:

Source Type	Category	Subcategory
G&A	-	-
FRNG	LUX	TEMP
FRNG	BONUS	-

   You must specify each combination of lookup values. The lookup will use columns with a Y in the configuration file.

3. Save and close the file.

15.2.3.11.3 Assigning Resource Class based on a Source Type, Category, and Subcategory Combination of Values

The ETL process uses the domainValues_Project_Cost_Resource_Class_TypeCatSub_psft.csv flat file to assign Resource Class to project cost records. If the ETL process finds the Resource Class in this flat file, it will not perform further lookups against other lookup tables to determine Project Cost Resource Class.

You must configure the following flat files to identify Resource Class based on a Source Type, Category, and Subcategory Combination of Values:

- file_Project_Cost_Resource_Class_TypeCatSub_config_psft.csv

Use this file to specify the columns (Source Type, Category, and Subcategory) to use in the lookup. A DAC parameter determines if this lookup should be performed for an implementation.

- domainValues_Project_Cost_Resource_Class_TypeCatSub_psft.csv

  The ETL process uses this flat file to list all Source Type, Category, Subcategory combinations of values to use for Resource Class. Enter values for only the columns that are selected in the file_Project_Cost_Resource_Class_TypeCatSub_config_psft.csv file. All columns must be included in the flat file and unselected columns must not contain a value. You must identify each row as either People (L) or Equipment (A) as the last value.

To configure file_Project_Cost_Resource_Class_TypeCatSub_config_psft.csv:

1. Using a text editor, open the file file_Project_Cost_Resource_Class_TypeCatSub_config_psft.csv file located in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).
2. Enter only one row with RowID of 1. Enter a Y in each column that represents the combination to be assigned a Resource Class. The columns are:
   - Row ID
   - Source Type
   - Category
   - Subcategory

   The following is an example of using a combination of Source Type and Category:

   ```
   1,Y,Y,
   ```

   In this example, Source Type and Category combinations stored in domainValues_Project_Cost_Resource_Class_TypeCatSub_psft.csv are classified as People or Equipment when the values match.
3. Save and close the file.

To configure the domainValues_Project_Cost_Resource_Class_TypeCatSub_psft.csv file:

1. Using a text editor, open the domainValues_Project_Cost_Resource_Class_TypeCatSub_psft.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
2. Enter Resource Type, Category, and Subcategory combinations that are to be considered as Resource Class People or Equipment. For Resource Class of People, enter an L as the last value. For Resource Class of Equipment, enter an A as the last value. The format is:

   ```
   XXXXX,XXXXX,XXXXX,X
   ```

   You must specify each combination of lookup values. Wildcards are not supported.

   The following is an example of classifying costs with LABOR or SUBCN Source Type/no Category as People costs and costs with DIRCT Source Type/HRDWR Category as Equipment costs:

   ```
   LABOR,,,L
   SUBCN,,,L
   DIRCT,HRDWR,,A
   ```

Note: This CSV file is used in conjunction with the file_Project_Cost_Resource_Class_TypeCatSub_config_psft.csv configuration file. In this example, this configuration file would contain the value 1,Y,Y,

In the example above, Project Costing records with the following Source Type, Category, and Subcategory combinations are classified as Resource Class:

Source Type	Category	Subcategory	Resource Class
LABOR	-	-	People
LABOR	TEMP	-	-
SUBCN	-	-	People
SUBCN	ANAL	-	-
DIRCT	-	-	-
DIRCT	HRDWR	-	Equipment

Note: You must specify each combination of lookup values. The lookup will use columns with a Y in the configuration file.

If the Resource Class is found in the Resource Class Type flat file, further lookups against other lookup tables for Resource Class will not be made.

3. Save and close the file.

15.2.3.11.4 Assigning Resource Class Based on a ChartField Combination of Values

The ETL process uses the domainValues_Project_Cost_Resource_Class_ChartField_psft.csv flat file to assign Resource Class to Project Cost records.

You must configure the following flat files to assign Resource Class based on a Chartfield combination of values:

- file_Project_Cost_Resource_Class_ChartField_config_psft.csv

  Use this flat file to specify the Chartfield columns to use in the lookup. A DAC parameter determines if this lookup is performed in an implementation.

- domainValues_Project_Cost_Resource_Class_ChartField_psft.csv

  Use this flat file to assign all ChartField combinations of values to a Resource Class. Enter values for only the columns that are selected in the file_Project_Cost_Resource_Class_ChartField_config_psft.csv file. All columns must be included in the flat file and unselected columns must not contain a value. You must identify each row as either People (L) or Equipment (A) as the last value.

To configure the file_Project_Cost_Resource_Class_ChartField_config_psft.csv:

1. Using a text editor, open the file_Project_Cost_Resource_Class_ChartField_config_psft.csv file located in the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).
2. Enter only one row with RowID of 1. Enter a Y in each column that represents the combination to be assigned a Resource Class. The columns are:
   - Row ID
   - Account
   - Alternate Account

- Operating Unit
- Fund
- Dept ID
- Program
- Class
- Budget
- Project
- Business Unit
- Project
- Activity
- Source Type
- Category
- Subcategory
- Affiliate
- Affiliate 1
- Affiliate 2
- ChartField 1
- ChartField 2
- ChartField 3

The following is an example of using a combination of Fund Code and Program:

```
,,,Y,,Y,,,,,,,,,,,,,,,
```

In this example, Fund Code and Program Code combinations stored in the domainValues_Project_Cost_Resource_Class_ChartField_psft.csv are classified as People or Equipment when the values match.

3. Save and close the file.

To configure the domainValues_Project_Cost_Resource_Class_ChartField_psft.csv:

1. Using a text editor, open the domainValues_Project_Cost_Resource_Class_ChartField_psft.csv file located in the $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\LkpFiles).
2. Enter ChartField combinations that are to be considered as Resource Class People or Equipment. For Resource Class of People, enter an L as the last value. For Resource Class of Equipment, enter an A as the last value. The format is:

```
X,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X,X
```

X represents a Chartfield combination.

Each combination of lookup values must be specified. Wildcards are not supported.

The following example shows how to classify costs with Fund Code FND01 and Program Code P2008 as People costs:

```
,,,FND01,, P2008,,,,,,,,,,,,,,,,,L
```

> **Note:** This CSV file is used in conjunction with the file_Project_Cost_Resource_Class_ChartField_config_psft.csv configuration file. In this example, this configuration file would contain the value
> ,,,Y,,Y,,,,,,,,,,,,,,,

In the above example, Project Costing records with the Fund Code FND01 and Program Code P2008 are classified as Resource Class People.

You must specify each combination of lookup values. Columns with a Y in the configuration file will be considered in the lookup.

3. Save and close the file.

15.2.3.11.5 How to Configure Project Cost Analytics This section describes how to configure Project Cost Analytics.

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.

   For information about how to log into DAC, see *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

2. Display the Source System Parameters tab.
3. Set the following parameters:

Parameter Name	Description
$$BURDEN_ANALYSIS_TYPE	Use this parameter to specify Analysis Types as Burden Cost for the lookup. Valid values are: ■ 1. Enables the implementation to perform this lookup. ■ 0. (Default) Disables this lookup.
$$BURDEN_TYPECATSUB	Use this parameter to specify a combination of Source Type, Category, and Subcategory values as Burden Cost for the lookup. Valid values are: ■ 1. Enables this lookup. ■ 0. (Default) Disables this lookup.
$$RESOURCE_CLASS_TYPECATSUB	Use this parameter to specify a combination of Source Type, Category, and Subcategory values to determine Resource Class for the lookup. Valid values are: ■ 1. (Default) Enables this lookup. ■ 0. Disables this lookup.
$$RESOURCE_CLASS_CHARTFIELD	Use this parameter to specify a ChartField combination of values to determine Resource Class for the lookup. Valid values are: ■ 1. Enables this lookup. ■ 0. (Default) Disables this lookup.

> **Note:** The $$BURDEN_ANALYSIS_TYPE and $$BURDEN_TYPECATSUB parameters are used for both Project Cost and Project Budget analytics. Changing these parameters for Project Cost affects Project Budget.

15.2.3.12 Deleting Certain Physical Joins from the RPD (PeopleSoft Only)

You must ensure that the logical table source 'Dim_W_MCAL_PERIOD_D_Fiscal_Period' for the dimension table 'Dim - Date Fiscal Calendar' is enabled. In addition, you must delete certain Physical layer joins because the Period_Wid in Revenue fact tables is not populated.

To update Project Analytics logical table sources and delete Physical layer joins:

1. Using the Administration Tool, open OracleBIAnalyticsApps.rpd.

   The OracleBIAnalyticsApps.rpd file is located at:

   ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_obis*n*\repository

2. Go to the Business Model and Mapping layer and expand the Core business model.
3. Scroll down to the Dim - Date Fiscal Calendar logical table and open its Sources folder.
4. In the list of logical table sources, right-click Dim_W_MCAL_PERIOD_D_Fiscal_Period.
5. Select **Properties**.
6. Display the General tab in the Properties dialog and ensure that the **Disabled** option is *not* selected.
7. Click **OK**.
8. In the Physical layer, delete the following joins:

   Dim_W_MCAL_PERIOD_D_Fiscal_Period.ROW_WID = Fact_W_PROJ_REVENUE_HDR_F_Revenue_Header.GL_ACCOUNTING_PERIOD_WID

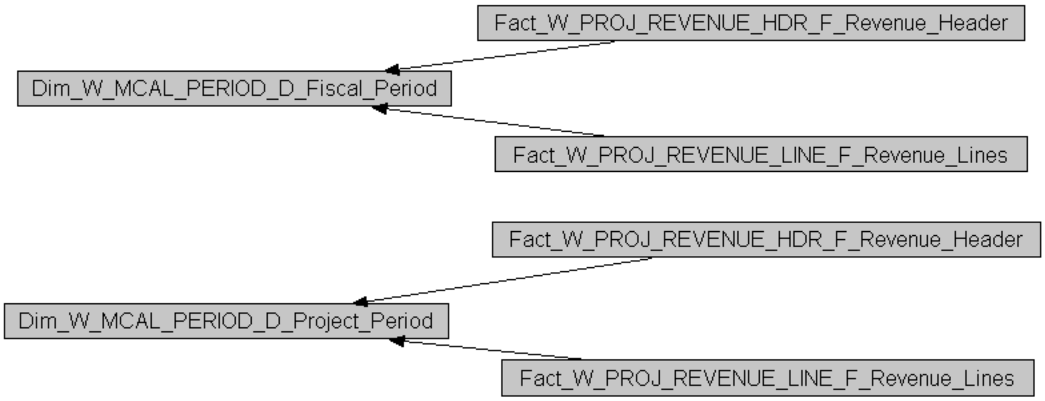
   Dim_W_MCAL_PERIOD_D_Fiscal_Period.ROW_WID = Fact_W_PROJ_REVENUE_LINE_F_Revenue_Lines.GL_ACCOUNTING_PERIOD_WID

   Dim_W_MCAL_PERIOD_D_Project_Period.ROW_WID = Fact_W_PROJ_REVENUE_HDR_F_Revenue_Header.PROJ_ACCOUNTING_PERIOD_WID

   Dim_W_MCAL_PERIOD_D_Project_Period.ROW_WID = Fact_W_PROJ_REVENUE_LINE_F_Revenue_Lines.PROJ_ACCOUNTING_PERIOD_WID

   Figure 15–1 shows the joins that you need to delete.

Figure 15–1 Oracle Project Analytics Physical Joins to Be Deleted for PeopleSoft


Fact_W_PROJ_REVENUE_HDR_F_Revenue_Header
Dim_W_MCAL_PERIOD_D_Fiscal_Period
Fact_W_PROJ_REVENUE_LINE_F_Revenue_Lines
Fact_W_PROJ_REVENUE_HDR_F_Revenue_Header
Dim_W_MCAL_PERIOD_D_Project_Period
Fact_W_PROJ_REVENUE_LINE_F_Revenue_Lines

9. Save the repository.
10. Restart the Oracle BI Server.
11. Clear the Oracle BI EE cache.

15.2.3.13 Operating Unit Based Security with PeopleSoft

All fact tables in Project Analytics are secured by Operating Unit (or Business Unit) security. This security filter uses the OU_ORG session variable, which is initialized by the list of Operating Units the user has access to in the OLTP system. An additional join is added to each fact report, so only transaction data for Projects that are owned by the Project Operating Units that the user is allowed to access are shown in the report. Dimension reports are not secured by filters.

For more information about how to add security to dimensions and implementation steps for operating unit based security with PeopleSoft, see *Oracle Business Intelligence Applications Security Guide*.

15.2.3.14 About Configuring Project Funding Fact

Funding is based on Funding Line, which represents allocations made to a project or task. The line level funding information is held in the Project Funding Line fact (W_ PROJ_ FUNDING_ LINE_F). PeopleSoft Enterprise will only source the Agreement Amount which is based on the Contracting Detail table PS_CA_DETAIL.

The Project Funding Header fact (W_PROJ_FUNDING_HDR_F) contains an additional aggregated Agreement Amount metric per PeopleSoft Enterprise Contract.

Funding Fact Canonical Date Both GL Date and Funding Allocation Date are not populated in the OLTP application. Therefore in the data warehouse, the GL Date and Funding Allocation Date for PeopleSoft Enterprise are based on the Last Update Date Time Stamp, using the GL Calendar of the Project Business Unit. This enables cross-functional analysis on GL Calendar. For example, cross analysis of funding and billing by Fiscal Year is not possible if there is no GL Date in the Funding fact. Customers who do not want to perform analysis based on GL Calendar can instead base it on Enterprise Calendar.

> **Note:** The GL date (Last Update Date Time Stamp) is the canonical date for this table and is also used for global exchange rate calculation.

15.2.3.15 About Configuring Project Retention Fact

The Retention fact table (W_PROJ_RETENTION_F) holds aggregated information about Retention Billing Metrics. Out of the box, the logical table source based on this table in the metadata repository (RPD) is disabled. If you want to use this table, you need to use the Oracle BI Administration Tool to open the RPD and activate it. In the Administration Tool, access the Fact_W_PROJ_RETENTION_F_Project_Retention LTS, which is located under Fact - Project Billing in the Business Model and Mapping layer, and clear the Disabled option.

The corresponding ETL tasks have not been deactivated out of the box in DAC. Therefore, if you do not use this fact table, access the PeopleSoft 8.9 and PeopleSoft 9.0 containers in DAC and deactivate these tasks:

- SDE_PSFT_ProjectRetentionFact
- SDE_PSFT_ProjectRetentionFact_Primary

Also, in DAC, access the DataWarehouse container and deactivate these tasks:

- SIL_ProjectRetention_Fact
- SIL_ProjectRetentionFact_IdentifyDelete
- SIL_ProjectRetentionFact_SoftDelete

15.2.4 Configuration Steps for Project Analytics for Universal

This section contains the following topics:

- Section 15.2.4.1, "What is Universal Adapter for Project Analytics?"
- Section 15.2.4.2, "Data Format Requirements"
- Section 15.2.4.3, "Sample Source Files for Universal Adapter for Project Analytics"
- Section 15.2.4.4, "Configuring CSV Files for the Code Dimension"
- Section 15.2.4.5, "How to Create Data Files for Universal Adapter for Project Analytics"

15.2.4.1 What is Universal Adapter for Project Analytics?

Universal Adapter for Project Analytics enables you to extract data from Project transactional applications for analysis, when pre-packaged business adapters for these sources are not available. Unlike the pre-packaged business adapters that can only be used for particular source systems, Universal Adapter for Project Analytics can be used for any source system as long as the source data can be presented in the specified flat file format.

Project source data could also reside in custom applications or legacy systems.

Universal Adapter for Project Analytics is shipped as part of any of the Oracle Project Analytics application.

15.2.4.2 Data Format Requirements

To use Universal Adapter for Project Analytics, present source data in flat files according to the following specifications:

- Data should be in comma delimited format (*.csv files).
- All date values should be in the format of YYYYMMDDHH24MISS. For example, 20071231140300 should be used for December 31, 2007, 2:03 pm.
- If a string data value contains one or more commas, then it should be double quoted.
- Data should start from the 6th line of each file. The first five lines of each file are skipped during the ETL process.

15.2.4.3 Sample Source Files for Universal Adapter for Project Analytics

Oracle Business Intelligence Applications provides a sample source file for each supported source file type. Typically, each sample source file contains sample data rows. Sample source files are located in *MW_HOME*\biapps\dwrep\Informatica\Sample Universal Source Files.

The sample source files for Universal Adapter for Project Analytics are:

- FILE_BUDGET.csv
- FILE_EVENT_DS.csv
- FILE_FINANCIAL_RESOURCE_DS.csv

- FILE_MFG_RESOURCE_DS.csv
- FILE_PROJ_BUDGET_FS.csv
- FILE_PROJ_CLASSIFICATION_DS.csv
- FILE_PROJ_CONTRACT_DS.csv
- FILE_PROJ_COST_LINE_FS.csv
- FILE_PROJ_EXP_LINE_FS.csv
- FILE_PROJ_FORECAST_FS.csv
- FILE_PROJ_FUNDING_HDR_FS.csv
- FILE_PROJ_FUNDING_LINE_FS.csv
- FILE_PROJ_INVOICE_LINE_FS.csv
- FILE_PROJ_RETENTION_FS.csv
- FILE_PROJ_REVENUE_LINE_FS.csv
- FILE_PROJECT_DS.csv
- FILE_TASK_DS.csv
- FILE_WORK_TYPE_DS.csv

15.2.4.4 Configuring CSV Files for the Code Dimension

In addition to the CSV files listed in Section 15.2.4.3, "Sample Source Files for Universal Adapter for Project Analytics," the following CSV files also need to be configured so that the code dimension is loaded properly and the code – name pairs are properly resolved in the dimension tables:

- file_codes_project_bill_hold.csv
- file_codes_project_budget_changereason.csv
- file_codes_project_budget_class.csv
- file_codes_project_budget_entrylevel.csv
- file_codes_project_budget_status.csv
- file_codes_project_budget_timephasedtype.csv
- file_codes_project_budget_versiontype.csv
- file_codes_project_budgettype.csv
- file_codes_project_contract_type.csv
- file_codes_project_cost_code_type.csv
- file_codes_project_event_class.csv
- file_codes_project_event_source_code.csv
- file_codes_project_expenditure_category.csv
- file_codes_project_expenditure_class.csv
- file_codes_project_finplantype.csv
- file_codes_project_funding_category.csv
- file_codes_project_fundingapproval.csv
- file_codes_project_invoice_class.csv

- file_codes_project_invoice_line_type.csv
- file_codes_project_invoice_status.csv
- file_codes_project_priority_code.csv
- file_codes_project_projecttype.csv
- file_codes_project_resource_class.csv
- file_codes_project_resource_type.csv
- file_codes_project_resource_type_code.csv
- file_codes_project_revenue_category.csv
- file_codes_project_revenue_distributed.csv
- file_codes_project_revenue_status.csv
- file_codes_project_role.csv
- file_codes_project_securitylevel.csv
- file_codes_project_service_type.csv
- file_codes_project_status.csv
- file_codes_project_task_priority_code.csv
- file_codes_project_task_status.csv
- file_codes_project_task_type.csv
- file_codes_project_transfer_status.csv
- file_codes_project_type_class.csv
- file_codes_project_uom.csv

As a general rule, use the default value 0 for numeric columns and 'N/A' or 'Unspecified' for string columns so that you do not encounter 'Not Null' errors when running the ETLs. Date fields can be null as well.

The dimension ID fields in the fact staging tables have to be populated with the integration_id of the various dimensions. This is important, otherwise the dimension wids fields in the fact tables will default to 0.

Use the following SQL to retrieve the foreign key information for project facts from the DAC metadata. In addition, refer to the *Oracle Business Analytics Warehouse Date Model Reference* for the star schema diagrams and other foreign key information:

```
SELECT A.NAME TBL, B.NAME COL, AA.NAME FK_TBL, BB.NAME FK_COL
FROM W_ETL_TABLE A, W_ETL_TABLE_COL B,W_ETL_TABLE AA, W_ETL_TABLE_COL BB
WHERE A.ROW_WID = B.TABLE_WID
AND B.FK_TABLE_WID = AA.ROW_WID
AND B.FK_COL_WID = BB.ROW_WID
AND A.NAME LIKE 'W_PROJ%F'
AND B.NAME LIKE '%WID'
AND B.NAME NOT IN ('ETL_PROC_WID')
ORDER BY 1,2;
```

Similarly, the common dimensions that Projects uses such as W_INT_ORG_D, W_MCAL_DAY_D, W_MCAL_CONTEXT_G, W_EMPLOYE_D, W_JOB_D, W_INVENTORY_PRODUCT_D, and so forth also need to be populated correctly from the source files.

W_MCAL_CONTEXT_G has a class field that holds two values: GL or PROJECTS. To resolve the project accounting dates in the fact tables, there must be data present in this table for class 'PROJECTS.'

15.2.4.5 How to Create Data Files for Universal Adapter for Project Analytics

To create data files, Oracle recommends that you use the sample source files that are provided. For a list of sample source files, see Section 15.2.4.3, "Sample Source Files for Universal Adapter for Project Analytics."

To create data files for Universal Adapter for Project Analytics:

1. Copy the sample source files (file_xxx.csv) from the *MW_HOME*\biapps\dwrep\Informatica\Sample Universal Source Files folder to the $PMServer\SrcFiles directory (for example, *INFA_HOME*\server\infa_shared\SrcFiles).
2. Delete the sample data rows from every sample source file.

   You must first have all of the empty files in the *INFA_HOME*\server\infa_shared\SrcFiles directory for the ETL Execution Plans to run without interruptions. Some ETL Execution Plans might contain tasks that you do not need, but would fail and cause interruption in your ETL run if source files expected by the tasks cannot be found in the *INFA_HOME*\server\infa_shared\SrcFiles directory.
3. Populate the files that you need in the *INFA_HOME*\server\infa_shared\SrcFiles directory, making sure that the data starts at line six.

   For example, to load budget data, you might populate the file_budget.csv file.
4. In DAC, create an Execution Plan to load the data from these files.

15.3 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle Project Analytics. It contains the following topics:

- Section 15.3.1, "Configuration Steps for Oracle Project Analytics for All Source Systems"
- Section 15.3.2, "Configuration Steps for Oracle Project Analytics for PeopleSoft"

15.3.1 Configuration Steps for Oracle Project Analytics for All Source Systems

This section contains the following topics:

- Section 15.3.1.1, "About Configuring Multi Calendar Support in Oracle Project Analytics"
- Section 15.3.1.2, "Configuring Multi Currency Support in Oracle Project Analytics"

15.3.1.1 About Configuring Multi Calendar Support in Oracle Project Analytics

Oracle Project Analytics supports both the GL Calendar and Project Calendar. Most facts have two separate keys: one that points to the GL Calendar Date and another that points to the Project Calendar Date. In Oracle EBS tables, the facts are typically sourced from GL_DATE and PA_DATE respectively.

> **Note:** In PeopleSoft, GL Calendar and Project Calendar are the same.

The Canonical Date is the GL Date for all facts. Because GL_ACCOUNTING_DT_WID is pointing to the multi calendar dimension (W_MCAL_DAY_D) table, and reports cannot combine dates from different calendars, all dashboards have to filter on the Project OU to have correct data in the Ledger Calendar.

The same thing applies to analysis by Project Dates because PROJ_ACCOUNTING_DT_WID points to the same table (W_MCAL_DAY_D). However, it uses the Project Calendar instead of the GL Calendar.

Users can use the Enterprise Calendar to view data across Operating Units in the same screen. Enterprise dates point to W_DAY_D, which is the same across all Operating Units in the data warehouse.

If a user wants to build a report showing all data for an Expenditure Operating Unit, this report must use the Enterprise Date fields instead of the GL or Project Date fields. This is because transactions owned by one Expenditure OU can reference multiple projects (in case of Cross Charge), and thus multiple Project Calendars. To view data across Operating Units in the same screen, users can use the Enterprise Calendar.

For information about how to configure fiscal calendars, see Chapter 3, "Configuring Common Areas and Dimensions."

15.3.1.2 Configuring Multi Currency Support in Oracle Project Analytics

All metrics are supported in multiple currencies in the dashboard. The supported currency types include:

- Transaction (Document) Currency
- Project Functional (Local) Currency
- Project Currency
- Three global currencies

Note: By default, Oracle BI Answers displays all metrics in local currency. Some metrics, such as Unearned Revenue, Unbilled Receivables, Realized Gains, and Realized Losses are only available in local currency.

Out of the box, every Project dashboard has a prompt called Currency Type. The values in this prompt are Local Currency, Project Currency, Global1 Currency, Global2 Currency, and Global3 Currency. When the user chooses one of these values and clicks the Go button for that prompt, all monetary values in the page are converted to that particular currency, and the Currency Code field in the report shows the currency code corresponding to the selected currency type.

Note: Some metrics, such as Unearned Revenue, are not available in the Project Currency.

The user can add the same functionality in custom reports by adding a similar prompt. For more information on extending this feature, see support documentation about dynamic reporting on multiple currencies on the My Oracle Support Web site.

To configure multi currency support in Oracle Project Analytics:

1. Using the Administration Tool, open OracleBIAnalyticsApps.rpd.

   The OracleBIAnalyticsApps.rpd file is located at:

```
ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_
obisn\repository
```

2. Click the Manage menu, Variables, Repository, and then Static.
3. Change the values of the GLOBAL1_CURR_CODE, GLOBAL2_CURR_CODE, and GLOBAL3_CURR_CODE variables to the global currencies configured in DAC.

   Out of the box, the currencies are set to USD. For information about how to configure global currencies, see Chapter 3, "Configuring Common Areas and Dimensions."

15.3.2 Configuration Steps for Oracle Project Analytics for PeopleSoft

This section contains the following topic:

- Section 15.3.2.1, "About Configuring Budget Fact for PeopleSoft"

15.3.2.1 About Configuring Budget Fact for PeopleSoft

Baselined Budgets are extracted into the Budget Fact (W_PROJ_BUDGET_F) table, the grain of which is Budget Line.

Defining Filters on Budget Metrics A user can create multiple versions for the same Budget type. Therefore, all exposed metrics are filtered by this filter: Current or Original Budget. One Project budget can have multiple versions. To show only one budget version at a time, every metric for the Current Version and the Original Version are shown. These flags are set based on the version number of Budget. If the budget has only one version, the Current and Original budget would show the same information.

15.4 Oracle Project Analytics Integrations

This section contains the following topics:

- Section 15.4.2, "Integrations with Project Analytics for PeopleSoft"
- Section 15.4.1, "Integrations with Project Analytics for Oracle EBS"

15.4.1 Integrations with Project Analytics for Oracle EBS

You can configure other analytics modules for Oracle EBS to use dimensions from Project Analytics for Oracle EBS. This section explains this integration and contains the following topics:

- Section 15.4.1.1, "About the Integration with Procurement and Spend Analytics for Oracle EBS"
- Section 15.4.1.2, "About the Integration with Financial Analytics for Oracle EBS"

15.4.1.1 About the Integration with Procurement and Spend Analytics for Oracle EBS

Oracle Project Analytics comes with an out-of-the box integration with Oracle Procurement and Spend Analytics. Leveraging this information, you can analyze the Project Commitments and historical procurement transactions (such as Requisitions from Projects and Purchase Orders associated with that) for a project. However, to leverage the commitments information in Project Analytics, you need to license and implement Oracle Procurement and Spend Analytics.

You must configure both Procurement and Spend Analytics and Project Analytics modules for Oracle EBS before you can configure this integration. For instructions on how to configure the integration between these two modules, see Section 4.3.2.2, "How to Enable Project Analytics Integration with Procurement and Spend Subject Areas."

> **Note:** Ensure that you have upgraded the Oracle Business Analytics Warehouse to Version 7.9.6 or later before you configure an integration. For more information, see *Oracle Business Intelligence Applications Upgrade Guide for Informatica PowerCenter Users*.

15.4.1.2 About the Integration with Financial Analytics for Oracle EBS

Oracle Project Analytics provides an out-of-the-box integration with Oracle Financial Analytics. Leveraging this information, Project users can get details of both AR invoices and AP invoices at a summary level for each project in multiple buckets in order to understand the cash flow. The users can then drill down to the receivable or payables information to view the specific customer invoices or suppler invoices causing the cash flow issues, and take remedial actions before they occur. Project Invoices functionality is currently part of Oracle Financial Analytics. You need to license Oracle Financials Analytics and implement AP and AR modules to leverage this functionality.

You must configure both Financial Analytics and Project Analytics modules for Oracle EBS before you can configure this integration. For instructions on how to configure the integration between these two modules, see Section 5.3.2.10, "How to Enable Project Analytics Integration with Financial Subject Areas."

> **Note:** Ensure that you have upgraded the Oracle Business Analytics Warehouse to Version 7.9.6 or later before you configure an integration. For more information, see *Oracle Business Intelligence Applications Upgrade Guide for Informatica PowerCenter Users*.

15.4.2 Integrations with Project Analytics for PeopleSoft

You can configure other analytics modules for PeopleSoft to use dimensions from Project Analytics for PeopleSoft. This section explains this integration and contains the following topics:

- Section 15.4.2.1, "About the Integration with Procurement and Spend for PeopleSoft"
- Section 15.4.2.2, "About the Integration with Financial Analytics for PeopleSoft"

15.4.2.1 About the Integration with Procurement and Spend for PeopleSoft

Oracle Project Analytics for PeopleSoft comes with out-of-the box integration with Procurement and Spend Analytics for PeopleSoft. To perform this integration, you need to license and implement Oracle Procurement and Spend Analytics.

You must configure both Project Analytics and Procurement and Spend Analytics modules for PeopleSoft before you can configure this integration. For instructions on how to configure the integration between these two modules, see Section 4.3.3.1, "How to Enable Project Analytics Integration with Procurement and Spend Subject Areas."

Note: Ensure that you have upgraded the Oracle Business Analytics Warehouse to Version 7.9.6 or later before you configure an integration. For more information, see *Oracle Business Intelligence Applications Upgrade Guide for Informatica PowerCenter Users*.

15.4.2.2 About the Integration with Financial Analytics for PeopleSoft

Financial Analytics for PeopleSoft can use dimensions from Project Analytics for PeopleSoft. To perform this integration, you need to license and implement Oracle Financial Analytics.

You must configure both Project Analytics and Financial Analytics modules for PeopleSoft before you can configure this integration. For instructions on how to configure the integration between these two modules, see Section 5.3.3.5, "How to Enable Project Analytics Integration with Financial Subject Areas."

Note: Ensure that you have upgraded the Oracle Business Analytics Warehouse to Version 7.9.6 or later before you configure an integration. For more information, see *Oracle Business Intelligence Applications Upgrade Guide for Informatica PowerCenter Users*.

Part IV

Configuring the Oracle BI Repository

Part IV contains instructions for configuring the Oracle BI Repository. It includes the following section:

- Chapter 16, "Configuring the Oracle BI Repository"

16 Configuring the Oracle BI Repository

This section describes how to configure the Oracle BI Repository for use with Oracle BI Applications. It contains the following topics:

Note: When you open OracleBIAnalyticsApps.rpd in the Oracle BI Administration Tool, you need to enter the repository password. See *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users* for more information.

16.1 How to Set 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 most 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 must 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 16–1 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 16–1 Invoice Fact Table Example

Date	Invoice Amount	Fulfill Amount	Payment Amount
January	4000	5000	4500

To implement date-specific metrics:

1. Using the Administration Tool, open OracleBIAnalyticsApps.rpd.

   The OracleBIAnalyticsApps.rpd file is located at:

   ```
   ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_
   obisn\repository
   ```

2. In the Physical layer, right-click Oracle Data Warehouse, 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.

16.2 How to Set Up Additional Time Series Metrics for Oracle Business Analytics Warehouse

The Oracle BI Repository provides a framework to add AGO metrics. The Oracle Business Analytics Warehouse is preconfigured with premapped AGO metrics; however, you can map other metrics by using the following procedure.

To set up additional time series metrics:

1. Using the Administration Tool, open OracleBIAnalyticsApps.rpd.

   The OracleBIAnalyticsApps.rpd file is located at:

   ```
   ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_
   obisn\repository
   ```

2. In the Physical layer, right-click the Oracle Data Warehouse object, and create a new Ago physical alias table.
3. In the Physical layer, create additional tables for each 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 Ago Key.
5. Map the 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 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.

16.3 How to Set Up Additional Dimension Tables for Oracle Business Analytics Warehouse

The Oracle Business Analytics Warehouse is preconfigured to map dimension tables required for analysis. The Physical layer in the Oracle BI Repository provides several other dimension table keys that can be used for certain specific analysis. If you need to set up any of the additional dimension tables in 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 Administration Tool, open OracleBIAnalyticsApps.rpd.

   The OracleBIAnalyticsApps.rpd file is located at:

   ```
   ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_
   obisn\repository
   ```
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.

16.4 How to Set Up Product Category Hierarchies

In Oracle Business Intelligence Applications, the following three hierarchies are supplied by default:

- General Hierarchy
- Purchasing Hierarchy
- UNSPSC Hierarchy

You can optionally choose to configure additional hierarchies. The following procedure shows an example of how to do this, based on one of the purchasing categories configured by default.

Note that you can use steps similar to the ones in this procedure to add any type of hierarchy to the RPD.

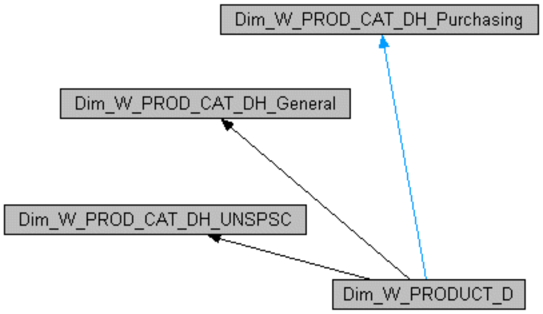
To configure product category hierarchies in the RPD file:

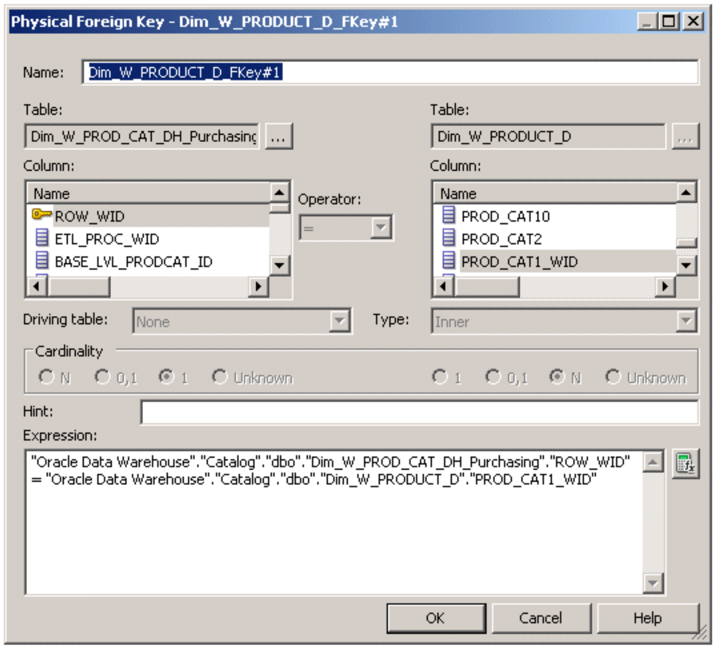
1. Using the Administration Tool, open OracleBIAnalyticsApps.rpd.

   The OracleBIAnalyticsApps.rpd file is located at:

   ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_obis*n*\repository

2. For the purchasing category example, create a link with the W_PRODUCT_D table's PROD_CATn_WID column.




Physical Foreign Key - Dim_W_PRODUCT_D_FKey#1

Name: Dim_W_PRODUCT_D_FKey#1

Table: Dim_W_PROD_CAT_DH_Purchasing

Column: ROW_WID, ETL_PROC_WID, BASE_LVL_PRODCAT_ID

Operator: =

Table: Dim_W_PRODUCT_D

Column: PROD_CAT10, PROD_CAT2, PROD_CAT1_WID

Driving table: None

Type: Inner

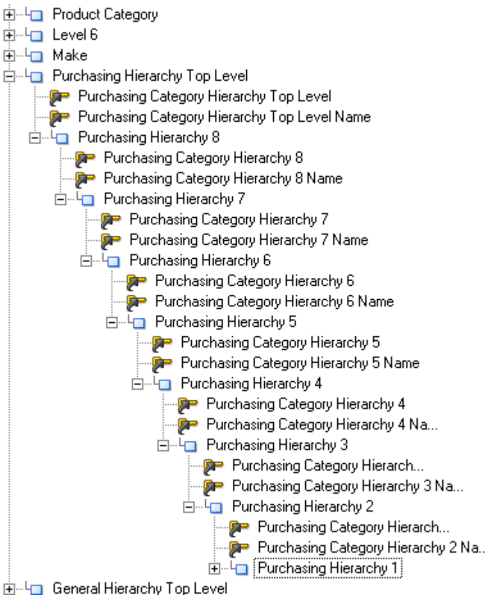
Cardinality: ○ N ○ 0,1 ● 1 ○ Unknown — ○ 1 ○ 0,1 ● N ○ Unknown

Hint:

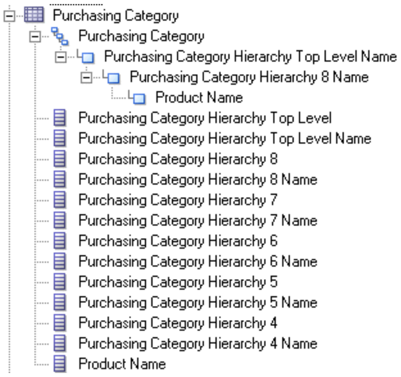
Expression:

```
"Oracle Data Warehouse"."Catalog"."dbo"."Dim_W_PROD_CAT_DH_Purchasing"."ROW_WID"
= "Oracle Data Warehouse"."Catalog"."dbo"."Dim_W_PRODUCT_D"."PROD_CAT1_WID"
```

3. Create the new dimension and dimension hierarchy in the Business Model and Mapping layer.


```
Product Category
Level 6
Make
Purchasing Hierarchy Top Level
    Purchasing Category Hierarchy Top Level
    Purchasing Category Hierarchy Top Level Name
    Purchasing Hierarchy 8
        Purchasing Category Hierarchy 8
        Purchasing Category Hierarchy 8 Name
        Purchasing Hierarchy 7
            Purchasing Category Hierarchy 7
            Purchasing Category Hierarchy 7 Name
            Purchasing Hierarchy 6
                Purchasing Category Hierarchy 6
                Purchasing Category Hierarchy 6 Name
                Purchasing Hierarchy 5
                    Purchasing Category Hierarchy 5
                    Purchasing Category Hierarchy 5 Name
                    Purchasing Hierarchy 4
                        Purchasing Category Hierarchy 4
                        Purchasing Category Hierarchy 4 Na...
                        Purchasing Hierarchy 3
                            Purchasing Category Hierarch...
                            Purchasing Category Hierarchy 3 Na...
                            Purchasing Hierarchy 2
                                Purchasing Category Hierarch...
                                Purchasing Category Hierarchy 2 Na..
                                Purchasing Hierarchy 1
General Hierarchy Top Level
```

4. Add the new columns to the Presentation layer.



- Purchasing Category
  - Purchasing Category
    - Purchasing Category Hierarchy Top Level Name
      - Purchasing Category Hierarchy 8 Name
        - Product Name
  - Purchasing Category Hierarchy Top Level
  - Purchasing Category Hierarchy Top Level Name
  - Purchasing Category Hierarchy 8
  - Purchasing Category Hierarchy 8 Name
  - Purchasing Category Hierarchy 7
  - Purchasing Category Hierarchy 7 Name
  - Purchasing Category Hierarchy 6
  - Purchasing Category Hierarchy 6 Name
  - Purchasing Category Hierarchy 5
  - Purchasing Category Hierarchy 5 Name
  - Purchasing Category Hierarchy 4
  - Purchasing Category Hierarchy 4 Name
  - Product Name

For information about configuring Product Category Hierarchies in DAC, see Section 3.2.1.1, "Configuration of Product Hierarchy (Except for GL, HR Modules)."

16.5 About the Ago Keys for Oracle Business Analytics Warehouse

The Ago Key fields are used to set up the time series metrics like Year Ago, Quarter Ago, and so on. The 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 Ago metrics. The joins in Oracle Business Analytics Warehouse uses the Ago fields in the W_DAY_D table.

16.6 About Oracle BI Time Repository Variables

The Oracle BI Repository is preconfigured with variables that are used for both reporting and internal usage.

Table 16–2 lists some example Oracle BI repository date variables and their descriptions. For a full list of variables, in the Oracle BI Administration Tool, choose Manage, then Variables, to display the Variable Manager, and refer to the Description fields for a brief description.

Table 16–2 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.

Table 16–2 (Cont.) Oracle BI Repository Date Variables

Variable Name	Description
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.
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.

16.7 About User Authentication

You need to configure the user authentication of the Oracle Business Analytics Warehouse, depending on your requirements. For more information about configuring user authentication, see *Oracle Business Intelligence Applications Security Guide* and *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

16.8 About the Security or Visibility Configuration

Oracle BI Applications includes a set of application roles that are used to control the visibility of subject areas in the Presentation layer. For more information about application roles in Oracle BI Applications, see *Oracle Business Intelligence Applications Security Guide*.

Table 16–3 lists some example application roles. You can view a full list of application roles in the policy store, or open the RPD file in the Oracle BI Administration Tool in online mode and select Manage, then select Identity.

Table 16–3 Application Roles

Application Role	Description
Administrators	The Administrators application role has all rights and privileges. It cannot be removed.
Agent Scorecard User	This application role is able to view Agent Scorecard application content.
AP Analyst	This application role is able to view application content for Oracle Payables Analytics.
AP Manager	This application role is able to view high-level application content for Oracle Payables Analytics.
AR Analyst	This application role is able to view application content for Oracle Receivables Analytics.
AR Manager	This application role is able to view high-level application content for Oracle Receivables Analytics.
CFO	This application role is able to view most of the Oracle Financial Analytics application content.
Contact Center and Agent Performance Analyst	This application role is able to view Contact Center Telephony Analytics and Agent Performance application content.
Contact Center and Agent Performance User	This application role is able to view a subset of Contact Center Telephony Analytics and Agent Performance application content.
Contact Center Sales Analyst	This application role is able to view Contact Center Telephony Analytics and Order Management Analytics application content.
Contact Center Sales User	This application role is able to view a subset of Contact Center Telephony Analytics and Order Management Analytics application content.
Controller	This application role is able to view application content for Oracle General Ledger and Profitability Analytics and Siebel Profitability Analytics.
Customer Service Analyst	This application role is able to view Customer Service for Oracle Contact Center Telephony Analytics application content.
Customer Service User	This application role is able to view a subset of Customer Service for Oracle BI Contact Center Telephony Analytics application content.
Contact Center Telephony Analytics User	This application role is able to view Oracle BI Contact Center Telephony Analytics application content.
Financial Analyst	This application role is able to view Oracle Financial Analytics application content.

Table 16–3 (Cont.) Application Roles

Application Role	Description
Human Resources Analyst	This application role is able to view Oracle HR Analytics application content.
Human Resources Vice President	This application role is able to view high-level application content for Oracle HR Analytics application.
Inventory Analyst	This application role 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 application role 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 application role 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.
Sales Executive Analytics	This application role is able to view high-level application content for the Order Management Analytics application.
Sales Manager	This application role is able to view most of the high-level application content for Oracle BI Contact Center Telephony Analytics application.
Sales Manager Analytics	This application role is able to view most of the high-level application content for Oracle BI Contact Center Telephony Analytics application.
Sales Operations Analytics	This application role is able to view operational application content for Oracle BI Contact Center Telephony Analytics application.
Sales Representative Analytics	This application role is able to view low-level application content for Oracle BI Contact Center Telephony Analytics application.
Sales Rev and Fulfill Analyst	This application role is able to view the content for Oracle BI Contact Center Telephony Analytics Revenue and Fulfillment application.
Sales Rev and Fulfill Exec	This application role 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 application role 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 application role is able to view low-level application content for Order Management Analytics Revenue and Fulfillment application.
Sales Revenue Analyst	This application role is able to view the content for Oracle BI Contact Center Telephony Analytics Revenue application.
Sales Revenue Exec	This application role is able to view the high-level application content for Oracle BI Contact Center Telephony Analytics Revenue application.
Sales Revenue Mgr	This application role is able to view most of the high-level application content for Oracle BI Contact Center Telephony Analytics Revenue application.
Sales Revenue Rep	This application role is able to view low-level application content for Oracle BI Contact Center Telephony Analytics Revenue application.
Service Delivery and Costs Analyst	This application role is able to view Service Delivery and Costs for Oracle BI Contact Center Telephony Analytics application content.
Service Delivery and Costs User	This application role is able to view a subset of Service Delivery and Costs for Oracle BI Contact Center Telephony Analytics application content.

Table 16–3 (Cont.) Application Roles

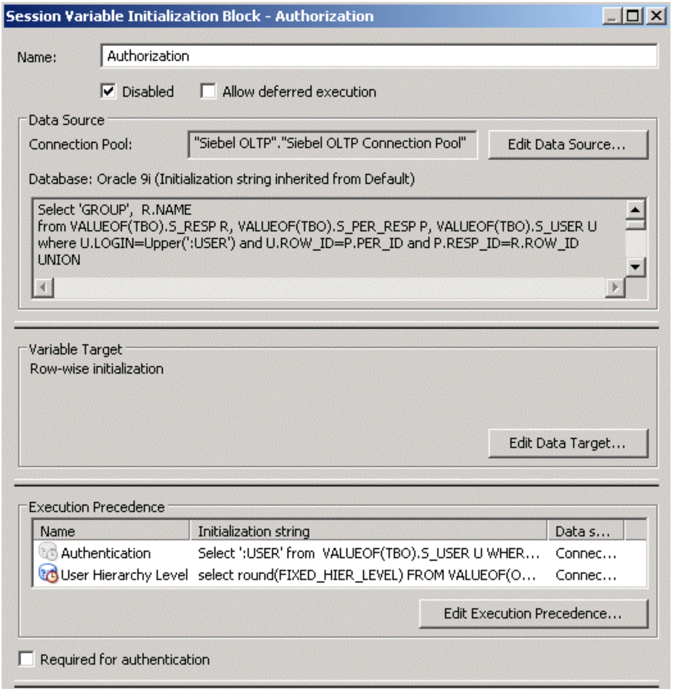
Application Role	Description
Supplier Performance Analyst	This application role is able to view Oracle's Procurement and Spend Analytics Family of Products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) content pertaining to supplier performance.
Supplier Performance Manager	This application role 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 application role 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.

16.9 About the Group Variable

The GROUP session variable in the RPD file retrieves the membership of a user in the application roles used for data and object security. Application role membership is managed in the policy store.

If you are using Oracle BI Applications with the Siebel CRM system, then you can leverage the Authorization session initialization 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, see *Oracle Business Intelligence Applications Security Guide*. Figure 16–1 shows an example of an initialization block that associates a user to a Group membership.

Figure 16–1 Initialization Block - Authorization Dialog



Name: Authorization

☑ Disabled ☐ Allow deferred execution

Data Source

Connection Pool: "Siebel OLTP"."Siebel OLTP Connection Pool"

Database: Oracle 9i (Initialization string inherited from Default)

```
Select 'GROUP',  R.NAME
from VALUEOF(TBO).S_RESP R, VALUEOF(TBO).S_PER_RESP P, VALUEOF(TBO).S_USER U
where U.LOGIN=Upper(':USER') and U.ROW_ID=P.PER_ID and P.RESP_ID=R.ROW_ID
UNION
```

Variable Target

Row-wise initialization

Execution Precedence

Name	Initialization string	Data s...
Authentication	Select ':USER' from VALUEOF(TBO).S_USER U WHER...	Connec...
User Hierarchy Level	select round(FIXED_HIER_LEVEL) FROM VALUEOF(O...	Connec...

☐ Required for authentication

16.10 About Configuring Usage Tracking for Oracle Business Analytics Warehouse

Oracle Business Analytics Warehouse supports the accumulation of usage tracking statistics. For more information about usage tracking, see *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

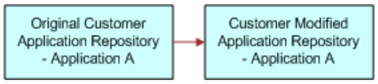
16.11 About the Incremental Deployment of the Oracle BI Applications Repository

Oracle BI Applications consists of various application families, for example, Supplier Performance Analytics, Contact Center Telephony Analytics, General Ledger and Profitability Analytics, and so on. You can purchase these applications at different times. You can customize functionality and incrementally add new applications.

This section describes the procedure for deploying multiple applications. You can repeat the procedure to add applications incrementally.

Figure 16–2 shows a single Oracle BI Applications environment. During installation, you will be asked to specify the application module(s) you have licensed, and the installer will extract the metadata corresponding to this module into one repository file. You can then modify the Oracle BI Repository to suit your business needs.

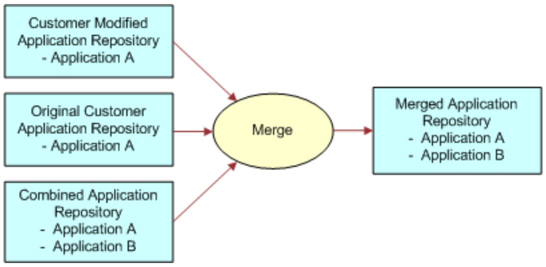
Figure 16–2 Oracle Business Analytics Warehouse Environment


Original Customer Application Repository - Application A
Customer Modified Application Repository - Application A

When you purchase another Oracle BI Applications application, you need to extract new metadata for all the modules that you have licensed. Use the merge utility in Oracle BI Administration Tool to perform a patch merge. For more information on merging repositories, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.

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 Figure 16–3.

Figure 16–3 Merging with an Oracle BI Repository


Customer Modified Application Repository - Application A
Original Customer Application Repository - Application A
Combined Application Repository
- Application A
- Application B
Merge
Merged Application Repository
- Application A
- Application B

You can repeat this merging procedure to add more Oracle BI applications to the Oracle BI Repository.

Part V

Customizing the Oracle Business Analytics Warehouse

Part V contains instructions for customizing the Oracle Business Analytics Warehouse. The tasks in this section are optional.

Part V includes the following section:

- Chapter 17, "Customizing the Oracle Business Analytics Warehouse"

17

Customizing the Oracle Business Analytics Warehouse

This section describes concepts and techniques for customizing the Oracle Business Analytics Warehouse.

Note: Before implementing customizations, you should already be familiar with Informatica PowerCenter.

This chapter includes the following main topics:

17.1 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 17.1.1, "What is Customization in Oracle Business Intelligence Applications?"
- Section 17.1.2, "About the Customization Process"

17.1.1 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.

For detailed information about tables and naming conventions, see *Oracle Business Analytics Warehouse Data Model Reference*.

17.1.2 About the Customization Process

This chapter explains how to customize your ETL 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 Informatica transformations that you need to modify.
- RPD Modification - having made the customizations in the ETL 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.

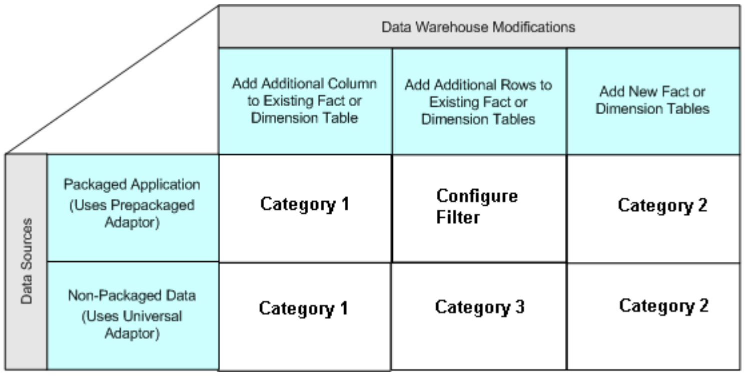
17.2 Oracle Business Analytics Warehouse Customization Scenarios

In customizing the Oracle Business Analytics Warehouse, various scenarios are available based on the type of your data source:

- Packaged applications (for example, Siebel CRM or Oracle EBS) use prepackaged adapters.
- Non-packaged data sources use the Universal adapter.

Figure 17–1 shows the categories of supported customization scenarios, based on the data source.

Figure 17–1 Supported Customizations Based on Data Source



		Data Warehouse Modifications		
		Add Additional Column to Existing Fact or Dimension Table	Add Additional Rows to Existing Fact or Dimension Tables	Add New Fact or Dimension Tables
Data Sources	Packaged Application (Uses Prepackaged Adaptor)	Category 1	Configure Filter	Category 2
	Non-Packaged Data (Uses Universal Adaptor)	Category 1	Category 3	Category 2

For detailed information about tables and naming conventions, see *Oracle Business Analytics Warehouse Data Model Reference*.

17.2.1 Types of Customizations

Figure 17–1 shows the following categories of customization:

- **Category 1.** In a Category 1 customization, you add additional columns from source systems that have prepackaged adapters and load the data into existing data warehouse tables.
- **Category 2.** In a Category 2 customization, you use prepackaged adapters to add new fact or dimension tables to the data warehouse. Category 2 customizations normally require that you build new SDE and SIL mappings.
- **Category 3.** In a Category 3 customization, you use the Universal adapter to load data from sources that do not have pre-packaged adapters.

17.2.2 Upgrade Consideration

One of the most difficult aspects about working with customizations is handling the customizations at the time of an upgrade. Informatica does not provide a 'diff-merge' capability that would automatically detect changes introduced by customers and add them into upgraded mappings. Therefore, customizations must be reapplied manually to upgraded mappings. Oracle BI Applications attempts to minimize the amount of effort required to reapply customizations after an upgrade. As long as the customization methodology is followed, the effort at upgrade time should be minimal and in many cases there may be no manual effort required at all.

17.3 Category 1 Customizations: Adding Columns to Existing Fact or Dimension Tables

This section includes the following topics:

- Section 17.3.1, "About Extending Mappings"
- Section 17.3.2, "Impact of Customization on Upgrade"
- Section 17.3.3, "Typical Steps to Extend Mappings in the Oracle Business Analytics Warehouse"
- Section 17.3.4, "Scenario for Extending the Oracle Business Analytics Warehouse"

17.3.1 About Extending Mappings

Category 1 customizations involve extracting additional columns from source systems for which pre-packaged adapters are included (for example, Oracle JD Edwards EnterpriseOne or Oracle) and loading the data into existing data 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 17.6, "Category 3 Customizations: Adding New Data as a Whole Row into a Standard Dimension Table.")

In order to see additional columns in the data warehouse, the columns must first be passed through the ETL process. The existing mappings and tables are extensible. Sample placeholders demonstrate how to pass and store additional data. Oracle BI Applications provides a methodology to extend preconfigured mappings to include these additional columns and load the data into existing tables.

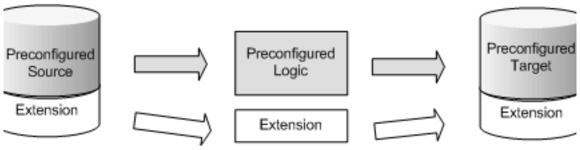
Oracle BI Applications recognizes two types of customization: extension and modification. The supported extension logic enables you to add to existing objects. For example, you can extract additional columns from a source, pass them through existing mappings, and populate new columns added to an existing table. Generally, Oracle BI 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 mapping. All extension logic should follow the same route through the mapping as X_CUSTOM. You can add additional transformations to the mapping, but they should follow the same route through the mapping as X_CUSTOM.

In the following figure, the preconfigured logic is shaded in gray. You should not modify anything contained within these objects. You should add customizations to the existing mapping, which allows them to run parallel to the existing logic.

Figure 17–2 Preconfigured Logic and Customizations



Because some objects need to be modified in order to allow for extensions, Oracle BI Applications breaks down extensions into the following categories:

- **Exposed objects.** These objects allow changes, but the changes must be in the form of extensions (additive), and must never modify existing preconfigured logic. These objects are included in the mapping when shipped and are usually sources, targets, and nonreusable transformations.

- **Encapsulated objects.** These objects cannot be extended. They attempt to hide as much of the shipped transformation logic as possible to prevent breaking the preconfigured logic. These objects are included in the mapping when shipped and are usually mapplets and reusable transformations.

- **Custom objects.** You add custom objects to a mapping. (They are not shipped by Oracle.) Custom objects can be sources, transformations (reusable and nonreusable), or mapplets. Reusable transformations and mapplets that are shipped are considered encapsulated objects, but when you add such objects to an existing mapping, they are considered custom objects to that particular mapping. For example, if you want to add another amount to a fact table and that amount must be converted from its original currency to the data warehouse currency, you would normally add the existing Currency Exchange mapplet to the mapping to convert this new amount. In this case, the mapplet is considered a custom object to this particular mapping; however, it is also encapsulated, so the internal logic must not be changed.

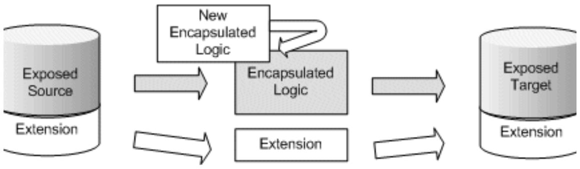
Note: Targets should not be added to a mapping.

17.3.2 Impact of Customization on Upgrade

When upgrading, you will deploy customized mappings on an individual basis. Only the actual mappings that have changed will be applied in your existing environment. This means any mappings that have not changed will not be affected, so any customizations made to these mappings remain. Only the mappings that have actually changed will require some work to reapply customizations. If you follow the recommended approach, the amount of work required to reapply customizations should be minimal.

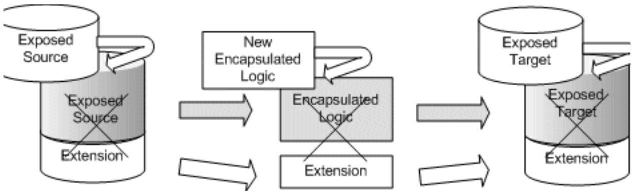
By encapsulating the logic as much as possible, any changes made to the preconfigured logic can be switched as either part of a patch release or upgrade without impacting any extension logic, as shown in the following figure.

Figure 17–3 Encapsulating Logic



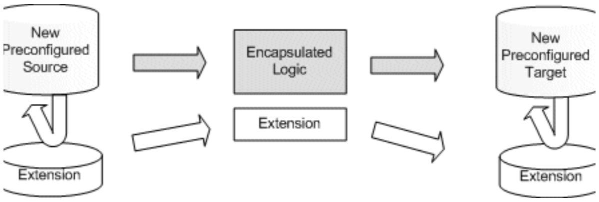
If there is a change to an exposed object, the new logic will always take precedence over the extension logic. However, rather than losing all of the extensions, much of the extension logic is retained and only has to be reapplied to the exposed objects. For example, if you add an additional column from the source and load it into the target, during an upgrade, the upgraded mapping brings additional columns from the source and loads them into the target.

Figure 17–4 Encapsulated Logic and Extension Logic



The source and target are completely replaced so any extensions to these are lost in Informatica (note that the columns will still exist in the database). However, the extension logic itself still exists after the upgrade. The source and target must be re-extended and then reconnected to the extension logic.

Figure 17–5 Re-extending and Reconnecting to Extension Logic



If you extend a mapping and the mapping…

- Does not change during the upgrade, all extensions are retained.
- Experiences changes to the encapsulated logic, all extensions are retained.
- Experiences changes to the exposed objects, extensions to these objects are lost but the underlying extension logic is retained. Extensions to exposed objects must be manually reapplied.

17.3.2.1 Points to Remember

- Encapsulated objects must never be customized unless directed by Oracle. Encapsulated objects are usually mapplets and reusable transformations.
- Exposed objects can be extended but must never be otherwise modified. Exposed objects may be completely replaced at upgrade.
- Custom objects are never changed during an upgrade.
- To minimize the work required for upgrading, try to minimize the number of changes to exposed objects by using custom objects. For example, rather than adding a table to the Source Qualifier to bring in a column from a related table, add a lookup to that table in the mapping.
- In customizing objects, you must evaluate the options and determine the best approach for your environment. If you find the custom object approach allows the ETL to run in an acceptable amount of time, then this is the preferred approach. If the custom object causes the ETL process to take too long, you may want to consider incorporating the extension into an exposed object.
- When you add custom columns to the Oracle Business Analytics Data Warehouse, you do not need to manually add change capture views. DAC automatically creates change capture views for all columns (including new columns) at runtime.

Note: Most SDE adapter folders use the concept of Business Component mapplets. These are extract mapplets that may contain relational, application, or flat file sources. The Siebel adapter folders do not use Business Component mapplets; the sources are exposed directly in the mapping. Usually, the Business Component mapplet can be treated as an exposed object and is the only mapplet object that should be modified.

17.3.3 Typical Steps to Extend Mappings in the Oracle Business Analytics Warehouse

The most common scenario for extending the data warehouse is to extract existing columns from a source and pass them through to an existing data warehouse table (either fact or dimension). This type of change generally requires extending an SIL mapping. If the data is coming from a packaged source, then you will also need to extend an appropriate SDE adapter mapping. If the data is coming from a non-packaged source, then you must use a Universal adapter mapping. (You will need to create a Universal adapter mapping if an appropriate one does not already exist).

To extend a mapping in the Oracle Business Analytics Warehouse:

1. Copy the mapping to a custom folder.
2. Extend the source and target tables by making changes to the tables in the database. You then import the source and target definitions into the custom folder (which replaces the existing definitions) or manually edit the existing definition.

   As a best practice, Oracle recommends that you rename 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 may add later to that table.
3. Extend the SDE mapping by bringing in the additional columns.
   a. Either modify the Source Qualifier (which is an exposed object) to include the column in the SQL Override or add a lookup (which is a custom object).
   b. Connect objects to the placeholder transformation 'Exp_Custom'.
   c. Connect placeholder transformations to the target table.
4. Extend the SIL mapping by bringing in the additional columns.
   a. Modify the Source Qualifier (which is an exposed object) to include the column in the SQL Override.
   b. Add the column to the Source Qualifier and pass it through the filter to the 'Exp_Custom' transformation, to the Update strategy, and to the target.
5. Copy the workflow to the custom folder.
6. Update DAC with the necessary changes.

17.3.4 Scenario for Extending the Oracle Business Analytics Warehouse

In this scenario, data is passed from an existing source to an existing data warehouse table. The company in this example has identified additional fields in a base table that need to be added to the data warehouse table W_ORG_D. The company used 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.

This scenario is applied to examples with the different source types that are supported: Prepackaged adapters for Siebel, non-Siebel prepackaged application sources, and non-packaged data.

This section includes the following topics:

- Section 17.3.4.1, "Tips for Modifying the SQL Override in a Source Qualifier"
- Section 17.3.4.2, "Example of Extracting Data from a Siebel Source"
- Section 17.3.4.3, "Including a Source Table for the Change Capture Process"
- Section 17.3.4.4, "Example of Extracting Data from a Non-Siebel Packaged Source"
- Section 17.3.4.5, "Example of Extracting Data from a Universal Source"
- Section 17.3.4.6, "Example of Loading Data into an Existing Target Table"

- Section 17.3.4.7, "Updating DAC"

17.3.4.1 Tips for Modifying the SQL Override in a Source Qualifier

- It is very important that the connected columns in the Source Qualifier appear in the same order in the SQL Override. A common mistake is to have the ports appear in the Source Qualifier in a different order than in the SQL Override.
- The column in the SELECT clause must reference the aliased name of the table if an alias is used. In the example Section 17.3.4.2, "Example of Extracting Data from a Siebel Source," the actual source for the new column LOGIN is S_CONTACT but the SELECT clause references the alias LAST_UPDATE_BY.
- Make sure you include a comma before new columns in a SELECT clause or before a new table in a FROM clause.
- A new table should always be defined using LEFT OUTER join syntax. Do not use INNER join or RIGHT OUTER join syntax, because you could lose records as a result.
- Make sure you define joins to match on a unique set of values. If you do not define a join that ensures a unique relationship, you may get a cartesian product, which changes the granularity and will result in duplicate errors downstream. If you cannot define a unique join, then you should bring the data in with a Lookup Transformation, which guarantees that at most one record will be returned.
- As a best practice, you should comment custom code you have introduced. Comments should include at least the developer's name and the date the code was added.

17.3.4.2 Example of Extracting Data from a Siebel Source

In this example, the company is using Siebel Business Analytics version 7.8 and has used the S_ORG_EXT_X.ATTRIB_04 field in the S_ORG_EXT extension table to capture data related to 'ACCOUNT_LOG'. The name of the last person to update the record is retrieved by joining to the S_CONTACT table on S_ORG_EXT.LAST_UP_BY.

Note: The Siebel adapter folders do not use Business Component mapplets; the sources are exposed directly in the mapping.

To extract data from a Siebel source:

1. Create a new folder named CUSTOM_SDE_SBL_78_Adapter.
2. Copy the SDE_OrganizationDimension mapping and workflow to this folder.
3. Edit the target definition W_ORG_DS to include the following columns:

Column Name	Data Type
X_ACCOUNT_LOG	VARCHAR2(10)
X_LAST_LOGIN	VARCHAR2(10)

Note: If the source table has been customized, it would be necessary to re-import the source table into the custom folder, replacing the existing version. For this example, the source table has not changed.

4. The extension table S_ORG_EXT_X is already joined in this mapping. Drag the ATTRIB_04 column from the source definition to the Source Qualifier. The column should appear after the X_CUSTOM column.
5. Since S_CONTACT is not already joined for the last update person, drag a copy of S_CONTACT into the mapping. (Delete the new Source Qualifier associated with this source if one exists.)
6. As a best practice, you should rename the source to indicate what it represents. In this case, rename the source to S_CONTACT_LAST_UPDATE_BY.
7. Drag the LOGIN column from the source definition to the Source Qualifier.
8. Drag ATTRIB_04 and LOGIN to the EXP_Custom expression.
9. As a best practice, rename these ports to indicate both the table and column they come from.

   In case the mapping is changed and the related exposed objects are replaced, this will make it easier to reconnect because the custom expression will not be replaced.
10. Connect the appropriate ports to the target definition
11. Edit the SQL override in the Source Qualifier.
    a. Add the ATTRIB_04 and LOGIN columns immediately after X_CUSTOM in the SELECT clause.
    b. Add the table to the FROM clause.
    c. Add the join criteria.

       The Siebel application can run on various database platforms. To maintain database independence, you should write the SQL using Informatica's join syntax, which will automatically be translated to the appropriate database syntax when run. When adding tables, be sure to follow the Informatica syntax when defining the join.

       Below is an example of the SQL with the modified code in bold font.

```
S_ADDR_ORG.LAST_UPD,
S_ORG_EXT_T.LAST_UPD,
0 AS X_CUSTOM
- Added by J.Smith on 1/10/2007
,S_ORG_EXT_X.ATTRIB_04
,LAST_UPDATE_BY.LOGIN

FROM
V_ORG_EXT S_ORG_EXT,
S_ORG_EXT BU_NAME,
...

S_ORG_EXT_T,
S_ORG_EXT_X,
S_ADDR_ORG,
…

S_MKT_SEG PRTNR_MKTSEG,
S_MKT_SEG TRGT_MKTSEG
-Added by J.Smith on 1/10/2007
,S_CONTACT LAST_UPDATE_BY

WHERE
```

```
{
V_ORG_EXT S_ORG_EXT
LEFT OUTER JOIN S_ORG_EXT_X ON
S_ORG_EXT.ROW_ID = S_ORG_EXT_X.PAR_ROW_ID
…

LEFT OUTER JOIN S_MKT_SEG TRGT_MKTSEG ON
ORG.PR_ORG_TRGT_MKT_ID = TRGT_MKTSEG.ROW_ID

- Added by J.Smith on 1/10/2007
LEFT OUTER JOIN S_CONTACT LAST_UPDATE_BY ON
S_ORG_EXT.LAST_UPD_BY = LAST_UPDATE_BY.ROW_ID
}
```

12. Save your changes.
13. Refresh and validate the session in Informatica PowerCenter Workflow Manager.

    This is necessary because it is possible that changes made to the mapping may invalidate the session.

17.3.4.3 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 mapping, it may be necessary to create an auxiliary change capture mapping so that when a row changes in the new table, the corresponding row in the main table is marked as changed. If you do not create an auxiliary process, it is possible that the new column from the new table is changed but if the base table does not experience any changes, this event will not be picked up. Keep in mind that auxiliary processing can have a negative impact on ETL performance. So, if it is not necessary to flag the main record as changed when there is a change in the related table, you may want to avoid building this mapping.

17.3.4.4 Example of Extracting Data from a Non-Siebel Packaged Source

In this example, the company is using Oracle Applications version 11.5.8 and has used the HZ_CUST_ACCOUNTS.ATTRIBUTE1 field to capture data related to 'ACCOUNT_LOG.' The name of the last person to update the record is already stored in the HZ_CUST_ACCOUNTS.LAST_UPDATE_LOGIN field. There is no need to join to any additional tables.

For Oracle Applications that run on an Oracle database, you do not need to use Informatica's SQL syntax when defining joins in a SQL Override. If you need to add additional tables, you can define the join using standard Oracle syntax.

If you add another table as a source, in addition to defining the join, you must also include the table's 'LAST_UPDATE_DATE' in the `WHERE` clause using the following syntax:

```
OR TABLE_NAME.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY
HH24:MI:SS')
)
AND
…
```

This ensures that changes to a record in that table will trigger an extract. If this were the only table to have an update and the other tables were not updated, then this change would not be detected.

> **Note:** Most SDE adapter folders use the concept of Business Component mapplets. These are extract mapplets that contain relational, application, or flat file sources. Generally, the Business Component mapplet can be treated as an exposed object and is the only mapplet object that should be modified. Keep in mind that you can modify exposed objects, but there is a risk that these changes may be lost at upgrade time.

To extract data from a non-Siebel packaged source:

1. Create a new folder named CUSTOM_SDE_ORA1158_Adapter.
2. Copy the SDE_ORA_OrganizationDimension_Customer mapping and workflow to this folder.
3. Edit the target definition W_ORG_DS to include the following columns:

Column Name	Data Type
X_ACCOUNT_LOG	VARCHAR2(10)
X_LAST_LOGIN	VARCHAR2(10)

> **Note:** If the source table has been customized, it would be necessary to re-import the source table into the custom folder, replacing the existing version. For this example, the source table has not changed.

4. Open the mapping.
5. Edit the Business Component 'mplt_BC_ORA_OrganizationDimension_Customer' by right-clicking the mapplet and selecting 'Open Mapplet.'

   Remember, the Business Component mapplets are the only mapplets you can normally edit. You should not edit any other mapplets unless directed by Oracle.
6. Drag the columns LAST_UPDATE_LOGIN and ATTRIBUTE1 to the Source Qualifier, and then drag these columns to the Mapplet Output.
7. Edit the Source Qualifier to include the new columns, as shown below:

```
SELECT
…

HZ_PARTIES.SIC_CODE
- Added by J.Smith on 1/10/2007
, HZ_CUST_ACCOUNTS.LAST_UPDATE_LOGIN
, HZ_CUST_ACCOUNTS.ATTRIBUTE1

FROM
HZ_CUST_ACCOUNTS, HZ_PARTIES
WHERE
…
```

8. Return to the mapping.
9. Add a new expression and rename it to 'X_CUSTOM.'
10. Connect the new columns from the Business Component mapplet to this expression.

11. As a best practice, you should rename these ports to indicate both the table and column they came from. If the mapping is changed and the related exposed objects are replaced, this will make it easier to reconnect, because the custom expression will not be replaced
12. Connect these columns to the appropriate columns in the target definition.
13. Save your changes.
14. Refresh and validate the session in Informatica PowerCenter Workflow Manager.

    This is necessary because it is possible that changes made to the mapping may invalidate the session.

17.3.4.5 Example of Extracting Data from a Universal Source

In this example, the company has a legacy mainframe with data they would like to incorporate into the data warehouse. This requires that data be preformatted to match the source definition for FILE_ORG. Since the preexisting source definition does not include columns for the additional data the company desires, the source definition must be modified to include these columns.

> **Note:** The Universal adapter folders do not use Business Component mapplets; the sources are exposed directly in the mapping.

To extract data from a Universal source:

1. Create a new folder named CUSTOM_SDE_Universal_Adapter.
2. Copy the SDE_Universal_OrganizationDimension mapping and workflow to this folder.
3. Edit the source definition to include the following columns:

Column Name	Data Type
ACCOUNT_LOG	String(10)
LAST_LOGIN	String(10)

4. Edit the target definition W_ORG_DS to include the following columns:

Column Name	Data Type
X_ACCOUNT_LOG	VARCHAR2(10)
X_LAST_LOGIN	VARCHAR2(10)

5. Open the mapping.
6. Drag the columns LAST_UPDATE_LOGIN and ATTRIBUTE1 to the Source Qualifier.
7. Add a new expression and rename it to 'EXP_CUSTOM.'
8. Connect the new columns from the Source Qualifier to this expression.
9. Connect these columns to the appropriate columns in the target definition.
10. Save your changes.
11. Refresh and validate the session in Informatica PowerCenter Workflow Manager.

This is necessary because it is possible that changes made to the mapping may invalidate the session.

17.3.4.6 Example of Loading Data into an Existing Target Table

Once the required data has been extracted and staged, it must be loaded into an existing target table in the data warehouse.

To load data into an existing target table in the data warehouse:

1. Create a new folder named CUSTOM_SILOS.
2. Copy the SIL_OrganizationDimension mapping and workflow to this folder.
3. Edit the source definition W_ORG_DS to include the following columns:

Column Name	Data Type
X_ACCOUNT_LOG	VARCHAR2(10)
X_LAST_LOGIN	VARCHAR2(10)

4. Edit the target definition W_ORG_D to include the following columns:

Column Name	Data Type
X_ACCOUNT_LOG	VARCHAR2(10)
X_LAST_LOGIN	VARCHAR2(10)

5. Open the mapping.
6. Drag the columns X_ACCOUNT_LOG and X_LAST_LOGIN to the Source Qualifier.
7. Drag the columns X_ACCOUNT_LOG and X_LAST_LOGIN from the Source Qualifier to the Filter.

   Normally, existing transformations should not be modified. Filters are active transformations, and it is not possible to route data around an active transformation and bring it back to the same data flow. In this case, the filter is considered an exposed object and may be modified, but any changes are at risk of being lost at upgrade time.
8. Drag the columns X_ACCOUNT_LOG and X_LAST_LOGIN from the Filter to the expression EXP_Custom. If you need to apply any transformations, you should do so in this expression.
9. Drag the columns X_ACCOUNT_LOG and X_LAST_LOGIN from the expression to the Update Strategy.

   The Update Strategy is another active transformation and is, therefore, considered an to be an exposed object, just like the Filter.
10. Connect these columns to the appropriate columns in the target definition.
11. Save your changes.
12. Refresh and validate the session in Informatica PowerCenter Workflow Manager.

    This is necessary because it is possible that changes made to the mapping may invalidate the session.

17.3.4.7 Updating DAC

After making these changes to the mappings, you must register the changes in DAC. You must include the table definition, with any additional columns or indexes, and the required changes so the tasks execute the modified sessions in the new custom folders. For information about registering data warehouse objects in DAC, see *Oracle Business Intelligence Data Warehouse Administration Console User's Guide*.

17.4 Other Types of Customizations Requiring Special Handling

This section includes the following topics:

- Section 17.4.1, "Modifying Category 2 SCD Triggers"
- Section 17.4.2, "Adding Dimensions to Existing Facts"
- Section 17.4.3, "Adding Date Dimensions to Existing Facts"
- Section 17.4.4, "Adding Currencies to an Existing Table"

17.4.1 Modifying Category 2 SCD Triggers

If you want to capture historical changes with dimensions that are configured as Category 1 dimensions, you need to modify them so that they can capture Category 2 changes. A common form of customization is to change the criteria that triggers a Category 2 change in a dimension. Most changes in a dimension are treated as Category 1 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 Category 2 change. You can extend the logic that triggers a Category 2 change by adding additional columns to the logic that tracks Category 2 changes. In addition, you can remove columns from this logic in case you do not want these types of changes to trigger a Category 2 change. Modifying the Category 2 tracking logic is one of the only exceptions to the rule that you should not make changes to shipped logic. The logic that tracks Category 2 changes is contained in exposed objects in each SIL dimension mapping that supports Category 2 changes.

There is a lookup between the Source Qualifier and the Filter. This lookup is used to determine if the record already exists in the target and, therefore, must be updated in addition to other system columns. Columns that track Category 2 changes are returned in this lookup and passed to the next expression. The columns returned by the lookup are compared with the columns passed from the staging table. If any of these columns are different, the record is flagged for a Category 2 change.

This expression contains a variable port named 'TYPE2_COLS_DIFF'. If this port is flagged as 'Y' then a Category 2 change will be triggered. If it is flagged as 'N' then a Category 1 change will be triggered.

To change the columns used to determine a Category 2 change, modify the lookup to pass any additional columns you want to be evaluated for Category 2 changes. Then, modify the variable port 'TYPE2_COLS_DIFF' to include this column when being evaluated.

For example, the SIL_BusinessLocationDimension mapping compares the following columns:

```
BUSN_LOC_NAME
CITY_NAME
POSTAL_CODE
```

If you wanted to include COUNTY as part of Category 2 logic, you would change the expression for 'TYPE2_COLS_DIFF' from the following:

```
IIF(VAR_BUSN_LOC_NAME != VAR_LKP_BUSN_LOC_NAME, 'Y',
IIF(VAR_CITY_NAME != VAR_LKP_CITY_NAME, 'Y',
IIF(VAR_POSTAL_CODE != VAR_LKP_POSTAL_CODE, 'Y', 'N')))
```

To this:

```
IIF(VAR_BUSN_LOC_NAME != VAR_LKP_BUSN_LOC_NAME, 'Y',
IIF(VAR_CITY_NAME != VAR_LKP_CITY_NAME, 'Y',
IIF(VAR_POSTAL_CODE != VAR_LKP_POSTAL_CODE, 'Y',
IIF(VAR_COUNTY != VAR_LKP_COUNTY, 'Y', 'N'))))
```

As mentioned previously, you would have to modify the lookup transformation and pass the COUNTY column and store in LKP_COUNTY, and then null-evaluate it and store it in a variable VAR_LKP_COUNTY. Also, the COUNTY column that came in from the stage table should also be null-evaluated and stored in another variable column called VAR_COUNTY.

Now that COUNTY is part of Category 2 logic, you need to remove this column from any other mappings that still assume this is a Category 1 column. The only other mapping that makes this assumption is SIL_BusinessLocationDimension_SCDUpdate. This mapping consists of the source table, followed by a Source Qualifier transformation, an Expression transformation, a Filter Transformation, and finally the target table.

To ensure that the newly added column COUNTY is treated as a Category 2 column in SIL_BusinessLocationDimension_SCDUpdate, make the following required changes:

1. Source Qualifier transformation changes:
   - Delete the COUNTY port.
   - From the SQL Override, remove the item "TARGET_TABLE.COUNTY" from the SELECT clause.
2. Expression transformation changes:
   - Delete the input-only port COUNTY.
   - Delete the corresponding variable port COUNTY_VAR.
   - Delete the corresponding output-only port COUNTY_OUT.
3. Filter transformation changes:
   - Delete the port COUNTY_OUT.

Note: The previous example illustrates a situation where you want to "add" a new column to the Category 2 set. In situations where you want to "remove" a column from the Category 2 set, you would need to do the opposite steps. For example, assume that you want to remove CITY_NAME from being a Category 2 column:

1. In the mapping SIL_BusinessLocationDimension, the TYPE2_COLS_DIFF expression would reduce to:

   ```
   IIF(VAR_BUSN_LOC_NAME != VAR_LKP_BUSN_LOC_NAME, 'Y',
   IIF(VAR_POSTAL_CODE != VAR_LKP_POSTAL_CODE, 'Y', 'N'))
   ```

2. In the mapping SIL_BusinessLocationDimension_SCDUpdate, the following changes are required:

   - Source Qualifier: Instead of removing a column, you will now add CITY_NAME to the select clause.

   - Expression Transformation: Follow other examples of Category 1 columns to add three new ports corresponding to CITY_NAME.

   - Filter Transformation: Add the new port CITY_NAME.

   - Make the appropriate connections.

17.4.2 Adding Dimensions to Existing Facts

This section covers adding a dimension (preexisting or custom) to an existing fact. It assumes you have already built the required process to populate this dimension.

This process involves extending both the fact staging table and the fact data warehouse table to include the new column. In Informatica, remember to define the tables using the Oracle database type. The staging table should be defined as a varchar2(80) field and named with in _ID suffix. The data warehouse table column should be defined as an integer and named with a _WID suffix.

The SDE fact mapping must be modified to pass through the unique identifier of the dimension key. This assumes that there is some relationship between the base table and this unique identifier. It may already be stored in the base table or stored by joining to a related table. Depending on the source system, this identifier may be based on a single column or derived from multiple columns. Table 17–1 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 17–1 Formats to Derive INTEGRATION_ID

Dimension	Foreign Key	When Source is Oracle Application	When Source is Siebel Application	When Source is Oracle's JD Edwards EnterpriseOne or JD Edwards World Application
W_AP_TERMS_D	-	TO_CHAR(TERM_ID)	Not applicable	PNPTC
W_BUSN_LOCATION_D	ASSET_LOC_WID	ASSET_LOC~' | | LOCATION_ID	Not applicable	Not applicable
W_BUSN_LOCATION_D	EMP_LOC_WID	EMP_LOC~' | | LOCATION_ID	Not applicable	Not applicable
W_BUSN_LOCATION_D	INVENTORY_LOC_WID	STORAGE_LOC' | | '~' | | ORGANIZATION_ID | | '~' | | SUBINVENTORY_CODE | | '~' | | INVENTORY_LOCATION_ID	Not applicable	Not applicable

Table 17–1 (Cont.) Formats to Derive INTEGRATION_ID

Dimension	Foreign Key	When Source is Oracle Application	When Source is Siebel Application	When Source is Oracle's JD Edwards EnterpriseOne or JD Edwards World Application
W_BUSN_LOCATION_D	PLANT_LOC_WID	'PLANT' || '~' || TO_CHAR(ORGANIZATION_ID)	Not applicable	Not applicable
W_BUSN_LOCATION_D	RECEIVING_LOC_WID	'RECIPIENT_LOC' || '~' || TO_CHAR(LOCATION_ID)	Not applicable	Not applicable
W_BUSN_LOCATION_D	STORAGE_LOC_WID	'STORAGE_LOC' || '~' || ORGANIZATION_ID || '~' || SECONDARY_INVENTORY_NAME || '~'	Not applicable	Not applicable
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	Not applicable	AN8 ||'~'|| CO
W_CUSTOMER_LOC_D	-	To get the customer location key, look up W_CUSTOMER_LOC_USE_D as below	Not applicable	AIAN8||'~'|| AICO
W_CUSTOMER_LOC_USE_D	-	TO_CHAR(SITE_USE_ID) - Get Site Use Id from HZ_CUST_ACCOUNT_ROLES	Not applicable	Not applicable
W_FREIGHT_TERMS_D	-	LOOKUP_CODE	Not applicable	Not applicable
W_GL_ACCOUNT_D	-	to_char(ccid)	Not applicable	AID||'~'|| SBL||'~'|| SBLT
W_INT_ORG_D	COMPANY_ORG_KEY	COMPANY'||'~'||TO_CHAR(SET_OF_BOOKS_ID)	S_ORG_EXT.ROW_ID	CO
W_INT_ORG_D	*_ORG_KEY	Remove any prefixes and use TO_CHAR()	S_ORG_EXT.ROW_ID	MCU

Table 17–1 (Cont.) Formats to Derive INTEGRATION_ID

<table>
<tr><th>Dimension</th><th>Foreign Key</th><th>When Source is Oracle Application</th><th>When Source is Siebel Application</th><th>When Source is Oracle’s JD Edwards EnterpriseOne or JD Edwards World Application</th></tr>
<tr><td>W_ORG_D</td><td>CUSTOMER_WID</td><td>TO_CHAR(CUSTOMER_ID) - CUSTOMER_ID is CUST_ACCOUNT_ID from HZ_CUST_ACCOUNTS</td><td>UNION OF S_ORG_EXT AND S_CONTACT. When source is S_ORG_EXT, ROW_ID is passed. When source is S_CONTACT, use ''C-'||ROW_ID - the ROW_ID is the ROW_ID from the S_PERSON (not from S_ORG_EXT). This is the new value passed to lookup the Contact Customers in W_ORG_D for resolving the ACCOUNT_WID (read as CUSTOMER_WID)</td><td>Not applicable</td></tr>
<tr><td>W_PAYMENT_METHOD_D</td><td>-</td><td>LOOKUP_CODE</td><td>Not applicable</td><td>SY||'~'||RT||'~'|| KY</td></tr>
<tr><td>W_PAYMENT_METHOD_D</td><td>-</td><td>TO_CHAR(TERM_ID)</td><td>Not applicable</td><td>SY||'~'||RT||'~'|| KY</td></tr>
<tr><td>W_PERSON_D</td><td>CUST_CONTCT_WID</td><td>TO_CHAR(PARTY_ID) - PARTY_ID from HZ_PARTY_RELATIONS</td><td>S_CONTACT.ROW_ID</td><td>Not applicable</td></tr>
<tr><td>W_PRODUCT_D</td><td>PRODUCT_WID</td><td>TO_CHAR(INVENTORY_ITEM_ID)</td><td>S_PROD_INT.ROW_ID</td><td>ITM</td></tr>
<tr><td>W_SALES_PRODUCT_D</td><td>-</td><td>TO_CHAR(INVENTORY_ITEM_ID) ||'~'||TO_CHAR(ORGANIZATION_ID)</td><td>Not applicable</td><td>Not applicable</td></tr>
</table>

If you are adding an existing dimension, the SIL mapping should be extended to include the preexisting reusable Lookup transformation to that dimension. Pass the dimension's INTEGRATION_ID through the mapping along the path identified by the X_CUSTOM column and connect it to the Lookup after the Filter transformation. Also, connect the DATASOURCE_NUM_ID to the Lookup. If the dimension is a slowly changing dimension, the fact table's standard or 'canonical' date should be passed to the lookup as well, even if the dimension has not been enabled to capture Category 2 changes.

Remember to connect the ROW_WID of the Lookup to the X_CUSTOM transformation and include logic to default this value to 0 if no record is returned from the Lookup. Pass this column on to the Update strategy, and then on to the target.

Update DAC to include the foreign key to this dimension in the fact table's definition. You should reassemble the subject Area and rebuild the Execution Plan to ensure that DAC populates this dimension table before this fact table starts to load.

17.4.3 Adding Date Dimensions to Existing Facts

If adding a date dimension to a fact table, you merely have to pass the date itself through the SDE mapping to the stage table. In the SIL mapping, pass the date along the same path as X_CUSTOM. Add the reusable expression EXP_DAY_DIMENSION_FK_RESOLUTION after the Filter. Connect the date to any input and connect the appropriate output to the EXP_Custom transformation, then on to the Update Strategy and finally to the target.

17.4.4 Adding Currencies to an Existing Table

Amounts must be converted from the original currency to the data warehouse currency. Along with the amount, you must pass the currency code if it is not already connected in the mapping. Depending on the source system, there may be more than one currency code

If the source is Siebel, there is generally only one currency type

Other sources may have several currency types. Be sure to read the section on configuring currency codes to get a better understanding of how these work.

If the SIL mapping does not already include it, add the mapplet MPLT_CURCY_CONVERSION_RATES after the Filter and connect all required input ports.

Connect the appropriate exchange rate(s) to the EXP_Custom expression. Use the appropriate exchange rate to convert the amount to the data warehouse currency. Pass the converted currency to the Update strategy then onto the target.

17.5 Category 2 Customizations: Adding Additional Tables

This section includes the following topics:

- Section 17.5.1, "About Creating New Dimension or Fact Tables"
- Section 17.5.2, "Using Custom Folders in the Oracle Business Analytics Warehouse"
- Section 17.5.3, "Creating Custom Informatica Workflows"
- Section 17.5.4, "Important Notes About Customizing the Oracle Business Analytics Warehouse"
- Section 17.5.5, "Including a Source Table for Change Capture Process"
- Section 17.5.6, "Adding a New Dimension in the Oracle Business Analytics Warehouse"
- Section 17.5.7, "Adding a New Fact Table in the Oracle Business Analytics Warehouse"
- Section 17.5.8, "Adding a New Dimension Table for a New Fact Table in the Oracle Business Analytics Warehouse"

17.5.1 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 may 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 ETL 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 may 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 data warehouse tables to maintain consistency and the ability to reference existing table structures. When you create a new table, you need to register the tables and indices in DAC. You will also have to register in DAC the new tasks for new Informatica workflows and then reassemble the appropriate Subject Area and rebuild the appropriate Execution Plan. For information about assembling Subject Areas and building Execution Plans, see the *Oracle Business Intelligence Data Warehouse Administration Console Guide*.

Note: If you are creating tables on a DB2-UDB database, make sure when you register the tables in DAC the Not Logged Initially option is enabled.

17.5.1.1 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.
- **DATASRC_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 DATASRC_NUM_ID columns as well as the following:

- **ROW_WID.** A sequence number generated during the ETL process, which is used as a unique identifier for the data warehouse.
- **ETL_PROC_WID.** Stores the ID of the ETL process information. The details of the ETL process are stored in the W_ETL_RUN_S table on the data warehouse side. This is also the Process ID on Current Run/Run History screen in DAC.

17.5.1.2 About the Oracle Business Analytics Warehouse DATASRC_NUM_ID Column

All the tables in the Oracle Business Analytics Warehouse schema have DATASRC_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.

Note: The DATASRC_NUM_ID is maintained by DAC. Ensure that each source system has a unique value assigned. It is possible to have multiple instances of the same source system (for example, a U.S.-based and a European-based Siebel transactional database both loading into the same data warehouse). The two different transactional database systems should be assigned different DATASOURCE_NUM_ID values in DAC. DAC is predefined with one entry for Siebel and the DATASOURCE_NUM_ID is assigned the value of 1. If you are going to extract from additional Siebel transactional database systems and load the data into the same data warehouse, a different DATASOURCE_NUM_ID must be assigned to each Siebel transactional database system.

17.5.2 Using Custom Folders in the Oracle Business Analytics Warehouse

If you want to make changes to the Oracle Business Analytics Warehouse, you must create a custom folder and make the changes in it. Do not change objects in any of the shipped folders unless explicitly directed by Oracle. The shipped folders and the objects within them may be overwritten in future upgrades.

The deployed repository does not include any custom folders; you must create your own. You should create a custom folder for each SDE folder you have deployed that will have customizations. These folders hold the extract mappings to various sources. You should also create a separate custom folder for customizations you want to make to the SILOS folder. Do not store customized extract and load mappings in the same folder.

The easiest way to modify an object is to copy an existing object from the shipped folder into the corresponding custom folder and to reuse existing business components, source and target definitions, transformations, mapplets, and mappings.

Note: If source tables are extended, then the source tables require manual editing in Informatica PowerCenter Designer. Do not import the tables from the database into the repository, because it changes the source table definitions for the entire Oracle Business Analytics Warehouse.

When importing new tables from the database into the custom folder, use the Oracle Business Analytics Warehouse and transactional database ODBC database connections (using the ODBC drivers supplied by your database vendor) to connect to the source and target databases.

After importing new table definitions, change the database type to Oracle in the Informatica Repository, regardless of the database platform you are using. This has no effect on the choice of a relational database. This step is very important because in Informatica if the database type of the source tables are not identical, all mappings and workflows that refer to the source tables will be invalid.

17.5.3 Creating Custom Informatica Workflows

You must create custom workflows for all customized mappings. The general requirements for creating custom workflows are as follows:

- Create the workflow such that each loads only one table. This helps to integrate the workflow with DAC.

- The workflow name should match a session name that is used inside the workflow. This helps DAC to collect certain statistics.
- The flag 'Fail parent if this task fails' should be checked for all the sessions within a workflow.
- The flag 'Fail parent if this task does not run' should be checked for all the sessions within a workflow.
- The Stop on Errors parameter should be set to 1 for all sessions within a workflow. This parameter is located in the Error Handling area of the Config Object tab in Informatica PowerCenter Designer.
- Set the appropriate source and target connection values in Informatica PowerCenter Designer.
- If the workflow is going to be used for full load command, you can choose to load in bulk mode (applicable only to Oracle and DB2-UDB databases). If the workflow is going to be used for the full load command in DAC, then in the Properties tab of Informatica PowerCenter Designer, set the Target Load type to Bulk. This requires that the target table does not have any indices during the load. DAC will drop the indices automatically; no action is required on your part.
- For all entities, such as dimension and fact tables, create two workflows, one to be used for a full load and the other to be used for an incremental load. Both workflows are based on the same mapping. The same mapping gets executed during both full and incremental loads. This provides an opportunity to tune each of these load scenarios.
- If the workflow is designed for loading a dimension in a full mode, make sure the workflow also includes a session for the unspecified row creation.
- When defining a task in DAC, you must choose the appropriate truncate option. This helps DAC to decide whether to drop and create indices on the table.
- You should not truncate target tables using the Informatica 'truncate target' option. It is especially important that DAC handle truncating tables when extracting from more than a single source system and loading into the same data warehouse. DAC will dynamically determine when tables need to be truncated. By setting the truncate option in the Informatica workflow, the table will always be truncated, limiting your ability to extract and load data from multiple sources. For example, you cannot extract data from two systems and load the data into the same staging table in parallel as the Informatica session will truncate the staging table when another session is running.
- If some sessions have to be run in serial order and if all the sessions need to be rerun upon failure of the workflow, then you should consider designing a single workflow that has sessions run in serial. If upon failure, not all of the sessions need to be run again, then consider designing separate workflows and define dependencies in DAC.
- The custom workflows can be plugged into an ETL process by registering them in DAC. All new tasks need to be registered in DAC, with the appropriate properties set. Also, you must register in DAC, source and target tables, task definitions, and dependencies.

17.5.4 Important Notes About Customizing the Oracle Business Analytics Warehouse

All custom work, unless stated otherwise, must be done in the Custom folder so that the custom work can be preserved during an Informatica Repository upgrade. Doing work on the standard folder should be avoided whenever possible. An upgrade of the Informatica Repository overrides any changes to the standard folder.

17.5.4.1 Additional Customization Instructions

- **Table definitions in Informatica.** Ensure that the SQL style is set to Oracle while importing the table definitions from external data sources. Even if the actual data source is of another database type, such as DB2 or MSSQL, it does not affect the logic of how data gets loaded.
- **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 updated again, which is a costly process. For example, the logic in the preconfigured SIL mappings looks up the destination tables based on the INTEGRATION_ID and DATASRC_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.
- **ETL process.** When using multiple sources for the data warehouse, you can decide to load from all of them at the same time or at different time frequencies using different Execution Plans.
- **Truncating target tables.** Truncating should be done through DAC. A single task has place holders for a full load, and one for an incremental load.
  - For the SDE workflows, the commands for full load and incremental load are the same. They should have the Truncate Always flag checked in DAC. For these kinds of tasks, the command for full load and incremental load are based on the same mapping.
  - For SIL workflows, the command can be different for full and incremental loads. They should have the Truncate For Full Load option checked in DAC. When a table gets truncated, the indices are automatically dropped and created after the data is loaded. The workflow associated with the full load command can have the Bulk Load option turned on for an optimized version of the mapping that quickly inserts data. Note that if there are indexes on the table, the bulk load may fail, so it is very important that the indices are registered in DAC and that you drop all of the indexes on this table during a full load if you use the bulk load option.
  - If a source requires an auxiliary task, it must be run only during incremental mode. So, for these tasks, the full load command is empty. No truncate options should be set.
- **ETL_PROC_WID.** Use the same ETL_PROC_WID in W_PARAM_G table in custom mappings. ETL_PROC_WID is a reference key to Run History in DAC. To use the same ETL_PROC_WID, copy the reusable lookup (called LKP_ETL_PROC_WID) defined in the SILOS folder. The input to the lookup is a constant (hard coded to 1).
- **DATASRC_NUM_ID.** Use a parameter to define this value in the mapping. DAC will automatically create a parameter file with the correct DATASOURCE_NUM_ID, which will be picked up by the parameter in the mapping. This enables you to make multiple copies of the same mapping when you have multiple instances of

the same transactional database type. You do not have to do any additional hard-coding other than to register the sources in DAC.

- **Creating 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 ETL will use for dimensions and facts (for example, ROW_WIDs of Dimensions and Facts, INTEGRATION_ID and DATASRC_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 OTB 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 DAC, either manually or by right-clicking on the certain table and invoking the Import Indices command.
- **Currency.** For currency-related data, populate the table with the base currency and exchange date field (in order to convert the data appropriately). The data for currency conversion should be maintained in the main data source. (The currency data is maintained by converting all the currency information into a single Base Currency code specified in DAC.)
- **Day Dimension.** For the data that is related to W_DAY_D, use the reusable transformation EXP_DAY_DIMENSION_FK_RESOLUTION. Note that this transformation will take a date as input and return the foreign key to the Day dimension as output in the appropriate format (YYYYMMDD) avoiding the need of costly join or lookup to the W_DAY_D dimension table each time for resolution. Copy the reusable transformation and use it.
- **List Of Values.** This applies in particular to Category 1 and II. The preconfigured columns that depend on a list of values have a language-dependent column and a language-independent column. Use the mapplet MPLT_LOV_TRANSLATION to populate the language-dependent and independent columns in the dimension tables. For fact tables, use MPLT_LOV_D_ROW_WID to create a new foreign key to the LOV dimension. You could also deal with translations directly in SQL overrides to improve performance.

17.5.5 Including a Source Table for Change Capture Process

This procedure applies to Siebel source tables only.

To include a source table for the change capture process:

1. Verify whether the source table is registered in DAC.

    a. If an entry is not in DAC, create a new record for the table and assign an image suffix to it.

    b. If the table is registered, make sure there is an image suffix assigned for this table.

2. If an image suffix does not exist for the source table, assign one.

    The image suffix should be three characters long. The recommended naming convention is to start with C. For example, use C1, C2, CA1, CA2, and so on.

3. Make sure the image suffix is not used by other tables by querying the tables list in DAC for that image suffix.

    DAC does not validate this information when entering data.

4. Create Image tables in the Siebel transactional database.

a. Right-click the table record in DAC, and choose Generate Change Capture Scripts.

   This will assist you in creating the image tables, necessary indices, and triggers if you plan on tracking deletes on the transactional database.

b. Execute these scripts on the transactional database if you have the appropriate permission, otherwise ask the OLTP DBA to create it.

5. For the task that is created for the extract process, set the Build Image flag to True and select the new table as an auxiliary table or primary table.

> **Note:** At the end of the change capture process, DAC creates a view on the actual source table. Use this view as the main source table in all the extract procedures. For example, if your new source table is S_COMPENSATION, then the default view name is V_COMPENSATION.

17.5.6 Adding a New Dimension in the Oracle Business Analytics Warehouse

Follow this procedure to add a new dimension in the Oracle Business Analytics Warehouse.

To add a new dimension and use it with an existing fact table:

1. Create a DDL for the new dimension based on the standard structure (with appropriate system columns). Create a staging table for this dimension.
2. Register the new source table and its staging table (if it does not already exist) in the DAC repository and associate it with the appropriate database connection.
3. Create a new custom map SDE_XYZ to populate the dimension stage. Instead of the actual source table (for example S_ABC), use the view that will be generated by the change capture process (for example V_ABC) in the SQL so that it extracts only the incremental data. Use existing reference maps as examples of how to populate the system columns. Make sure you truncate the stage table in corresponding tasks.
4. Create a new custom map SIL_XYZ to populate the new dimension from the stage table. Use the above referenced map as example for how to populate the system columns.
5. Register the new dimension table in DAC and associate it with the appropriate database connection.

   If you are planning to build a new dimension incrementally, assign an image suffix to the source table.
6. Register the workflows as tasks in DAC.
7. For SDE mapping of the dimension make sure you set the Build Image flag to True, and the Truncate Always option to True. And in the list of source tables, mark the primary/auxiliary source(s) of this dimension.
8. For SIL workflows of the dimension make sure you set only Truncate for Full Load option to True.
9. Make sure the target table of the SDE_XYZ is defined as source table for SIL_XYZ.

17.5.7 Adding a New Fact Table in the Oracle Business Analytics Warehouse

Follow this procedure to add a new fact table in the Oracle Business Analytics Warehouse.

To add a new fact table:

1. Create a DDL for the new fact based on the standard structure (with appropriate system columns). Create a staging table for this fact.
2. Register the new source table (if it does not already exist) in the DAC repository and associate it with a database connection.
3. Create the change capture tables by right-clicking and selecting Generate Change Capture Scripts. For instructions, see Section 17.5.5, "Including a Source Table for Change Capture Process."
4. Create SDE mappings to populate the custom stage table. Use the view created by change capture as the main table in the SQL so that it extracts only the incremental data. Use the reference maps (above) as examples of how to populate the system columns. Be sure to truncate the stage table in corresponding workflows.
5. Create SIL mapping to populate the custom fact table. Use reference maps as examples of how to populate the system columns.
6. Use lookups or SQL override joins to dimension tables for populating dimension foreign keys (ROW_WIDs) pointing to the existing dimension.
7. In DAC, register the target tables.
8. Create new tasks for the workflows.
9. For the SDE task, make sure you have the Build Image flag set to True, and list all the source tables that it queries from. Choose one or more tables as primary or auxiliary. For the target tables choose the staging table. Set the Truncate Always flag to True.
10. For the SIL task, list all the dimensions that will be required under source tables.

17.5.8 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 mapping (SIL), use lookups for getting foreign keys to the new dimension.
2. Use existing maps as examples.

17.6 Category 3 Customizations: Adding 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 structure. Non-system columns can include the null value.

2. Create a custom SDE mapping to load the data into the staging table in the custom folder for this purpose. The staging table must be populated with incremental data (rows that have been added or changed since the last Refresh ETL process), for performance reasons.
3. Populate the INTEGRATION_ID column with the unique identifier for the record.

   The combination of INTEGRATION_ID and DATASRC_NUM_ID is unique. When importing the data, ensure that a unique identifier for the external data source is inserted in the DATASRC_NUM_ID column. The DATASRC_NUM_ID is set to 1 for mappings that source data from the Siebel transactional database. This is a reserved value and is used in all standard mappings. For example, a value of 2 can be defined for DATASRC_NUM_ID in the custom SDE mapping. The standard SDE mappings populate the INTEGRATION_ID column of the dimension staging table (used for resolving the dimension's Siebel transactional database 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 mappings to populate the dimension target tables.
5. Modify the SDE and SIL mappings of all the related fact tables (fact tables that need to be linked to this dimension).

   The custom fact SDE mappings must populate the foreign key column of the changed dimension (using a custom map table process to convert from Siebel's row IDs to the external data source row IDs). The custom SIL mapping should be modified to use the appropriate DATASRC_NUM_ID, because the standard SIL mappings assume DATASRC_NUM_ID for the dimensions are the same as the fact table's DATASRC_NUM_ID.

   It is very important to decide when the data is going to be loaded. If it is going to be loaded along with the Siebel source data, you must be careful with how you handle failure recovery. The preconfigured workflows truncate the target staging table prior to loading. Upon failure, when the DAC Server restarts the task, all 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 Informatica workflow. Upon failure of either of the extracts, the whole workflow gets rerun. 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 workflows need not depend on the preconfigured SDE workflows and can use the Truncate Table option for failure recovery. In this case, in the DAC Design view, define a new Execution Plan in the Execution Plans tab, and define the new data source in the Database Connections child tab. Make sure the shared SIL process depends on the SDE processes from both sources.

17.7 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 data warehouse, and includes the following topics:

- Section 17.7.1, "Extracting Additional Data"
- Section 17.7.2, "Setting Up the Delimiter for Source Files"

17.7.1 Extracting Additional Data

You can configure extract mappings and mapplets 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 mapping to include data from these tables.

17.7.1.1 Extracting New Data Using an Existing Source Table

Extract mappings generally consist of a source table or Business Component, an Expression transformation, and a staging table. If you want to extract new data using the existing mapping, you have to modify the extract mapping to include the new data by performing the following tasks:

To modify an existing mapping to include new data:

1. Modify the existing Business Component to extract information from the source, and add it to an appropriate extension column.

   > **Tip:** You can perform calculation transformations in the Business Component mapplet or the source adapter mapplet in the extract mapping. However, do not use performance-expensive calculations in the extract that could tie up your source transaction system. For these types of calculations, it is recommended that you perform them in the source adapter mapplet.

2. Modify the Expression transformation to perform any necessary transformations.
3. Connect all input and output ports within the extract mapping so that the data moves from the source or Business Component to the Expression transformation and through the source adapter mapplet, and finally to the staging table's appropriate extension column.

You have to determine which type of extension column to map the data to in the staging table. After you modified the extract mapping, you would also have to modify the corresponding load mapping to make sure the extension columns that you added are connected all the way from the staging table to the warehouse table.

17.7.1.2 Extracting Data from a New Source Table

Business Components are packaged as mapplets, which reside in source-specific folders within the repository. Business Components are used to extract data from the source system. You can configure these mapplets to perform the following:

- **Extract data from a new source table**
- **Set incremental extraction logic**

The following procedure contains instructions for adding a new table to the Business Component. The procedure includes adding a new source definition, connecting the ports to the Source Qualifier, editing the Source Qualifier, connecting the ports to the Output transformation, and editing the Output transformation.

To add a new source table to an existing Business Component mapplet:

1. In Informatica PowerCenter Designer, open the applicable source system configuration folder.
2. Open Mapplet Designer tool.
3. Drag the Business Component mapplet into Mapplet Designer to view the transformations that comprise the Business Component.
4. Expand the Sources folder, and copy a source table into the mapplet by dragging and dropping the table into Mapplet Designer.
5. Connect the applicable ports from the new source definition to the Source Qualifier by clicking on the port in the new source table and dragging it to the connecting port in the Source Qualifier.
6. Double-click the Source Qualifier to open the Edit Transformations box.

   In the Ports tab, make any changes to the new ports for data type, precision, scale, or all these values, as necessary.
7. Connect the applicable ports from the Source Qualifier to the Mapplet Output transformation (MAPO).

   > **Note:** In some cases, the Business Component contains an Expression transformation between the Source Qualifier and the MAPO.

8. In the Properties tab, make changes to the SQL statement as necessary.
9. Validate and save your changes to the repository.

17.7.2 Setting Up the Delimiter for Source Files

You must ensure that the delimiter used in the CSV files is not present in the source data. Oracle Business Analytics Warehouse is preconfigured with a comma as the source file delimiter. If your data contains commas, then you need to enclose the data fields using a character that is not present in the data set. For example, common enclosing characters include single quotes and double quotes.

To set up the delimiter for source files:

1. Open the CSV file.
2. Enclose the data fields with the enclosing character that you have identified.

   You can identify an enclosing character by identifying a character that is not present in the source data. Common enclosing characters include single quotes and double quotes.
3. Save and close the CSV file.
4. Identify all the source definitions associated with the modified files.
5. Change the properties for each of these source definitions to use the enclosing character.
6. Validate and save your changes to the repository.

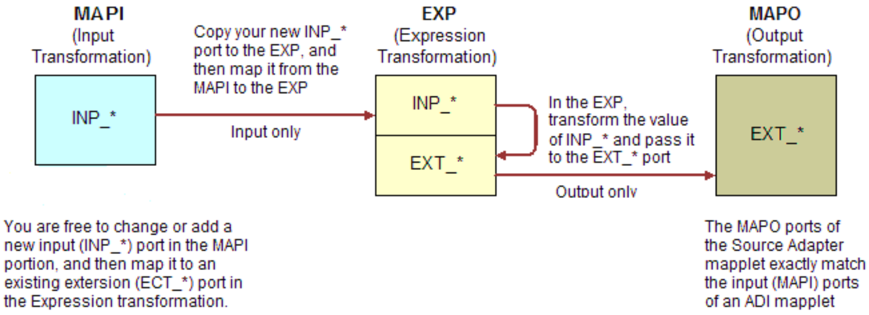
17.7.3 Configuring a Source Adapter Mapplet

The majority of all source-specific transformations occur in the source adapter mapplet; source-independent transformations generally take place in the Analytic

Data Interface (load mapping). The source adapter mapplet converts source-specific data elements into standard formats and then stores them in a staging table. The source independent loading mapping then picks up these records, which are already transformed into standard format.

Figure 17–6 illustrates the three components of the source adapter mapplet that allow transformations of data to occur. The three components are Mapplet Input (MAPI), Expression transformation (EXP), and Mapplet Output (MAPO).

Figure 17–6 Components of the Source Adapter Mapplet


MAPI
(Input Transformation)
INP_*
Copy your new INP_* port to the EXP, and then map it from the MAPI to the EXP
Input only
EXP
(Expression Transformation)
INP_*
EXT_*
In the EXP, transform the value of INP_* and pass it to the EXT_* port
Output only
MAPO
(Output Transformation)
EXT_*
You are free to change or add a new input (INP_*) port in the MAPI portion, and then map it to an existing extersion (ECT_*) port in the Expression transformation.
The MAPO ports of the Source Adapter mapplet exactly match the input (MAPI) ports of an ADI mapplet

In Figure 17–6, if the input data is transformed, the data is passed to the Expression transformation (EXP) as input only. After the data is transformed, it is output through a new port, which is prefixed with EXT_. If the data is not transformed, it comes in as input-only and leaves through an output-only port.

If you want to add a new transformation, you must add a new port to contain the expression that is used to transform the data.

To add a new port to the source adapter mapplet:

1. In Informatica PowerCenter Designer, open the applicable source system configuration folder.
2. Open the applicable source adapter mapplet.
3. Double-click the MAPI component of the mapplet, and add a new input port following the `INP_*` naming convention.
4. Copy the new input port from the MAPI to the Expression transformation.
5. Connect the new port from the MAPI to the Expression transformation.
6. In the Expression transformation, uncheck the Output indicator for the new input port; you use the value from this port in an transformation expression.
7. Perform any necessary transformations within the Expression transformation.

   The transformed data is passed out of an EXT_* output-only port.
8. Connect the port from the Expression transformation to the MAPO.
9. Validate and save your repository.

17.8 Configuring Loads

The Oracle Business Analytics Warehouse prepackages load mappings for every data warehouse table.

17.8.1 Filtering and Deleting Records

Note: This section does not apply to Oracle's Siebel sources.

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 data 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 data 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

Caution: It is important to note that delete and primary extract mappings must always be disabled together; you may not disable only one type.

17.8.2 About Primary Extract and Delete Mappings Process

Note: This section does not apply to Oracle's Siebel adapter, which does not have Primary Extract mappings.

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.

Figure 17–7 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 17–7 Extract and Load Mappings

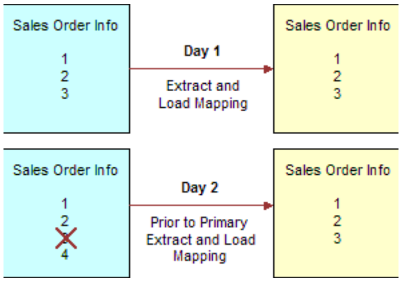
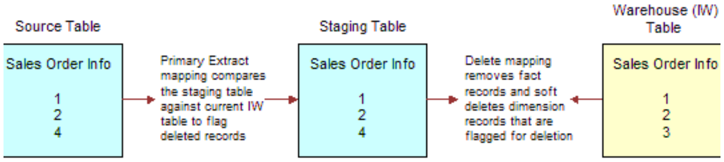


Figure 17–8 shows the primary extract and delete process that occurs when day two's information is extracted and loaded into the Oracle Business Analytics Warehouse from the source. The initial extract brings record four into the Oracle Business Analytics Warehouse. Then, using a primary extract mapping, 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 17–8. When the extract and load mappings run, the new sales order is added to the warehouse.

Figure 17–8 Primary Extract and Delete Mappings


Source Table
Sales Order Info
1
2
4
Primary Extract mapping compares the staging table against current IW table to flag deleted records
Staging Table
Sales Order Info
1
2
4
Delete mapping removes fact records and soft deletes dimension records that are flagged for deletion
Warehouse (IW) Table
Sales Order Info
1
2
3

17.8.3 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 data 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

This topic provides procedures for these management tasks.

17.8.3.1 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 17.8.3, "About Working with Primary Extract and Delete Mappings."

17.8.3.2 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.

As shown in the following procedures, there are two different ways to enable the sessions, depending on the modules you implemented.

To enable primary extract and delete sessions on Tasks:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab.
3. Query for all tasks containing the string 'Delete' or 'Primary'.
4. Deselect the Inactive check box for those tasks.
5. Reassemble your subject areas and rebuild your execution plans.

To enable primary extract and delete sessions on Configuration Tags:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Configuration Tags tab.
3. Query for all configuration tags containing the string 'Identify and Soft Delete'.
4. Select the tag of your module, and then click the subtab 'Subject Areas'.
5. Deselect the Inactive check box for the subject areas.
6. Reassemble your subject areas and rebuild your execution plans.

17.9 Configuring Slowly Changing Dimensions

The Oracle Business Analytics Warehouse provides Category 2 slowly changing dimension (SCD) functionality, which enables 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 chosen by the user after data has been extracted and transformed to be source-independent. Users may configure Oracle BI Applications to support both Category 1 SCDs, in which data is overwritten with updates, and Category 2 SCDs, in which the original records are maintained while a new record stores the updated data. Choosing Category 1 or Category 2 SCDs depends on identifying your historically significant attributes.

Most dimensions use Category 1 updates by default. If you need change a dimension to Category 2 SCD update, use the following procedure.

To turn a dimension to Category 2 SCD update enabled:

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Display the Tasks tab.
3. Query for the SIL task that populating the dimension.
4. Display the Parameters subtab, and set the value for $$TYPE2_FLG to Y.

A few dimensions are set up by default to be Category 2 SCDs in the standard repository. These dimensions are listed in Table 17–2.

Table 17–2 Dimensions Set Up as Category 2 SCDs By Default

Dimension	Adapter
W_CUSTOMER_FIN_PROFL_D	PeopleSoft
W_EMPLOYEE_D	Oracle EBS and PeopleSoft
W_INT_ORG_D	PeopleSoft
W_INVENTORY_PRODUCT_D	Oracle EBS
W_POSITION_D	All
W_PRODUCT_D	Oracle EBS, Oracle Siebel, and Oracle JD Edwards EnterpriseOne
W_Party_ORG_D	Oracle JD Edwards EnterpriseOne
W_BUSN_LOCATION_D	PeopleSoft
W_HR_POSITION_D	All
W_JOB_D	All
W_PAY_GRADE_D	All
W_PAY_TYPE_D	All
W_POSITION_DH	All

Note: If you want to turn off the TYPE2_FLG for any of the dimensions listed in Table 17–2, you must customize the source-dependent extract mapping. Make the following changes:

1. If the SQL is not yet overridden, you must override it to filter out the historical as well as the future effective records. In other words, the ultimate SQL should always fetch the latest records, as effective at the time of the session run. If the SQL is already overridden, change it to achieve the same result mentioned above. There should not be any duplicate INTEGRATION_ID values in the staging table as a result of the SQL. Although the unique index, if any, typically is on INTEGRATION_ID, DATASOURCE_NUM_ID, and SRC_EFF_FROM_DT, you should form your SQL in such a way that the uniqueness is not violated, even if SRC_EFF_FROMT_DT is kept out of the equation.

   The SQL change will vary depending on the OLTP data model and the entity in question. In some cases, you can get the record based on the MAX(EFFDT) from the source, whereas in other cases, where future data is supported, you can select only records effective at the time of the session run. The Informatica parameter $$$SessStartTime gives the system date and time of the computer where Informatica Server is running. Note that this might be different than the database hosting your OLTP data.
2. Disconnect the ports that bring the source effective from and to dates.
3. In a downstream transformation, hardcode the value of SRC_EFF_FROM_DT as 01/01/1899 (or use the Oracle BI Applications standard $$LOW_DT parameter value). The SRC_EFF_TO_DT can be left as NULL.
4. Set the value of $$TYPE2_FLG as 'N' for the task loading the table where you intend this behavior.
5. Optionally, you can disable the corresponding SCDUpdate mapping, if you want to save on overall ETL time. If you choose to disable it, you might have to rebuild your execution plan.

   Note that keeping the SCDUpdate mapping enabled does not harm the functionality.

17.9.1 About Identifying Historically Significant Attributes

You may 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 may 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 may have a load that populates the telephone number field. If your business does not perform data analysis on phone number history, then this information may be considered a *historically insignificant* attribute.

Identifying attributes as significant or insignificant enables you to determine the category of SCD you require. However, before you can select the appropriate type of SCD, you must understand their differences.

17.9.1.1 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 data warehouse. Records with changes that are important but not historically tracked are overwritten in the data warehouse, based on the primary key.

17.9.2 Category 1 and Category 2 Slowly Changing Dimensions

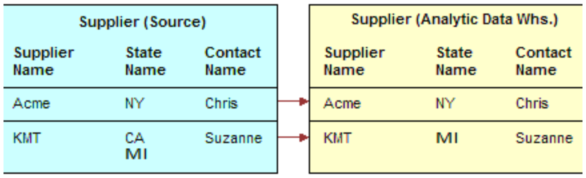
After you have correctly identified your significant and insignificant attributes, you can configure the Oracle Business Analytics Warehouse based on the type of slowly changing dimension (SCD) that best fits your needs—Category 1 or Category 2.

17.9.2.1 Category 1 Slowly Changing Dimension

A Category 1 SCD overwrites the column's value and is the default SCD for the Oracle Business Analytics Warehouse. Although a Category 1 does not maintain history, it is the simplest and fastest way to load dimension data. Category 1 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 may want to use Category 1 when changing incorrect values in a column.

In Figure 17–9, 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 data 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 17–9 An Example Type 1 Slowly Changing Dimension



Supplier (Source)

Supplier Name	State Name	Contact Name
Acme	NY	Chris
KMT	CA MI	Suzanne

Supplier (Analytic Data Whs.)

Supplier Name	State Name	Contact Name
Acme	NY	Chris
KMT	MI	Suzanne

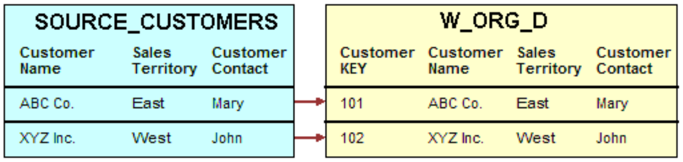
17.9.2.2 Category 2 Slowly Changing Dimension

A Category 2 SCD creates another record and leaves the old record intact. Category 2 is the most common SCD because it enables 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). Figure 17–10 shows how an extract table (SOURCE_CUSTOMERS) becomes a data 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 17–10 An Example Type 2 Slowly Changing Dimension



SOURCE_CUSTOMERS

Customer Name	Sales Territory	Customer Contact
ABC Co.	East	Mary
XYZ Inc.	West	John

W_ORG_D

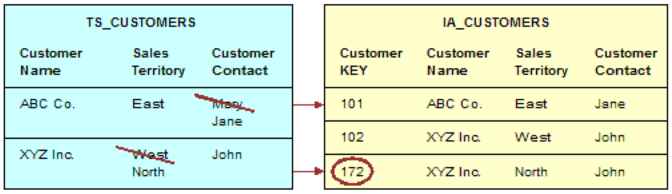
Customer KEY	Customer Name	Sales Territory	Customer Contact
101	ABC Co.	East	Mary
102	XYZ Inc.	West	John

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. Figure 17–11 shows that records for the two customers, ABC Co., and XYZ inc. have changed. 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 Category 1 SCD is applied and Mary is overwritten with Jane. Because the change in ABC's record was a Category 1 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 Category 2 slowly changing dimension is required.

As shown in Figure 17–11, 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 17–11 An Example Type 2 Slowly Changing Dimension



TS_CUSTOMERS

Customer Name	Sales Territory	Customer Contact
ABC Co.	East	~~Mary~~ Jane
XYZ Inc.	~~West~~ North	John

IA_CUSTOMERS

Customer KEY	Customer Name	Sales Territory	Customer Contact
101	ABC Co.	East	Jane
102	XYZ Inc.	West	John
172	XYZ Inc.	North	John

17.9.2.3 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 Category 2 logic creates effective dates.
- If the source supplies one of the two effective dates, then the Oracle Business Analytics Warehouse automatically populates the missing effective dates using a wrapper mapping. This situation is discussed in this section.

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 may 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, as shown in Table 17–3.

Table 17–3 Records Before a Wrapper Session in W_CUSTOMER

Customer Name	Sales Territory	Customer Contact	Effective From	Effective To	Current
ABC	East	Jane	1/1/2001	1/1/3714	Y
XYZ	West	John	1/1/2001	1/1/3714	Y
XYZ	North	John	1/1/2002	1/1/3714	Y

Note your first record for XYZ still shows as effective from January 1, 2001 to January 1, 3714, while a second record has been added for XYZ in the North territory with the new effective from date of January 1, 2002. In this second record the effective to date remains the same, January 1, 3714.

When the wrapper session executes, the effective dates for the first XYZ are corrected (January 1, 2001-January 1, 2002), and the Current Flag is adjusted in the Analytic Data Interface (load mapping) so that only the second record (January 1, 2002-January 1, 3714) is set to Y. After the wrapper session completes its work, you have Category 2 information for XYZ in your data warehouse rather than two disparate records, as shown in Table 17–4.

Table 17–4 Records After a Wrapper Session in W_CUSTOMER

Customer Name	Sales Territory	Customer Contact	Effective From	Effective To	Current
ABC	East	Jane	1/1/2001	1/1/3714	Y
XYZ	West	John	1/1/2001	1/1/2002	N
XYZ	North	John	1/1/2002	1/1/3714	Y

In the previous paragraph, the wrapper session corrected the effective to dates and current flag. However, if the record's dates had been correct, the wrapper mapping would simply have set the current flag as needed, because its logic is set to check dates and flags and only adjust columns that contain discrepancies.

17.10 About Stored Lookups

A lookup transformation enables you to specify a reference table, and then retrieve information such as code descriptions, exchange rates, and currency codes. The main types of preconfigured lookups within the Oracle Business Analytics Warehouse are:

- Section 17.11, "Codes Lookup."
- Dimension Key Lookups. For more information, see Section 17.12, "About Resolving Dimension Keys."

17.11 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 Analytic Data Interface's architecture uses components, as well as both fact and dimension tables, to facilitate lookup functionality. The following components and process are used in a lookup:

17.11.1 W_CODES Table

The load control table W_CODES consolidates all codes for future reference and assigns them a category and a single language for efficient lookup capability.

17.11.2 Codes Mappings

The Oracle Business Analytics Warehouse uses mappings designed to extract codes from source systems and populate the W_CODES table in preparation for use by the load mapping.

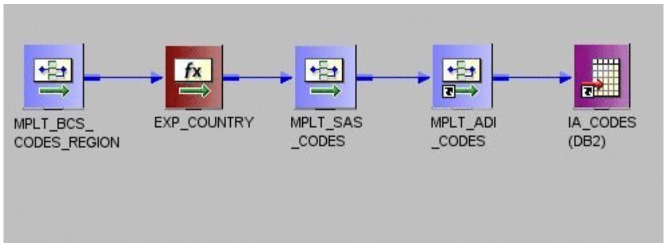
To understand how codes mappings function, it is helpful to first understand the columns within W_CODES. Table 17–5 describes these columns.

Table 17–5 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
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]. Figure 17–12 shows an example of a code mapping in Informatica PowerCenter Designer.

Figure 17–12 Example of a Code Mapping in Informatica PowerCenter Designer



17.11.3 Codes Mapplets

There are several mapplets that support the codes mappings in preparation for the source-independent load mapping. They are as follows:

- **Source adapter mapplets.** The source adapter mapplet connects the source-specific input attributes of CODES and the attributes from control or warehouse tables to the expression transform for mapping them. The naming convention for the source adapter codes mapplet is MPLT_SA_CODES.
- **Business Component mapplets.** The Business Component mapplet makes the source system attributes of CODES_CUST_CLASS available to the extract mapping. The naming convention for the Business Component codes mapplet is MPLT_BC_CODES_[CATEGORY].
- **ADI Mapplet.** The Analytic Data Interface (load mapping) mapplet is source system independent and resolves the codes for the target table. The naming convention for the load mapping codes mapplet is MPLT_ADI_CODES.

The load mapping integrates multiple source system codes by designating one source system instance as a master in a mapping. All other source system codes are then mapped to the master. When the load mapping encounters a code that requires definition, it references the load control lookup table to match the source system code to an Oracle Business Analytics Warehouse source-independent code, which retains all the source system codes' original functionality.

The following columns are used to designate a source system instance as the master source system:

- **MASTER_ID.** Code for the source system designated as master.
- **DATASOURCE_NUM_ID.** Unique identifier for the source system.

17.11.4 Configuring Extension Column Code Description Lookups

You can configure dimension and fact load sessions to perform specific lookups by editing the category of the data to be extracted from the W_CODES table and loading the code information into a target table. If the code and code name do not exist in the W_CODES table, then you must add them to the table. To configure the lookup, create a session override; do not modify the load mapping in the load mapping.

To configure sessions for lookups:

1. In Informatica PowerCenter Workflow Manager, open the applicable source system configuration folder.
2. Open the Edit Tasks box.
3. In the Transformations tab, edit the SQL statement for the lookup.

   For example, you may wish to edit the following lookup:

   ```
   MPLT_ADI_SUPPLIERS.LKP_SPLR_ATTR1
   ```

4. Edit the SQL statement to use the desired code category.
5. Edit the SQL statement from `'GENERIC'` to the category you wish to use for the lookup.

17.12 About Resolving Dimension Keys

By default, dimension key resolution is performed by the Oracle Business Analytics Warehouse in the load mapping. The load mapping uses prepackaged, reusable lookup transformations to provide prepackaged 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.

17.12.1 Resolving the Dimension Key Using Lookup

If the dimension key is not provided to the load mapping through database joins, the load mapping performs the lookup in the dimension table. The load mapping does this using prepackaged lookup transformations.

The load mapping uses the integration ID, the DATASOURCE_NUM_ID and Lookup date in looking up the dimension key. All these columns are necessary for the load mapping to return the dimension key. The ports are described in Table 17–6.

Table 17–6 Columns Used in the load mapping Dimension Key Lookup

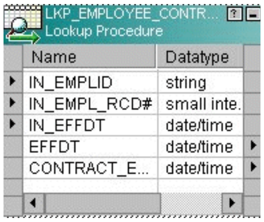
Port	Description
INTEGRATION ID	Uniquely identifies the dimension entity within its source system. Formed from the transaction in the source adapter of the fact table.
DATASOURCE_NUM_ID	Unique identifier of the source system instance.
Lookup Date	The primary date of the transaction; for example, receipt date, sales date, and so on.

In Figure 17–3, the Supplier Products Key Lookup transformation illustrates the three input columns needed for the load mapping lookup—the INTEGRATION ID, DATASOURCE_NUM_ID, and Date (lookup date). The transformation then outputs the Supplier Product key (the dimension key) to the data warehouse table W_SUPPLIER_PRODUCT_D.

If Category 2 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 Category 2 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 lookup for Employee Contract Data shown in Figure 17–13, you can see the effective dates used to provide the effective period of employee contracts.

Figure 17–13 Lookup for Employee Contract Data



LKP_EMPLOYEE_CONTR...
Lookup Procedure

Name	Datatype
IN_EMPLID	string
IN_EMPL_RCD#	small inte.
IN_EFFDT	date/time
EFFDT	date/time
CONTRACT_E...	date/time

17.13 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 enables 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.

17.13.1 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 mapping until the data reaches the W_EVENT_GRP_CODE column in the W_EVENT_TYPE_DS staging table.

The load mapping then ports the extracted source values (H and R from source system A, and 1 and 2 from source system B) into the source adapter mapplet. Within the source adapter, source values are translated into domain values (HIR and REH) based on a set of rules that are particular to your business practices.

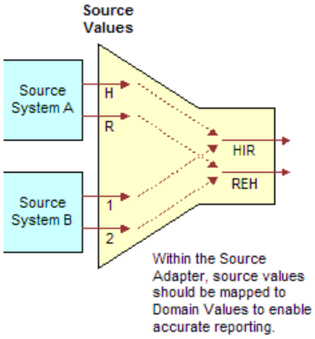
17.13.1.1 Preparing to Define the Rules

You must define the rules so that the source adapter 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 may 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 applicable source adapter mapplet. To set up the logic, modify the Expression transformation in the source adapter mapplet for each affected column. For information on setting up the rules for domain values, see Section 17.15, "Configuring the Domain Value Set Using Informatica PowerCenter Designer."

Figure 17–14 illustrates how the source values are converted to the domain values—HIR and REH.

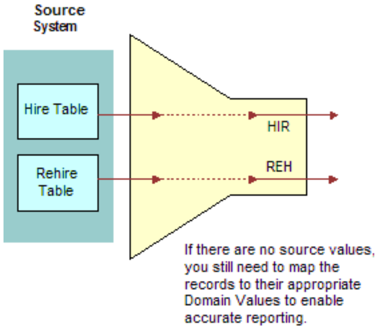
Figure 17–14 Source Values Translated to Domain Values



Within the Source Adapter, source values should be mapped to Domain Values to enable accurate reporting.

Figure 17–15 illustrates a different situation where the records may not contain a source value that flags the record as Hire or Rehire. In this case, the source system stores hires in one table and rehires in another table. To make this work, one possible solution is to modify the extract mappings to populate the W_EVENT_GRP_CODE column with HIR or REH. If the field is populated in the extract mapping, you can then carry those same values through the source adapter mapplet.

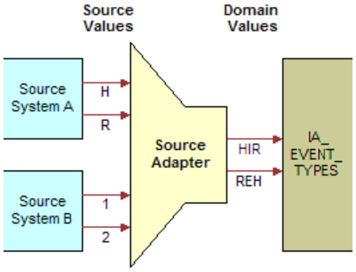
Figure 17–15 Source Values in Different Tables Translated to Domain Values



If there are no source values, you still need to map the records to their appropriate Domain Values to enable accurate reporting.

After the source adapter mapplet 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, as illustrated in Figure 17–16.

Figure 17–16 HIR and REH Values Populating the W_EVENT_TYPES Table



17.13.2 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:

17.13.2.1 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)
```

17.13.2.2 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
```

17.13.2.3 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)
```

17.13.2.4 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)
```

17.13.2.5 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)
```

17.13.2.6 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)
```

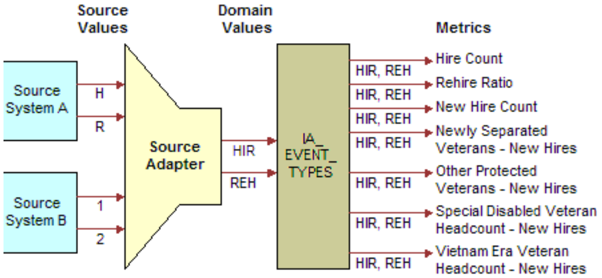
17.13.2.7 Vietnam Era Veteran Head count - New Hires

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)
```

Each of these metric calculations is based on the domain values HIR and REH. All records whose source values are converted to one of these domain values are included in the metric calculations, as shown in Figure 17–17.

Figure 17–17 Metric Values From HIR and REH Domain Values



17.13.3 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 may be hired internally.

If you have an event that represents both a New Hire and a New Position, you may 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.

17.14 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 Informatica PowerCenter Designer. For more information on configuring domain values using Informatica PowerCenter Designer, see Section 17.15, "Configuring the Domain Value Set Using Informatica PowerCenter Designer."

For a list of CSV worksheet files and their domain values for your application, see your application configuration chapter.

To map source values to domain values using CSV worksheet files:

1. Identify all 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 may 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 $PMServer\LkpFiles directory (for example, *INFA_HOME*\server\infa_shared\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 worksheet file.

17.15 Configuring the Domain Value Set Using Informatica PowerCenter Designer

If there are no worksheet files to map the source system values to the domain values, you need to modify the values using Informatica PowerCenter Designer. For more information on configuring the domain value set with CSV worksheet files, see Section 17.14, "Configuring the Domain Value Set with CSV Worksheet Files."

Configuring the domain value set for a particular column, using Informatica PowerCenter Designer, 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 using Informatica PowerCenter Designer:

1. Identify all 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 may 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 Informatica PowerCenter Designer, open the applicable source adapter mapplet.

5. Open the Expression transformation.

6. Locate the applicable port's expression so that you can modify it.

7. Edit the port's expression to map your source values to the existing domain values.

   Alternately, if you want to add additional domain values, add them in this same expression.

8. Save and validate your changes to the repository.

17.16 Configuring Conformed Dimensions

This section provides procedures on configuring objects that apply to more than one application and contains the following topic:

- Section 17.16.1, "Configuring Conformed Dimensions for Universal Adapter"

17.16.1 Configuring Conformed Dimensions for Universal Adapter

This section provides configuration procedures for modifying dimensions that are loaded using Universal Adapter.

17.16.1.1 Product Effective Dates in the Products Dimension

The Oracle Business Analytics Warehouse stores product Effective To (SRC_EFF_TO_DT) and From (SRC_EFF_FROM_DT) dates in the Products dimension table, W_PRODUCTS. In addition, the Products dimension stores a Support Withdrawn Date column, SPRT_WITHDRAWN_DT.

By default, the Support Withdrawn Date takes precedence over the product Effective To Date. This prioritization means that if you supply a value for the Support Withdrawn Date column in your flat file upload, the Oracle Business Analytics Warehouse uses that value as the product Effective To value as well, overwriting anything in the SRC_EFF_TO_DT column. You can change this default behavior by modifying the Products Expression in the Universal Source Products Extract mapping.

To modify the product Effective To Date logic for a flat file extract:

1. In Informatica PowerCenter Designer, open the Configuration for Universal Source folder.
2. In the SDE_Universal_ProductDimension mapping, open the W_PRODUCT_D expression.
3. Edit the logic for the SRC_EFF_TO_DT_OUT port.
4. Validate and save your changes.

A

Configuring Universal Adapter for CRM Analytics

This appendix provides instructions for Configuring Universal Adapter for CRM Analytics, and contains the following topics:

A.1 What is Universal Adapter for CRM Analytics?

Universal Adapter for CRM Analytics enables you to extract data from CRM transactional applications for analysis, when pre-packaged business adapters for these sources are not available. Unlike the pre-packaged business adapters that can only be used for particular source systems, Universal Adapter for CRM Analytics can be used for any source systems as long as the source data can be presented in the specified flat file format.

The CRM source data could also reside in custom applications or legacy systems. Universal Adapter for CRM Analytics is shipped as part of any of the Oracle BI Applications CRM Analytics products and is not separately priced.

A.2 Data Format Requirements

To use Universal Adapter for CRM Analytics, source data should be presented in flat files according to the following specifications:

- Data should be in comma delimited format (*.csv files).
- All date values should be in the format of YYYYMMDDHH24MISS. For example, 20071231140300 should be used for December 31, 2007, 2:03 pm.
- If a string data value contains one or more commas, then it should be double quoted.
- Data should start from line six of each file. The first five lines of each file will be skipped during ETL process.

A.3 About the Sample Source Files Provided

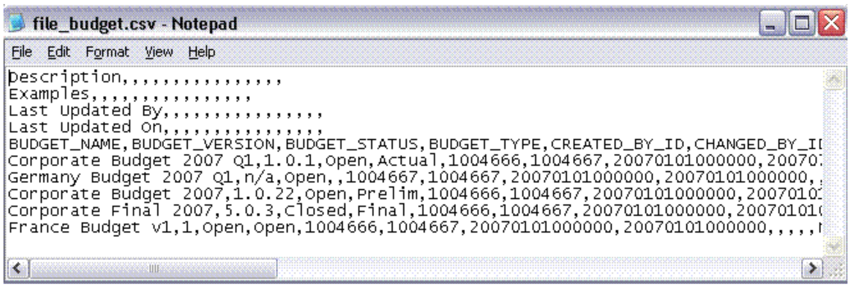
Oracle BI Applications provides a sample source file for each supported source file type. Typically, each sample source files contain sample data rows. Sample source files are located in:

MW_HOME\biapps\dwrep\Informatica\Sample Universal Source Files

For a list of sample source files, see Section A.5, "List of Sample Source Files for Universal Adapter for CRM Analytics."

The sample source file file_budget.csv is shown in Figure A–1.

Figure A–1 Sample Universal Adapter Source File



```
Description,,,,,,,,,,,,,,,,
Examples,,,,,,,,,,,,,,,,
Last Updated By,,,,,,,,,,,,,,,,
Last Updated On,,,,,,,,,,,,,,,,
BUDGET_NAME,BUDGET_VERSION,BUDGET_STATUS,BUDGET_TYPE,CREATED_BY_ID,CHANGED_BY_I
Corporate Budget 2007 Q1,1.0.1,open,Actual,1004666,1004667,20070101000000,200701
Germany Budget 2007 Q1,n/a,open,,1004667,1004667,20070101000000,20070101000000,,
Corporate Budget 2007,1.0.22,open,Prelim,1004666,1004667,20070101000000,2007010
Corporate Final 2007,5.0.3,Closed,Final,1004666,1004667,20070101000000,2007010
France Budget v1,1,open,open,1004666,1004667,20070101000000,20070101000000,,,,
```

A.4 How to Create Data Files for Universal Adapter for CRM Analytics

To create data files, Oracle recommends that you use the sample source files provided. For a list of sample source files, see Section A.5, "List of Sample Source Files for Universal Adapter for CRM Analytics."

To create data files for Universal Adapter for CRM Analytics:

1. Copy the sample source files (file_xxx.csv) from the *MW_HOME*\biapps\dwrep\Informatica\Sample Universal Source Files folder to the *INFA_HOME*\server\infa_shared\SrcFiles directory.

2. Delete the sample data rows from every sample source file.

   You must have all of the empty files in the *INFA_HOME*\server\infa_shared\SrcFiles folder first for ETL Execution Plans to run without interruptions. Some ETL Execution Plans might contain tasks that you do not need but that would fail and cause interruption in your ETL run if source files expected by the tasks cannot be found in the *INFA_HOME*\server\infa_shared\SrcFiles folder.

3. Populate the files that you need in the *INFA_HOME*\server\infa_shared\SrcFiles folder, making sure that the data starts at line six.

   For example, to load opportunity dimension data, you could populate the file_opty.ds.csv file.

4. In DAC, create an Execution Plan to load the data from these files.

   For instructions on creating an Execution Plan, see the *Oracle Business Intelligence Data Warehouse Administration Console User's Guide*.

A.5 List of Sample Source Files for Universal Adapter for CRM Analytics

This section lists sample source files for Universal Adapter for CRM Analytics, and contains the following topics:

- Section A.5.1, "List of Source Files for Seed Data and Common Dimensions"
- Section A.5.2, "List of Source Files for Specific Subject Areas"

A.5.1 List of Source Files for Seed Data and Common Dimensions

This section lists source files for seed data and common dimensions. These source files are mandatory and should be populated when using the CRM Universal Adapter.

Table A–1 Seed Data and Common Dimension Source Files

CRM Subject Area	Associated Source Files
Seed Data	■ file_codes_common ■ file_costlst_g ■ file_dual_g ■ file_exch_rate ■ file_lov_d ■ file_lst_of_val_g
Common Dimension	■ file_employee ■ file_geo_ds ■ file_int_org ■ file_int_org_dh ■ file_party_org_ds ■ file_party_per_ds ■ file_position_ds ■ file_product ■ file_user

A.5.2 List of Source Files for Specific Subject Areas

This section lists CRM Subject Areas and their related source files. Populate these source files selectively, depending on your business needs.

Table A–2 Subject Areas and Associated Source Files

CRM Subject Area	Associated Source Files
CRM - Automotive Vehicle Sales	■ file_asset_ds ■ file_household_ds ■ file_vehicle_fs
CRM - Consumer Goods - Performance	■ file_performance_fs
CRM - Consumer Goods - Promotion	■ file_plan_promo_ds ■ file_promotion_fs
CRM - Consumer Goods - Retail Audit	■ file_invloc_ds ■ file_plan_promo_ds ■ file_rtlaudit_fs

Table A–2 (Cont.) Subject Areas and Associated Source Files

CRM Subject Area	Associated Source Files
CRM - Consumer Goods Trade Funds	■ file_fndhdr_fs ■ file_fndtxn_fs ■ file_fund_ds ■ file_fund_fs ■ file_plan_promo_ds ■ file_terr
CRM - Forecasting	■ file_dtl_fcst_fs ■ file_fcst_ser_ds ■ file_industry_ds ■ file_opty_ds ■ file_source ■ file_sum_fcst_fs ■ file_terr
CRM - Loyalty	■ file_loy_accrual_item_fs.csv ■ file_loy_activity_fs.csv ■ file_loy_cost_fs.csv ■ file_loy_member_ds.csv ■ file_loy_member_marketing_fs.csv ■ file_loy_member_status_hist_fs.csv ■ file_loy_member_tier_hist_fs.csv ■ file_loy_member_tier_move_a_tmp.csv ■ file_loy_partner_point_block_fs.csv ■ file_loy_point_block_ds.csv ■ file_loy_point_type_ds.csv ■ file_loy_program_ds.csv ■ file_loy_promotion_ds.csv ■ file_loy_promotion_enrollee_fs.csv ■ file_loy_promotion_member_fs.csv ■ file_loy_promotion_product_fs.csv ■ file_loy_redemption_item_fs.csv ■ file_loy_referral_ds.csv ■ file_loy_revenue_fs.csv ■ file_loy_statement_fs.csv ■ file_loy_tier_tiercls_ds.csv ■ file_loy_transaction_id_ds.csv ■ file_loy_voucher_ds.csv

Table A–2 (Cont.) Subject Areas and Associated Source Files

CRM Subject Area	Associated Source Files
CRM - Marketing	▪ file_camp_hist_fs ▪ file_camp_opty_fs ▪ file_geo_region_ds ▪ file_household_ds ▪ file_household_fs ▪ file_industry_ds ▪ file_ld_wave_ds ▪ file_mktg_cost_fs ▪ file_mktg_goal_fs ▪ file_mktg_lead_fs ▪ file_mktg_lead_opty_h ▪ file_mktg_lead_ds ▪ file_offer_ds ▪ file_ofr_prod_fs ▪ file_opty_ds ▪ file_period_ds ▪ file_person_fs ▪ file_regn_ds ▪ file_resp_prod_fs ▪ file_response_ds ▪ file_response_fs ▪ file_segment_ds ▪ file_source ▪ file_terr ▪ file_vendor_ds ▪ file_wave_ds
CRM - Price	▪ file_order_ds.csv ▪ file_orderitem_fs.csv ▪ file_ordit_wtr_log_fs.csv ▪ file_pri_segment_ds.csv ▪ file_pri_strategy_ds.csv ▪ file_pwf_element.csv ▪ file_qteit_wtr_log_fs.csv ▪ file_quote_ds.csv ▪ file_quoteitem_fs.csv

Table A–2 (Cont.) Subject Areas and Associated Source Files

CRM Subject Area	Associated Source Files
CRM - Sales	■ file_agree_ds ■ file_entlmnt_ds ■ file_geo_region_ds ■ file_industry_ds ■ file_invoice_fs ■ file_offer_ds ■ file_oiprdattr_ds ■ file_opty_cmpt_fs ■ file_opty_con_fs ■ file_opty_ds ■ file_opty_stg_fs ■ file_order_ds ■ file_orderitem_fs ■ file_person_fs ■ file_prdattrnm_ds ■ file_qiprdattr_ds ■ file_quote_ds ■ file_quoteitem_fs ■ file_regn_ds ■ file_revn_fs ■ file_segment_ds ■ file_source ■ file_srvreq_ds ■ file_sstage_ds ■ file_terr
CRM - Service	■ file_activity_fs ■ file_agree_ds ■ file_agreeitem_fs ■ file_assess_ds ■ file_asset_ds ■ file_asset_fs ■ file_claimelmnt_ds ■ file_entlmnt_ds ■ file_opty_ds ■ file_quote_ds ■ file_region_ds ■ file_source ■ file_srvreq_ds ■ file_srvreq_fs ■ file_survey_fs

Table A–2 (Cont.) Subject Areas and Associated Source Files

CRM Subject Area	Associated Source Files
CRM - Telecom	■ file_agree_ds ■ file_entlmnt_ds ■ file_industry_ds ■ file_offer_ds ■ file_oiprdattr_ds ■ file_opty_ds ■ file_order_ds ■ file_orderitem_fs ■ file_prilstitem_ds ■ file_qiprdattr_ds ■ file_quote_ds ■ file_quoteitem_fs ■ file_regn_ds ■ file_segment_ds ■ file_source ■ file_srvreq_ds ■ file_sstage_ds ■ file_terr
CRM - Pharma - Account Call	■ file_accnt_rnk_ds ■ file_affl_aa_hs ■ file_affl_ac_hs ■ file_affl_ca_hs ■ file_affl_cc_hs ■ file_area_ds ■ file_call_accnt_fs ■ file_call_attnd_fs ■ file_con_rnk_ds ■ file_postn_con_ds
CRM - Pharma - Contact Call	■ file_affl_ac_hs ■ file_affl_cc_hs ■ file_area_ds ■ file_call_con_fs ■ file_con_rnk_ds ■ file_postn_con_ds
CRM - Pharma - Direct Sales	■ file_accnt_rnk_ds ■ file_affl_aa_hs ■ file_affl_ca_hs ■ file_alignmt_dhs ■ file_area_ds ■ file_ins_plan_ds ■ file_syndd_ds_fs

Table A–2 (Cont.) Subject Areas and Associated Source Files

CRM Subject Area	Associated Source Files
CRM - Pharma - Indirect Sales	■ file_accnt_rnk_ds ■ file_affl_aa_hs ■ file_affl_ca_hs ■ file_alignmt_dhs ■ file_area_ds ■ file_ins_plan_ds ■ file_syndd_ids_fs
CRM - Pharma - Medical Education	■ file_area_ds ■ file_con_rnk_ds ■ file_med_att_tmp ■ file_med_cst_tmp ■ file_med_ed_fs ■ file_med_evt_ds ■ file_med_plan_ds ■ file_postn_con_ds
CRM - Pharma - Objective	■ file_accnt_rnk_ds ■ file_affl_aa_hs ■ file_affl_ac_hs ■ file_affl_ca_hs ■ file_affl_cc_hs ■ file_alignmt_dhs ■ file_area_ds ■ file_call_accnt_fs ■ file_call_con_fs ■ file_con_rnk_ds ■ file_ins_plan_ds ■ file_objective_fs ■ file_plan_promo_ds ■ file_postn_con_ds ■ file_syndd_ds_fs ■ file_syndd_ids_fs ■ file_syndd_rx_fs
CRM - Pharma - Physician Plan Prescription	■ file_affl_aa_hs ■ file_affl_ac_hs ■ file_affl_ca_hs ■ file_affl_cc_hs ■ file_alignmt_dhs ■ file_area_ds ■ file_con_rnk_ds ■ file_postn_con_ds ■ file_syndd_xpt_fs

Table A–2 (Cont.) Subject Areas and Associated Source Files

CRM Subject Area	Associated Source Files
CRM - Pharma - Physician Prescription	▪ file_affl_ac_hs ▪ file_affl_cc_hs ▪ file_alignmt_dhs ▪ file_area_ds ▪ file_con_rnk_ds ▪ file_ins_plan_ds ▪ file_postn_con_ds ▪ file_syndd_rx_fs
CRM - Pharma - Plan Prescription	▪ file_affl_aa_hs ▪ file_affl_ca_hs ▪ file_alignmt_dhs ▪ file_area_ds ▪ file_syndd_pt_fs
CRM - Pharma - Weekly Physician Prescription	▪ file_affl_ac_hs ▪ file_affl_cc_hs ▪ file_alignmt_dhs ▪ file_area_ds ▪ file_con_rnk_ds ▪ file_ins_plan_ds ▪ file_postn_con_ds ▪ file_syndd_w_fs
CRM - Pharma - Profile Rank	▪ file_accnt_rnk_ds ▪ file_affl_aa_hs ▪ file_affl_ac_hs ▪ file_affl_ca_hs ▪ file_affl_cc_hs ▪ file_area_ds ▪ file_con_rnk_ds ▪ file_per_rank_fs ▪ file_postn_con_ds

Table A–2 (Cont.) Subject Areas and Associated Source Files

CRM Subject Area	Associated Source Files
CRM - Financial Service - Accounts and Application	■ file_agree_ds ■ file_asset_ds ■ file_asset_fs ■ file_balhstry_fs ■ file_fn_holding_ds ■ file_fn_holding_fs ■ file_industry_ds ■ file_ins_claim_ds ■ file_offer_ds ■ file_opty_ds ■ file_region_ds ■ file_regn_ds ■ file_revn_fs ■ file_segment_ds ■ file_source ■ file_src_offr_hs ■ file_srvreq_ds ■ file_srvreq_fs ■ file_sstage_ds ■ file_terr
CRM - Financial Service - Accounts and Insurances	■ file_asset_ds ■ file_asset_fs ■ file_claimelmnt_ds ■ file_industry_ds ■ file_ins_claim_ds ■ file_ins_claim_fs ■ file_terr
CRM - Financial Service - Balance History	■ file_asset_ds ■ file_balhstry_fs ■ file_industry_ds
CRM - Public Sector - Benefit	■ file_case_ds ■ file_benefit_fs ■ file_bnft_plan_ds

Table A–2 (Cont.) Subject Areas and Associated Source Files

CRM Subject Area	Associated Source Files
CRM - Public Sector - Case	■ file_activity_fs ■ file_case_ds ■ file_case_fs ■ file_incident_ds ■ file_lead_ds ■ file_period_ds ■ file_region_ds ■ file_srvreq_ds ■ file_srvreq_fs
CRM - Public Sector - Incident	■ file_case_ds ■ file_incident_ds ■ file_incident_fs ■ file_period_ds
CRM - Public Sector - Lead	■ file_case_ds ■ file_lead_ds ■ file_lead_fs ■ file_period_ds

Index