

Oracle® Business Intelligence Applications

Configuration Guide for Informatica PowerCenter Users

Version 7.9.5.1

E13766-01

January 2009

E13766-01

Copyright © 2009, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

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

Contents

Preface	xv
Audience	xv
Documentation Accessibility	xvi
Related Documents	xvi
Conventions	xvi
1 What's New in This Release	
1.1 What's New in <i>Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users?</i>	1-1
2 Overview of Configuring Oracle BI Applications	
2.1 High-Level Overview of Configuring Oracle BI Applications	2-1
3 Configuring Common Areas and Dimensions	
3.1 Source-Independent Configuration Steps	3-1
3.1.1 How to Configure Initial Extract Date	3-1
3.1.2 How to Configure Global Currencies	3-2
3.1.3 How to Configure Exchange Rate Types	3-3
3.1.4 How to Configure Fiscal Calendars	3-3
3.1.4.1 Setting Up the Fiscal Calendar by Fiscal Week	3-4
3.1.4.2 Setting Up the Fiscal Calendar by Fiscal Month	3-5
3.1.4.3 Reloading the Time Dimension Tables After the Data Warehouse Is Loaded....	3-5
3.1.4.4 Notes on Setting Up Fiscal Calendars	3-6
3.1.5 About Configuring GL Account Hierarchies	3-6
3.1.5.1 How to Configure GL Account Hierarchies Using GL Accounting Flexfield Value Sets Definitions	3-8
3.1.5.2 How to Configure GL Account Hierarchies Using Financial Statement Generator (FSG) Report Definition (for Oracle EBS)	3-11
3.1.6 Configuration Steps for Controlling Your Data Set for All Source Systems	3-15
3.1.6.1 How to Configure Data Source Num IDs	3-15
3.2 Oracle EBS-Specific Configuration Steps	3-16
3.2.1 Configuration Required Before a Full Load for Oracle EBS	3-17
3.2.1.1 Configuration of Product Hierarchy (Except for GL, HR Modules)	3-17
3.2.1.1.1 How to Identify Category Sets from EBS	3-17
3.2.1.1.2 How to Configure DAC Source System Parameters for	

	Product Hierarchy	3-18
3.2.1.1.3	How to Configure the Hierarchy Mapping with the Segments	3-18
3.2.1.2	Configure the Master Inventory Organization in Product Dimension Extract for Oracle 11i Adapter (Except for GL & HR Modules).....	3-19
3.2.1.3	How to Map Oracle GL Natural Accounts to Group Account Numbers	3-19
3.2.1.4	How to Do Group Account Correction for Oracle Applications.....	3-22
3.2.2	Configuration Steps for Controlling Your Data Set for Oracle EBS	3-23
3.2.2.1	How to Configure the Country Region and State Region Name	3-23
3.2.2.2	How to Configure the State Name.....	3-24
3.2.2.3	How to Configure the Country Name.....	3-25
3.2.2.4	How to Configure the Make-Buy Indicator.....	3-26
3.3	PeopleSoft-Specific Configuration Steps	3-26
3.3.1	Configuration Required Before a Full Load for PeopleSoft	3-26
3.3.2	Configuration Steps for Controlling Your Data Set for PeopleSoft.....	3-26
3.3.2.1	Internal Organization Hierarchy Extraction Customization.....	3-26
3.4	Siebel-Specific Configuration Steps.....	3-28
3.4.1	Configuration Required Before A Full Load for Siebel Source Systems	3-28
3.4.2	Configuration Steps for Controlling Your Data Set for Siebel Source Systems.....	3-28

4 Configuring Oracle Procurement and Spend Analytics

4.1	Overview of Oracle Procurement and Spend Analytics	4-1
4.1.1	Oracle Procurement and Spend Analytics Module	4-1
4.1.2	Supplier Performance Analytics Module.....	4-2
4.2	Configuration Required Before a Full Load for Oracle Procurement and Spend Analytics.....	4-3
4.2.1	Configuration Steps for Oracle Procurement and Spend Analytics for All Source Systems	4-3
4.2.1.1	How to Deploy Stored Procedures	4-3
4.2.2	Configuration Steps for Oracle Procurement and Spend Analytics for Oracle EBS..	4-4
4.2.2.1	How to Configure the DAC Parameter for Purchase Cycle Line.....	4-4
4.2.2.2	Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics	4-4
4.2.2.2.1	List of Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics	4-4
4.2.2.2.2	To configure domainValues_PaymentType_ora12.csv	4-6
4.2.2.2.3	To configure domainValues_Status_Purch_Approve_ora11i.csv	4-6
4.2.2.2.4	To configure domainValues_Status_Purch_Cycle_ora11i.csv.....	4-6
4.2.2.2.5	To configure domainValues_Xact_Types_Purch_Orders_ora11i.csv.....	4-7
4.2.2.2.6	To configure domainValues_Xact_Types_Purch_Requisitions_ora11i.csv ..	4-7
4.2.2.2.7	To configure domainValues_Xact_Types_PO_Line_Type_ora11i.csv	4-7
4.2.2.2.8	To configure domainValues_Xact_Types_PO_Line_Type_CODE1_ora11i.csv	4-8
4.2.2.2.9	To configure domainValues_ShipmentType_ora12.csv	4-8
4.2.3	Configuration Steps for Oracle Procurement and Spend Analytics for PeopleSoft ..	4-9
4.2.4	Configuration Steps for Oracle Procurement and Spend Analytics for Universal	4-9
4.2.5	Configuration Steps for Controlling Your Data Set.....	4-9
4.2.5.1	Configuration Steps for Oracle Procurement and Spend Analytics for All Source Systems	4-9

4.2.5.1.1	About Configuring the Purchase Receipts Aggregate Table	4-9
4.2.5.1.2	How to Configure the Purchase Receipts Aggregate Table.....	4-10
4.2.5.1.3	About Configuring the Purchase Cycle Lines Aggregate Table	4-11
4.2.5.1.4	How to Configure the Purchase Cycle Lines Aggregate Table	4-12
4.2.5.2	Configuration Steps for Oracle Procurement and Spend Analytics for Oracle EBS	4-12
4.2.5.2.1	How to Extract Particular Purchase Order Records.....	4-12
4.2.5.3	Configuration Steps for Oracle Procurement and Spend Analytics for PeopleSoft	4-13
4.2.5.4	Configuration Steps for Oracle Procurement and Spend Analytics for Universal.....	4-13
4.2.5.4.1	About Configuring Oracle Procurement and Spend Analytics for Universal Source	4-13
4.2.5.4.2	How to Configure the Preferred Merchant Flag.....	4-14
4.2.5.4.3	How to Configure the Customer Billable Indicator	4-15
4.2.5.4.4	How to Configure the Receipts Indicator	4-15
4.2.5.4.5	How to Configure Expense Payment Types	4-15
4.2.5.4.6	How to Configure the Default Expense Distribution Percentage	4-16
4.2.5.4.7	How to Configure Lookup Dates for Currency Conversion	4-16

5 Configuring Oracle Financial Analytics

5.1	Overview of Oracle Financial Analytics	5-1
5.2	Configuration Required Before a Full Load for Financial Analytics.....	5-2
5.2.1	Configuration Steps for Financial Analytics for All Source Systems.....	5-2
5.2.2	Configuration Steps for Financial Analytics for Oracle EBS	5-2
5.2.2.1	About Configuring Domain Values and CSV Worksheet Files for Oracle Financial Analytics	5-3
5.2.2.2	How to Configure domainValues_Xact_Types_DocTypes_ora12.csv for Oracle General Ledger and Profitability Analytics.....	5-3
5.2.2.3	How to Configure domainValues_Xact_Types_DocTypes_ora11i.csv for Oracle General Ledger and Profitability Analytics.....	5-4
5.2.2.4	How to Specify the Ledger or Set of Books for which GL Data is Extracted.....	5-5
5.2.3	Configuration Steps for Financial Analytics for PeopleSoft.....	5-7
5.2.3.1	About Configuring Domain Values and CSV Worksheet Files for Oracle Financial Analytics	5-8
5.2.3.2	How to Configure Domain Values for Oracle Receivables Analytics	5-8
5.2.3.3	How to insert beginning GL Balances into Oracle BI Applications	5-9
5.2.3.4	GL Account Dimension, Chartfields and Hierarchy for PeopleSoft.....	5-9
5.2.3.5	Group Account Configuration	5-11
5.2.4	Configuration Steps for Financial Analytics for Universal	5-12
5.2.5	Configuration Steps for Controlling Your Data Set.....	5-12
5.2.5.1	Configuration Steps for Financial Analytics for All Source Systems.....	5-12
5.2.5.1.1	How to Include UnApplied Payments in AR Aging Calculations	5-12
5.2.5.1.2	How to Configure the AP/AR Aging Tables.....	5-13
5.2.5.2	Configuration Steps for Financial Analytics for Oracle EBS.....	5-14
5.2.5.2.1	How GL Balances Are Populated in Oracle EBS	5-14
5.2.5.2.2	How to Configure Oracle Profitability Analytics Transaction Extracts.....	5-14

5.2.5.2.3	How to Configure Cost Of Goods Extract (for Oracle EBS 11i).....	5-15
5.2.5.2.4	How to Configure AP Balance ID for Oracle Payables Analytics.....	5-16
5.2.5.2.5	How to Configure AR Balance ID for Oracle Receivables Analytics and Oracle General Ledger and Profitability Analytics.....	5-16
5.2.5.2.6	How to Configure the AR Adjustments Extract for Oracle Receivables Analytics	5-17
5.2.5.2.7	How to Configure the AR Schedules Extract.....	5-17
5.2.5.2.8	How to Configure the AR Cash Receipt Application Extract for Oracle Receivables Analytics	5-18
5.2.5.2.9	How to Configure the AR Credit-Memo Application Extract for Oracle Receivables Analytics	5-18
5.2.5.2.10	How to Set Up Drill Down in Oracle BI Answers from General Ledger to Subledger	5-19
5.2.5.3	Configuration Steps for Financial Analytics for PeopleSoft.....	5-20
5.2.5.3.1	Overview of PeopleSoft Tree Data Structures.....	5-20
5.2.5.3.2	About PeopleSoft Trees in Oracle Financial Analytics	5-20
5.2.5.3.3	General Ledger Chartfield Hierarchy Extraction Customization	5-24
5.2.5.3.4	Internal Organization Hierarchy Extraction Customization	5-28
5.2.5.4	Configuration Steps for Financial Analytics Chain for Universal.....	5-31
5.2.5.4.1	About the Customer Costs Lines and Product Costs Lines Tables for Oracle's Profitability Analytics.....	5-31
5.2.5.4.2	How to Configure the Customer Costs Lines and Product Costs Lines Tables for Oracle's Profitability Analytics.....	5-32
5.2.5.4.3	How to Setup Drill Down in Oracle BI Answers from General Ledger to Subledger	5-32
5.2.5.4.4	How to Configure Budgets for Oracle Profitability Analytics	5-32

6 Configuring Oracle Supply Chain and Order Management Analytics

6.1	Overview of Oracle Supply Chain and Order Management Analytics.....	6-1
6.2	Configuration Required Before A Full Load for Oracle Supply Chain and Order Management Analytics	6-2
6.2.1	Configuration Steps for Oracle Supply Chain and Order Management Analytics for All Source Systems	6-2
6.2.2	Configuration Steps for Oracle Supply Chain and Order Management Analytics for Oracle EBS.....	6-2
6.2.2.1	About Configuring Domain Values and CSV Worksheet Files for Oracle Supply Chain and Order Management Analytics.....	6-3
6.2.2.2	How to Configure Invoice Type Domain Values	6-4
6.2.2.3	How to Configure Pick Types Domain Values	6-5
6.2.2.4	How to Configure Order Types Domain Values	6-5
6.2.2.5	How to Configure Pick Status Domain Values.....	6-5
6.2.2.6	How to Configure Invoice Status Domain Values.....	6-6
6.2.2.7	How to Configure Order Overall Status Domain Values.....	6-7
6.2.2.8	How to Configure Pay Method Domain Values.....	6-7
6.2.2.9	How to Configure Movement Types Domain Values	6-8
6.2.2.10	How to Configure Quantity Types for Product Transactions	6-8
6.2.3	Configuration Steps for Oracle Supply Chain and Order Management Analytics for PeopleSoft	6-9

6.2.4	Configuration Steps for Oracle Supply Chain and Order Management Analytics for Universal	6-9
6.2.5	Configuration Steps for Controlling Your Data Set.....	6-9
6.2.5.1	Configuration Steps for Oracle Supply Chain and Order Management Analytics for All Source Systems	6-9
6.2.5.1.1	Tracking Multiple Attribute Changes in Bookings	6-10
6.2.5.1.2	Process of Aggregating Oracle Supply Chain and Order Management Analytics Tables	6-11
6.2.5.1.3	About Tracking Multiple Products for Oracle Supply Chain and Order Management Analytics.....	6-14
6.2.5.1.4	Adding Dates to the Order Cycle Time Table for Post-Load Processing...	6-14
6.2.5.1.5	About Configuring the Backlog Period Date for Oracle Supply Chain and Order Management Analytics	6-15
6.2.5.1.6	Example of How Backlog Data Is Stored in the Backlog History Table....	6-15
6.2.5.1.7	About Configuring the Customer Status History Fact for Post-Load Processing In Oracle Supply Chain and Order Management Analytics.....	6-18
6.2.5.1.8	Configuring the Customer Status History Fact table	6-18
6.2.5.1.9	How to Configure the Customer Status History Fact table	6-19
6.2.5.1.10	About Configuring the Inventory Monthly Balance Table	6-19
6.2.5.1.11	How to Configure the Inventory Monthly Balance.....	6-20
6.2.5.1.12	About Configuring the Product Transaction Aggregate Table	6-21
6.2.5.1.13	How to Configure the Product Transaction Aggregate Table	6-21
6.2.5.2	Configuration Steps for Oracle Supply Chain and Order Management Analytics for Oracle EBS	6-22
6.2.5.2.1	How to Configure Sales Order Lines Data Storage.....	6-23
6.2.5.2.2	About the Handling of Booked and Nonbooked Orders in the Order Lines and Bookings Table	6-23
6.2.5.2.3	How to Configure Sales Schedule Lines Data Storage	6-24
6.2.5.2.4	About the Handling of Booked and Nonbooked Orders in the Sales Schedule Lines Table	6-25
6.2.5.2.5	About Loading Bookings at the Schedule Line Level.....	6-25
6.2.5.2.6	How to Configure Early and Late Tolerances for Shipments.....	6-26
6.2.5.2.7	How to Configure Sales Invoice Lines Data Storage	6-27
6.2.5.2.8	How to Configure the Sales Invoice Extract.....	6-27
6.2.5.2.9	How to Configure the Order Lines Execution Plan and OTB ETL	6-28
6.2.5.2.10	How to Configure Procurement and Spend Analytics for Oracle Supply Chain and Order Management Analytics	6-29
6.2.5.2.11	How to Configure Oracle Financial Analytics for Oracle Supply Chain and Order Management Analytics	6-29
6.2.5.2.12	About Tracking Attribute Changes in Bookings	6-29
6.2.5.2.13	About Viewing the Data Warehouse Changes by Salesperson ID	6-30
6.2.5.2.14	How to Configure Different Types of Backlog Calculations	6-31
6.2.5.2.15	Adding Closed Orders to Backlog Calculations.....	6-31
6.2.5.2.16	How to Configure the Bill of Materials Explosion Options	6-32
6.2.5.2.17	How to Configure the Left Bound and Right Bound Calculation Option	6-36
6.2.5.3	Configuration Steps for Oracle Supply Chain and Order Management Analytics for PeopleSoft	6-37
6.2.5.4	Configuration Steps for Oracle Supply Chain and Order Management Analytics for Universal	6-37

7 Configuring Oracle Human Resources Analytics

7.1	Overview of Oracle Human Resources Analytics.....	7-1
7.2	Configuration Required Before A Full Load for Oracle HR Analytics	7-2
7.2.1	Configuration Steps for Oracle HR Analytics for All Source Systems	7-2
7.2.2	Configuration Steps for Oracle HR Analytics for Oracle EBS.....	7-2
7.2.2.1	About Domain Values and CSV Worksheet Files for Oracle HR Analytics	7-3
7.2.2.2	How to Configure the domainValues_Employee_Ethnic_Group_Code_ora<ver>.csv	7-8
7.2.2.3	How to Configure the domainValues_Employee_Sex_MF_ora<ver>.csv.....	7-8
7.2.2.4	How to Configure the domainValues_Employee_Veteran_Status_Code_ora<ver>.csv	7-9
7.2.2.5	How to Configure the domainValues_Employment_Cat_ora<ver>.csv	7-9
7.2.2.6	How to Configure the domainValues_Employment_Exempt_Flg_ora<ver>.csv	7-10
7.2.2.7	How to Configure the domainValues_Employment_Full_Time_Flg_ora<ver>.csv.....	7-10
7.2.2.8	How to Configure the domainValues_Employment_Status_ora<ver>.csv.....	7-10
7.2.2.9	How to Configure the domainValues_EventTypes_ora<ver>.csv	7-11
7.2.2.10	How to Configure the domainValues_HRPosition_Active_Pos_Flg_ora<ver>.csv	7-11
7.2.2.11	How to Configure the domainValues_HRPosition_Exempt_Flg_ora<ver>.csv.....	7-11
7.2.2.12	How to Configure the domainValues_Job_Eeo_Cat_Code_ora<ver>.csv.....	7-12
7.2.2.13	How to Configure the domainValues_Job_Flsa_Stat_Code_ora<ver>.csv	7-12
7.2.2.14	How to Configure the domainValues_Pay_Type_Grp_Code_ora<ver>.csv ...	7-12
7.2.2.15	How to Configure the domainValues_Pay_Type_Flg_ora<ver>.csv	7-13
7.2.2.16	How to Configure Address Types for HR Profile	7-13
7.2.2.17	How to Configure Phone Types for HR Profile	7-14
7.2.2.18	How to Configure Education Degree Codes for Employee Dimension.....	7-15
7.2.2.19	About Configuring Key Flexfields	7-15
7.2.2.20	How to Configure the Key Flexfields for the Job Dimension	7-16
7.2.2.21	How to Configure the Key Flexfields for the HR Position Dimension.....	7-17
7.2.2.22	How to Configure the Key Flexfields for the Pay Grade Dimension	7-17
7.2.2.23	How to Configure Multi-Segmented Flexfields.....	7-18
7.2.2.24	How to Configure Flags for the Pay Type Dimension.....	7-19
7.2.2.25	How to Configure Classification Names for Payroll.....	7-19
7.2.3	Configuration Steps for Oracle HR Analytics for PeopleSoft	7-20
7.2.3.1	Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft HCM	7-21
7.2.3.2	How to Configure the domainValues_Employee_Ethnic_Group_Code_psft.csv	7-24
7.2.3.3	How to Configure the domainValues_Employee_Sex_MF_psft.csv	7-24
7.2.3.4	How to Configure the domainValues_Employee_Veteran_Status_Code_psft.csv	7-25
7.2.3.5	How to Configure the domainValues_Employment_Category_psft.csv	7-25
7.2.3.6	How to Configure the domainValues_Employment_Exempt_Flg_psft.csv	7-25
7.2.3.7	How to Configure the domainValues_Employment_Full_Time_Flg_psft.csv	7-26

7.2.3.8	How to Configure the domainValues_Employment_Status_psft.csv	7-26
7.2.3.9	How to Configure the domainValues_Emp_EventType_Grp_Code_psft.csv .	7-26
7.2.3.10	How to Configure the domainValues_Emp_EventType_SubG_Code_psft.csv	7-27
7.2.3.11	How to Configure the domainValues_HRPosition_Active_Pos_Flg_psft.csv .	7-27
7.2.3.12	How to Configure the domainValues_HRPosition_Exempt_Flg_psft.csv	7-28
7.2.3.13	How to Configure the domainValues_Job_Eeo_Cat_Code_psft.csv	7-28
7.2.3.14	How to Configure the domainValues_Job_Flsa_Stat_Code_psft.csv	7-29
7.2.3.15	How to Configure the domainValues_Pay_Type_Grp_Code_psft.csv	7-29
7.2.3.16	How to Configure the domainValues_Emp_EventType_EventReason_Code_psft.csv	7-30
7.2.3.17	How to Configure the domainValues_Emp_EventType_EventClass_Code_psft.csv	7-30
7.2.3.18	How to Configure the file_pension_comp_flg.csv	7-30
7.2.3.19	How to Configure Compensation Flag and Pension Compensation Flag for Pay Type Dimension	7-31
7.2.3.20	How to Configure the Performance Review Type ETL Runtime Parameter ...	7-31
7.2.3.21	How to Configure Event Types using domainValues_EventTypes_psft.csv ..	7-31
7.2.3.22	How to Configure Department Trees for Oracle HR Analytics	7-32
7.2.4	Configuration Steps for Oracle HR Analytics for Universal	7-36
7.2.5	Configuration Steps for Controlling Your Data Set.....	7-36
7.2.5.1	Configuration Steps for Oracle HR Analytics for All Source Systems	7-36
7.2.5.1.1	How to Aggregate the Payroll Table for Oracle HR Analytics.....	7-37
7.2.5.1.2	About Configuring the Employee Snapshot Fact tables.....	7-37
7.2.5.1.3	How to Configure the Rolling Period for Storing Employee Daily Snapshot Data	7-39
7.2.5.1.4	How to Configure the Granularity of the Employee Monthly Snapshot Data	7-39
7.2.5.2	Configuration Steps for Oracle HR Analytics for Oracle EBS	7-39
7.2.5.3	Configuration Steps for Oracle HR Analytics for PeopleSoft	7-39
7.2.5.3.1	How to Tune Performance for the Payroll Table for PeopleSoft HCM HR Analytics	7-40
7.2.5.4	Configuration Steps for Oracle HR Analytics for Universal.....	7-40

8 Configuring Oracle Sales Analytics

8.1	Overview of Oracle Sales Analytics	8-1
8.2	Configuration Required Before A Full Load.....	8-1
8.2.1	How to Deploy Stored Procedures	8-1
8.2.2	Configuration Steps for Controlling Your Data Set	8-2
8.2.2.1	How to Configure the Snapshot Frequency in the Pipeline Fact	8-2
8.3	Configuring Usage Accelerator	8-2
8.3.1	What is Usage Accelerator?.....	8-2
8.3.1.1	Configuring Usage Accelerator	8-2
8.3.1.1.1	Tracking of Targeted Users.....	8-3
8.3.1.1.2	Other Siebel Required Responsibilities	8-3
8.3.1.2	About Deployment Date	8-6
8.3.1.3	About Metric Definitions and Variables	8-6

8.3.1.3.1	User Adoption Metrics	8-6
8.3.1.3.2	Opportunity Updates Metrics	8-6
8.3.1.3.3	Completeness Metrics.....	8-7
8.3.1.3.4	Coverage Metrics.....	8-7
8.3.2	About the Usage Accelerator Execution Plan.....	8-8
8.3.3	Using Usage Accelerator with Siebel 7.5.3 Applications	8-8
8.3.3.1	Usage Accelerator Application Integration with Siebel 7.5.3 CRM Application	8-8
8.3.3.2	Usage Accelerator-Analytics Adapter Files.....	8-10

9 Configuring Oracle Contact Center Telephony Analytics

9.1	Overview of Oracle Contact Center Telephony Analytics.....	9-1
9.2	Configuration Required Before A Full Load for Contact Center Telephony Analytics ..	9-1
9.2.1	About the Dimension Key Resolution Process for Universal Source	9-2
9.2.2	How to Load the Dimension Table	9-2
9.2.3	How to Load the Fact Table	9-3
9.2.4	Setting Up The DATASOURCE_NUM_ID Parameter for the Universal Adapter	9-3
9.2.5	About Configuring the Event Type Columns	9-3
9.2.5.1	The Activity Type Column.....	9-3
9.2.5.2	Call Type (or Contact Type) Column	9-4
9.2.5.3	Call Event Type Column	9-4
9.2.5.4	After-Call Work Activity (ACW) Type Column	9-5
9.2.6	About Configuring The Contact Reason Column.....	9-5
9.2.7	About Configuring Contact Status Type Column	9-6
9.2.8	Setting Up the Representative Activities Table.....	9-6
9.2.9	Setting Up the Contact Center Performance Table.....	9-7
9.2.10	Setting Up the Benchmarks and Targets Table	9-8
9.2.11	Logical Delete of Records From Fact Tables	9-10
9.2.12	How to Configure Dates and Times	9-11
9.2.13	Configuring Flags for Oracle Contact Center Telephony Analytics	9-12
9.2.14	How to Exclude Representative Data for Post-Load Processing.....	9-13
9.2.15	How to Configure Alternate Load Plan to Enable Direct Load to the Entry-Level Rep Activity Aggregates	9-13
9.2.16	How to Configure Alternate Load Plan to Enable Direct Load to the Entry-Level Contact Performance Aggregates	9-14
9.2.17	How to Set Up the Integration_Id of the Contact Center Telephony Facts in the Warehouse	9-15
9.2.18	Setting Up the IVR Menu and IVR Navigation Profile Tables.....	9-16
9.2.18.1	Determining Navigation Profile and Transfer Flag for a Call	9-16
9.2.18.2	Slowly Changing Dimension Behavior of IVR Menu and IVR Navigation Profile Dimensions	9-17
9.2.19	How to View Duration Metrics in hh:mm:ss Format in Oracle BI Answers	9-17
9.2.20	Configuration Steps for Controlling Your Data Set.....	9-18

10 Configuring the Oracle BI Repository

10.1	Modifying the Oracle BI Repository for Siebel Applications Versions 6.3 and 7.5.....	10-2
10.2	Configuring the Oracle BI Repository Connections	10-2
10.2.1	About the Predefined Connection Pools in the Oracle Business	

	Analytics Warehouse	10-2
10.2.1.1	How to Configure Dynamic Data Source Names.....	10-4
10.2.2	How to Configure the Oracle BI Repository Connections.....	10-4
10.2.3	How to Configure Oracle BI Repository Variables.....	10-5
10.3	Setting up Date-Specific Metrics.....	10-5
10.4	Setting Up Additional Time Series Metrics for Oracle Business Analytics Warehouse	10-6
10.5	Setting Up Additional Dimension Tables for Oracle Business Analytics Warehouse...	10-7
10.6	About the Period Ago Keys for Oracle Business Analytics Warehouse	10-7
10.7	About Oracle BI Time Repository Variables.....	10-7
10.8	About User Authentication	10-9
10.9	About the Security or Visibility Configuration	10-9
10.10	About the Group Variable	10-12
10.11	About Configuring Usage Tracking for Oracle Business Analytics Warehouse	10-12
10.12	About the Incremental Deployment of the Oracle BI Applications Repository	10-12

11 Customizing the Oracle Business Analytics Warehouse

11.1	Oracle Business Analytics Warehouse Customization Scenarios.....	11-1
11.1.1	Types of Customizations	11-2
11.1.2	Upgrade Consideration	11-2
11.2	Category 1 Customizations: Adding Columns to Existing Fact or Dimension Tables..	11-2
11.2.1	About Extending Mappings.....	11-3
11.2.2	Impact of Customization on Upgrade	11-4
11.2.2.1	Points to Remember	11-5
11.2.3	Typical Steps to Extend Mappings in the Oracle Business Analytics Warehouse..	11-6
11.2.4	Scenario for Extending the Oracle Business Analytics Warehouse.....	11-6
11.2.4.1	Tips for Modifying the SQL Override in a Source Qualifier.....	11-7
11.2.4.2	Example of Extracting Data from a Siebel Source	11-7
11.2.4.3	Including a Source Table for the Change Capture Process	11-9
11.2.4.4	Example of Extracting Data from a Non-Siebel Packaged Source	11-9
11.2.4.5	Example of Extracting Data from a Universal Source.....	11-11
11.2.4.6	Example of Loading Data into an Existing Target Table.....	11-12
11.2.4.7	Updating DAC.....	11-13
11.3	Other Types of Customizations Requiring Special Handling.....	11-13
11.3.1	Modifying Category 2 SCD Triggers	11-13
11.3.2	Adding Dimensions to Existing Facts.....	11-14
11.3.3	Adding Date Dimensions to Existing Facts	11-16
11.3.4	Adding Currencies to an Existing Table	11-17
11.4	Category 2 Customizations: Adding Additional Tables.....	11-17
11.4.1	About Creating New Dimension or Fact Tables	11-17
11.4.1.1	Required Columns.....	11-18
11.4.1.2	About the Oracle Business Analytics Warehouse DATASRC_NUM_ID Column.....	11-18
11.4.2	Using Custom Folders in the Oracle Business Analytics Warehouse.....	11-19
11.4.3	Creating Custom Informatica Workflows.....	11-19
11.4.4	Important Notes About Customizing the Oracle Business Analytics Warehouse	11-20
11.4.4.1	Additional Customization Instructions.....	11-20
11.4.5	Including a Source Table for Change Capture Process.....	11-22

11.4.6	Adding a New Dimension in the Oracle Business Analytics Warehouse.....	11-23
11.4.7	Adding a New Fact Table in the Oracle Business Analytics Warehouse	11-23
11.4.8	Adding a New Dimension Table for a New Fact Table in the Oracle Business Analytics Warehouse	11-24
11.5	Category 3 Customizations: Adding New Data as a Whole Row into a Standard Dimension Table	11-24
11.6	Configuring Extracts.....	11-25
11.6.1	Extracting Additional Data	11-25
11.6.1.1	Extracting New Data Using an Existing Source Table.....	11-26
11.6.1.2	Extracting Data from a New Source Table.....	11-26
11.6.2	Setting Up the Delimiter for Source Files.....	11-27
11.6.3	Configuring a Source Adapter Mapplet.....	11-27
11.7	Configuring Loads.....	11-28
11.7.1	Filtering and Deleting Records	11-28
11.7.2	About Primary Extract and Delete Mappings Process.....	11-29
11.7.3	About Working with Primary Extract and Delete Mappings	11-30
11.7.3.1	Deleting the Configuration for Source-Archived Records	11-30
11.7.3.2	Enabling Delete and Primary Extract Sessions	11-31
11.8	Configuring Slowly Changing Dimensions.....	11-31
11.8.1	About Identifying Historically Significant Attributes.....	11-32
11.8.1.1	About the Extract View.....	11-32
11.8.2	Category 1 and Category 2 Slowly Changing Dimensions	11-33
11.8.2.1	Category 1 Slowly Changing Dimension.....	11-33
11.8.2.2	Category 2 Slowly Changing Dimension.....	11-33
11.8.2.3	Effective Dates.....	11-34
11.9	About Stored Lookups	11-35
11.10	Codes Lookup.....	11-35
11.10.1	W_CODES Table	11-36
11.10.2	Codes Mappings	11-36
11.10.3	Codes Mapplets	11-37
11.10.4	Configuring Extension Column Code Description Lookups	11-37
11.11	About Resolving Dimension Keys	11-37
11.11.1	Resolving the Dimension Key Using Lookup	11-38
11.12	About Domain Values.....	11-39
11.12.1	About the Domain Value Conversion Process	11-39
11.12.1.1	Preparing to Define the Rules.....	11-39
11.12.2	About the Importance of Domain Values	11-40
11.12.2.1	Hire Count	11-41
11.12.2.2	Re-hires Ratio	11-41
11.12.2.3	New Hire Count	11-41
11.12.2.4	Newly Separated Veterans - New Hires	11-41
11.12.2.5	Other Protected Veterans - New Hires.....	11-41
11.12.2.6	Special Disabled Veteran Head count - New Hires.....	11-41
11.12.2.7	Vietnam Era Veteran Head count - New Hires.....	11-42
11.12.3	About Extending the Domain Value Set	11-42
11.13	Configuring the Domain Value Set with CSV Worksheet Files.....	11-42
11.14	Configuring the Domain Value Set Using Informatica PowerCenter Designer.....	11-43
11.15	Configuring Conformed Dimensions	11-44

11.15.1	Configuring Conformed Dimensions for Universal Adapter	11-44
11.15.1.1	Product Effective Dates in the Products Dimension	11-44

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, 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:

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

Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users contains instructions for configuring and customizing Oracle BI Applications release 7.9.5.1.

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.
- On the Oracle Technology Network at http://www.oracle.com/technology/documentation/bi_apps.html (to register for a free account on the Oracle Technology Network, go to <http://www.oracle.com/technology/about/index.html>).

Audience

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

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

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

Related Documents

For more information, see the following documents in the Oracle BI Applications release 7.9.5.1 documentation set (available at http://www.oracle.com/technology/documentation/bi_apps.html):

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

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.
<i>italic</i>	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 BI Applications"](#)

What's New in This Release

This section describes the changes in this version of the documentation to support release 7.9.5.1 of the software.

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

The *Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users* contains configuration and customization information that previously appeared in *Oracle Business Intelligence Applications Installation and Configuration Guide*.

For this release, quality enhancements were made to many sections in *Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users*. This guide also includes the following new and changed topics:

- [Section 3.1.5.1, "How to Configure GL Account Hierarchies Using GL Accounting Flexfield Value Sets Definitions,"](#) was updated to include configuration information for Oracle Federal Financial Analytics.
- [Section 5.2.5.3, "Configuration Steps for Financial Analytics for PeopleSoft,"](#) was updated to include configuration information for Oracle Federal Financial Analytics.
- [Section 7.2.3.22, "How to Configure Department Trees for Oracle HR Analytics,"](#) was added.

Overview of Configuring Oracle BI Applications

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

2.1 High-Level Overview of Configuring Oracle BI Applications

The list below provides a high-level overview of the tasks you must complete to configure Oracle BI Applications.

Note: Before you perform the tasks below, 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 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, "Siebel-Specific Configuration Steps"](#)
3. Perform the tasks in one of the following sections depending on the application 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"](#)
4. Configure the Oracle BI Repository by following the tasks in [Chapter 10, "Configuring the Oracle BI Repository."](#)
5. (Optional) Customize the Oracle Business Analytics Warehouse by following the tasks in [Chapter 11, "Customizing the Oracle Business Analytics Warehouse."](#)

Part II

Configuring Oracle BI Applications

Part II contains instructions for the mandatory tasks required to configure Oracle BI Applications.

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*.

To configure any of the application families within Oracle BI Applications, you first need to perform the steps in the section entitled, [Chapter 3, "Configuring Common Areas and Dimensions."](#)

You then need to perform the tasks in one of the following sections depending on the application 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"](#)

Configuring Common Areas and Dimensions

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

To configure Oracle BI Applications, you first need to perform the steps in [Section 3.1, "Source-Independent Configuration Steps."](#)

You then need to 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, "Siebel-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, "How to Configure Fiscal Calendars"](#)
- [Section 3.1.5, "About Configuring GL Account Hierarchies"](#)
 - [Section 3.1.5.1, "How to Configure GL Account Hierarchies Using GL Accounting Flexfield Value Sets Definitions"](#)
 - [Section 3.1.5.2, "How to Configure GL Account Hierarchies Using Financial Statement Generator \(FSG\) Report Definition \(for Oracle EBS\)"](#)

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.

When you set the Initial Extract Date parameter, make sure that you set it to the beginning of an accounting period and not a date in the middle of an accounting period. For example, if you decide to extract data from June 2005, and the June 2005 accounting period starts 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 your custom container from the drop-down list to the right of the Execute button.
2. Click 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 in.
- 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. In the target table, there will be two amount columns and three 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.
2. Select your custom container from the drop-down list to the right of the Execute button.
3. Click the Source System Parameters tab.
4. Locate the following parameters, and set the currency code values for them in the Value field:

- \$\$GLOBAL1_CURR_CODE (for the document currency).
- \$\$GLOBAL2_CURR_CODE (for the local currency).
- \$\$GLOBAL3_CURR_CODE (for the global currency).

Make sure that you spell the currencies as they are spelled in your source OLTP system.

5. 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 allows 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 allows 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, select your custom container from the drop-down list to the right of the Execute button.
2. Click 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 How to Configure Fiscal Calendars

When you configure time dimension tables, W_DAY_D is the base table that represents the time dimension in the Oracle Business Analytics Warehouse. There are two lines of aggregate dimension tables built on this table, as follows:

- Regular calendar tables
- Fiscal calendar tables

Table 3–1 shows the tables used to store calendar information.

Table 3–1 Tables in Base Table W_DAY_D Used to Store Calendar Information

Regular Calendar Tables in W_DAY_D	Fiscal Calendar Tables in W_DAY_D
W_WEEK_D	W_FSCL_WEEK_D
W_MONTH_D	W_FSCL_MONTH_D
W_QTR_D	W_FSCL_QTR_D
W_YEAR_D	W_FSCL_YEAR_D

There are two parameters `$$START_DATE` and `$$END_DATE` for the task `SIL_DayDimension` that need to be 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. Once the records are created in `W_DAY_D`, the aggregate calendar tables are loaded by their respective SIL mapping.

By default, Oracle BI Applications supports one fiscal calendar. Fiscal data is first loaded in the `W_DAY_D` table and then the SIL mappings read data from `W_DAY_D` and load data into the aggregate fiscal time dimension tables, such as fiscal week, fiscal month, fiscal quarter and fiscal year.

You may choose to provide fiscal calendar information in terms of the fiscal weeks of your organization or in terms of the fiscal months of your organization. In either case, The SIL mappings are designed to derive the fiscal week from the start and end date of a fiscal month by grouping into periods of seven days each.

3.1.4.1 Setting Up the Fiscal Calendar by Fiscal Week

In this option you provide data for the fiscal year, fiscal month, fiscal week and start date of fiscal week. The fiscal month information is derived using the 4-4-5 rule for grouping weeks into months. The fiscal week end date is derived based on the start date of the next week that is supplied in the data file. Fiscal months are grouped into sets of four months each to determine the fiscal quarter.

To set up the fiscal calendar by fiscal week

- Using a text editor, open the file `fiscal_week.csv`, which is located in the `\OracleBI\dwrep\Informatica\SrcFiles` directory. Enter the fiscal year, fiscal month, fiscal week and the start date of the fiscal week in `YYYYMMDD`. Records must be inserted in ascending order of fiscal year, fiscal month, fiscal week. Save and close the file.

Note: Make sure you enter accurate fiscal data because there is no check done within the Informatica mappings.

- In DAC, do the following:
 - Go to the Design view, and select your custom container from the drop-down list.
 - Click the Tasks tab.
 - Query for the task `SIL_DayDimension_FiscalMonth_Extract`, and inactivate it by deselecting the Inactive check box.
 - Query for the task `SIL_DayDimension_FiscalWeek_Extract`, and activate it by selecting the Inactive check box.

3.1.4.2 Setting Up the Fiscal Calendar by Fiscal Month

In this option you can provide data at the level of fiscal month. The SIL_DayDimension_FiscalMonth_Extract task divides the fiscal month into fiscal weeks of seven days each. If the number of days in the fiscal month is not in multiples of seven, the last week will have a fewer number of days.

To configure the fiscal calendar by fiscal month

1. Using a text editor, open the file `fiscal_month.csv`, which is located in the `\OracleBI\dwrep\Informatica\SrcFiles` directory. Enter the fiscal year, fiscal month and the start date of `Fiscal_Month` in `YYYYMMDD` format. Save and close the file.

Note: Make sure that you enter accurate fiscal data because there is no check done within the Informatica mappings.

2. In DAC, do the following:
 - a. Go to the Design view, and select your custom container from the drop-down list.
 - b. Click the Tasks tab.
 - c. Query for the task `SIL_DayDimension_FiscalWeek_Extract`, and inactivate it by deselecting the Inactive check box.
 - d. Query for the task `SIL_DayDimension_FiscalMonth_Extract`, and activate it by selecting the Inactive check box.

3.1.4.3 Reloading the Time Dimension Tables After the Data Warehouse Is Loaded

The data in time dimension tables is loaded once during the initial full load. Subsequently, the `SIL_*_UpdateFlag` mappings run everyday to update the domain value codes, which indicate whether a day, week, month, quarter or year is 'Current', 'Next' or 'Previous' as of the current day. The `SIL_Fiscal_UpdateFlag` mappings also update the flags that indicate whether a fiscal week, month, quarter or year is 'Current', 'Previous' or 'Next' with respect to the system date.

You might want to extend the range of data that you have in your time dimension tables sometime after the data warehouse is in production. In order to achieve this, follow the steps below to initiate the full load ETL run of `W_DAY_D` and all the aggregate time dimension tables.

To set up the load strategy of the time dimension table

1. In DAC, go to the Setup view, click the Physical Data Sources tab, and then click on the connection `DataWarehouse`.
2. Select the Refresh Dates subtab in the bottom pane.

The Refresh Dates subtab displays the refresh dates for all of the tables. Double-click on the refresh date for `W_DAY_D`, and make it null. Do the same for the following aggregate time dimension tables: `W_WEEK_D`, `W_QTR_D`, `W_MONTH_D`, `W_YEAR_D`, `W_FSCL_WEEK_D`, `W_FSCL_QTR_D`, `W_FSCL_MONTH_D`, and `W_FSCL_YEAR_D`.

3. In the Design view, select your custom container from the drop-down list.
4. Click the Tasks tab, and query for the task `SIL_DayDimension`.

5. Set the `$$START_DATE` and `$$END_DATE` parameters to the appropriate start date and end date of the new date range.
6. Save the task.

Note: Make sure that the new date range defined by the parameters `$$START_DATE` and `$$END_DATE` encompasses the old date range entirely. Otherwise, records can be lost. Also make sure you have provided the fiscal information for the new date range also in the appropriate fiscal input file. The next time you run your ETL execution plan, the calendar dimension will be truncated and reloaded with the new date range that you have specified. Since the calendar dimensions use smart keys, none of the fact tables that refer to these tables need to be reloaded.

3.1.4.4 Notes on Setting Up Fiscal Calendars

The following notes pertain to the process of setting up the fiscal calendar:

- If there is a week (starting on a Sunday and ending on a Saturday) that falls across two calendar years, the week is counted in both years. For example, the week that starts on 12/30/2007 will be counted in both 2007 and 2008. In 2007, the week start date will 12/30/2007 and the end date will be 12/31/2007. In 2008, this will be the first week with start date as 01/01/2008 and end date as 01/05/2008.
- `W_DAY_D` stores 31 records for each month 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'.

3.1.5 About Configuring GL Account Hierarchies

Configuring GL account hierarchies is required if you are deploying one of the following applications:

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

There are two ways to configure general ledger account hierarchies:

- Using General Ledger Accounting Flexfield Value Sets Definitions
For instructions, see [Section 3.1.5.1, "How to Configure GL Account Hierarchies Using GL Accounting Flexfield Value Sets 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 3.1.5.2, "How to Configure GL Account Hierarchies Using Financial Statement Generator \(FSG\) Report Definition \(for Oracle EBS\)."](#)

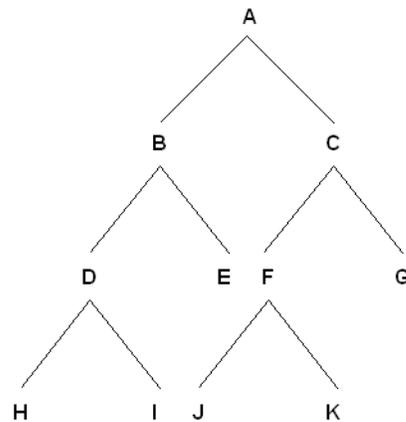
Whichever method you choose to set up GL account hierarchies, you store the hierarchy information in the W_HIERARCHY_D table.

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.

The figure below shows an example of this hierarchy for US Acct.

Figure 3–1 Example of Hierarchy for US Acct



[Table 3–2](#) shows how the hierarchy for US Acct would be stored in the W_HIERARCHY_D table.

Table 3–2 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

Table 3–2 (Cont.) 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
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

3.1.5.1 How to Configure GL Account Hierarchies Using GL Accounting Flexfield Value Sets 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 3–3 Example Chart of Accounts

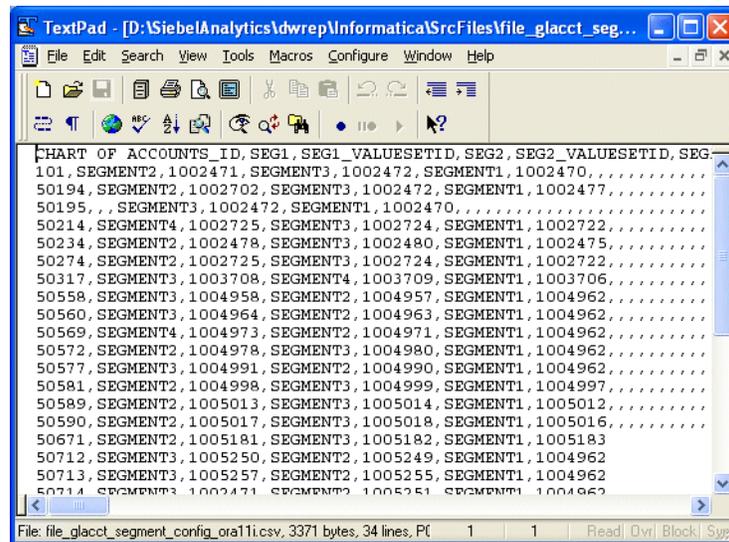
Segment Type	US Chart of Account (4256) value	APAC Chart of Account (4257) value
Company	Stores in segment 3	Stores in segment 1
Natural Account	Stores in segment 4	Stores in segment 3
Cost Center	Stores in segment 5	Stores in segment 2
Geography	Stores in segment 2	Stores in segment 2
Line of Business (LOB)	Stores in segment 1	Stores in segment 4

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 make sure 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.

About the ETL Process for GL Accounting Flexfields

Before you run the ETL process for GL accounts, you need to specify the segments that you want to analyze using the ETL configuration file named file_glacct_segment_config_<source system>.csv, located in \OracleBI\dwrep\Informatica\SrcFiles.

Figure 3–2 Screen Shot of file_glacct_segment_config_ora.csv Opened in a Text Editor

In file_glacct_segment_config_<source system>.csv, you need to specify the segments of the same type in the same column. For example, you might store all Cost Center segments from all 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 Cost Center segments from all COAs in ACCOUNT_SEG2_CODE column in W_GL_ACCOUNT_D.
- Store all Natural Account segments from all COAs in ACCOUNT_SEG3_CODE column in W_GL_ACCOUNT_D.
- Store all LOB segments from all COAs in ACCOUNT_SEG4_CODE column in W_GL_ACCOUNT_D.
- In W_GL_BALANCE_A (where you store GL account balances at aggregated level), you want to store GL account balances at Company and Cost Center level instead of at GL Code Combination level.

Figure 3–3 shows how file_glacct_segment_config_<source system>.csv would be configured to implement the business requirements specified above.

Figure 3–3 Screenshot of file_glacct_segment_config_ora.csv in text editor

	A	B	C	D	E	F	G	H	I
1	CHART OF ACCOUNTS, SEG1	SEG1_VALUESETID	SEG2	SEG2_VALUESETID	SEG3	SEG3_VALUESETID	SEG4	SEG4_VALUESETID	
2	4256	SEGMENT3	1002471	SEGMENTE	1002472	SEGMENT4	100247C	SEGMENT1	1002478
3	4257	SEGMENT1	1002702	SEGMENT2	1002472	SEGMENT3	1002477	SEGMENT4	1002479
4	AGGREGATION Y			Y					

Configuring file_glacct_segment_config_<source system> for U.S. Federal Financials 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.

To configure hierarchies with GL accounting flexfields

1. Configure file_glacct_segment_config_<source system>.csv.
 - a. Navigate to OraceBI\dwrep\Informatica\SrcFiles.
 - b. Open file_glacct_segment_config_<source system>.csv in a text editor.
 - c. Specify the segments you want to analyze.
2. In DAC, do the following:
 - a. Navigate to the Subject Areas tab, and query for 'Financials – General Ledger'.
 - b. In the Configuration Tags subtab, do the following:
 - * Query for the tag 'Oracle – Extract Value Set Hierarchies', and make sure the Inactive check box is not selected.
 - * Query for the tag 'Oracle – Extract FSG Hierarchies', and make sure the Inactive check box is selected.
 - c. Click Assemble to reassemble the subject area.
 - d. 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*.
 - e. Run the execution plan for General Ledger Accounts.
3. A HIER_CODE filter should be specified in the Business Model layer to restrain the output of the logical table to have only one hierarchy. You must set the HIER_CODE to be filtered by the valueset IDs that are applicable for that particular hierarchy. The list of the valueset IDs would be the same as the valueset IDs you configured in the CSV file in the first set.

To specify a filter in the Business Model layer of the Oracle BI Repository to restrain the output of logical table, do the following using the Oracle BI Administration Tool:

- a. Expand each logical table for the hierarchies, for example, Dim - GL ValueSetHierarchy1, and open the logical table source under it.
- b. Go to the Content tab.
- c. In the 'use WHERE clause...' text box, apply a filter on the corresponding physical table alias of W_GL_SEGMENT_D.

For example:

```
"Oracle Data Warehouse".Catalog.dbo.Dim_W_GL_SEGMENT_D_Segment1.SEGMENT_
LOV_ID IN (<comma separated valuesetids>)
```

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

4. 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.

3.1.5.2 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 3.1.5, "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.

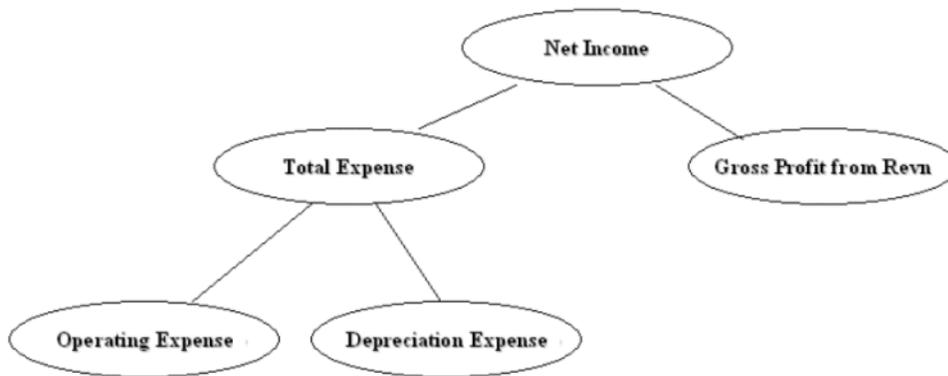
Oracle FSG hierarchies are extracted from following Oracle EBS source tables:

- **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 diagram below shows an example hierarchy, with the top level Net Income node having two child nodes, Total Expense, and Gross Profit from Revn, and the Total Expense node having two child nodes, Operating Expense, and Depreciation Expense.

The diagram below shows how an income state is derived from a hierarchy.



The hierarchy above would be converted into a flattened hierarchy and stored in W_HIERARCHY_D in the following format:

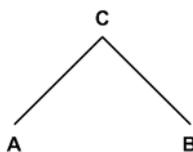
Table 3-4 Example of Flattened Hierarchy Stored in W_HIERARCHY_D

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 via 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 (see diagram below).



About the ETL Process for Oracle FSG Report

Before you run the ETL process for GL accounts, you need to 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 `\OracleBI\dwrep\Informatica\SrcFiles`.

Figure 3–4 Screen Shot of `file_gl_hierarchy_assignment_ora.csv` Opened in a Text Editor

	A	B	C	D	E	F
1	CHART_OF_ACCOUNTS	Hierarchy1_AXIS_SET_ID	Hierarchy2_AXIS_SET_ID	Hierarchy3_AXIS_SET_ID	Hierarchy4_AXIS_SET_ID	Hierarchy5_AXIS_SET_ID
2	101	1003	1922	1903	1924	1000
3	50194	1306		1245	2587	2589
4	50195	1754	1756	2001	2003	
5	50214		1744	1746	1749	1757
6	50234	1758				
7	50274	1758				
8	50317	2330	2331	2424		
9	50558		2107			
10	50560	2647	2648			
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						

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 need to 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 the following fields set as below:

CHART OF ACCOUNTS - 101

HIER1_AXIS_SET_ID - 1001

HIER3_AXIS_SET_ID - 1003

DATASOURCE_NUM_ID - 2

(Leave the other row values blank.)

This row indicates that for all of the GL accounts with `CHART_OF_ACCOUNTS=101` and `DATASOURCE_NUM_ID=2`, assigning hierarchies with `axis_set_id=1001`, null,

1003, null, null, null to HIER1~HIER6 columns respectively. Therefore, after extraction and loading, for those affected GL account rows, HIER1 column will be the foreign key to 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 file_gl_hierarchy_assignment_ora.csv to specify the hierarchies you want to reference for each CHART_OF_ACCOUNTS.
 - a. Navigate to OraceBI\dwrep\Informatica\SrcFiles.
 - b. Open file_gl_hierarchy_assignment_ora.csv in a text editor.
 - c. Specify the segments you want to analyze.
2. In DAC, do the following:
 - a. Go to the Design view, and select your custom container from the drop-down list.
 - b. Click the Subject Areas tab, and query for 'Financials – General Ledger'.
 - c. In the Configuration Tags subtab, do the following:
 - * Query for the tag 'Oracle – Extract FSG Hierarchies', and make sure the Inactive check box is not selected.
 - * Query for the tag 'Oracle – Extract Value Set 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. 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 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 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 layer, look for all the logical fact table that has logical join to the logical hierarchy table Dim - FSG Hierarchy1. You will need to create a similar logical join between the new logical hierarchy dimensions that you created and these logical facts, as follows:
 - a. Under each of the logical fact table, open the logical table sources and go to the content tab. Under aggregation content, select the 'Show unmapped' check box. It will show all the hierarchies you created in the previous step. For each of these hierarchies, select the logical level to 'Detail'.
 - b. In the business model diagram, create a new complex join between each of the new logical hierarchy tables and each of the logical facts. In the join, make sure the cardinality is (0,1) on the dimension side and N on the fact side.
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.

3.1.6 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 Num IDs"](#)

3.1.6.1 How to Configure Data Source Num IDs

DATASOURCE_NUM_ID is a system column in the Oracle Business Analytics Warehouse that uniquely identifies a data source category and indicates from which source systems the data comes. For example, the value '1' indicates one of the Siebel data sources, and the value 2 indicates an Oracle 11.5.8. data source. [Table 3–5](#) shows the data sources that are supported by Oracle BI Applications and their associated DATASOURCE_NUM_ID values.

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. If you create a new data source without using one of the pre-defined templates, you must specify the correct

DATASOURCE_NUM_ID for that data source category. For example, if you specify an Oracle EBS data source, you must specify the DATASOURCE_NUM_ID value '9'. For more information about specifying data sources, see *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

Table 3–5 Data Sources and Associated DATASOURCE_NUM_ID Values

Data Source Name	Data Source Number
ORA_11_5_8	2
ORA_11_5_9	5
ORA_11_5_10	4
ORACLE_R12	9
PSFT_8_4_FINSCM	7
PSFT_8_8_FINSCM	8
PSFT_8_8_HCM	6
SEBL_63	1
SEBL_771	1
SEBL_753	1
SEBL_78	1
SEBL_80	1
SEBL_VERT_771	1
SEBL_VERT_753	1
SEBL_VERT_78	1
SEBL_VERT_80	1
SRV_SEBL_80	1
UNIV	3

To configure a DATASOURCE_NUM_ID value

1. In DAC, go to the Setup view, and click the Physical Data Sources tab.
2. Select the appropriate data source from in list.
3. On the Edit subtab, edit the value in the Data Source Number field.
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"](#).

3.2 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, "Configure the Master Inventory Organization in Product Dimension Extract for Oracle 11i Adapter \(Except for GL & HR Modules\)"](#)
- [Section 3.2.1.3, "How to Map Oracle GL Natural Accounts to Group Account Numbers"](#)
- [Section 3.2.1.4, "How to Do Group Account Correction for Oracle Applications"](#)

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.

Oracle EBS enables users to group the products using categories and category sets. While Oracle EBS provides pre-packaged category sets, you can also define your own category sets.

By default, the Oracle Business Analytics Warehouse extracts product categories where the CATEGORY_SET_ID is 2 or 27. However, it is likely that the categories you extract from the source system are different from these prepackaged category sets.

To configure your product category sets, do the following:

- Identify the category sets you want to report (for more information, see [Section 3.2.1.1.1, "How to Identify Category Sets from 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"](#)).
- Configure the hierarchy mapping with the segments (for more information, see [Section 3.2.1.1.3, "How to Configure the Hierarchy Mapping with the Segments"](#)).

3.2.1.1.1 How to Identify Category Sets from 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.
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 note down 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 your custom container from the drop-down list.
2. Click the Source System Parameters tab.
3. Locate the INVPROD_CAT_SET_ID and PROD_CAT_SET_ID 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 INVPROD_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.
 - b. Click Save.

Note: The grain of the Product dimension is at the Master level. Therefore, the category set chosen as a value for the Product Dimension parameter (PROD_CAT_SET_ID) must be a Category Set controlled at a Master level but not at the Org level.

3.2.1.1.3 How to Configure the Hierarchy Mapping with the Segments 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 hierarchy mapping with the segments

1. In Informatica PowerCenter Designer, open the appropriate Oracle Applications 11i folder, and expand the Mapplets folder.
2. In the Maplet Designer, edit the mplt_SA_ORA_ProductDimension maplet.
3. Double-click the EXP_PRODUCTS Expression transformation to display the Edit Transformations dialog box.
4. In the Ports tab, scroll down to find the hierarchy code ports named EXT_PROD_HIER<X>_CODE.

Hierarchy levels are named with the convention EXT_PROD_HIER<X>_CODE, where <X> denotes the level within the hierarchy. For example, if you want to edit the first level of your hierarchy, you must edit the definition for EXT_PROD_HIER1_CODE port.

The first two levels are pre-configured as follows:

```
EXT_PROD_HIER1_CODE = IIF(ISNULL(INP_SEGMENT1) OR ISNULL(INP_SEGMENT2), NULL, INP_SEGMENT1 || '~' || INP_SEGMENT2)
```

```
EXT_PROD_HIER2_CODE = INP_SEGMENT1
```

5. Click the Expression field for the appropriate EXT_PROD_HIER<X>_CODE port, then click the down arrow to display the Expression Editor, and modify the expression.
6. Validate and save your changes to the repository.

Notes

- The default product hierarchy level is two. If you only have one product hierarchy level, you need to make the following modifications:
 1. In the EXP_PRODUCTS Expression transformation in the mplt_SA_ORA_ProductDimension mapplet, modify the EXT_PROD_HIER1_CODE port to INP_SEGMENT1.
 2. In the SDE_ORA_ProductDimension mapping, open the EXP_W_PRODUCT_DS transformation expression and modify the PROD_HIER1_CODE port expression to IN_PROD_HIER1_CODE.
 3. If you have more than two hierarchy levels, then map the other columns from the appropriate source columns.

3.2.1.2 Configure the Master Inventory Organization in Product Dimension Extract for Oracle 11i Adapter (Except for GL & HR Modules)

In Oracle 11i applications, the products are defined in a Master Organization and then copied into the other Inventory Organizations for transactions. The Product dimension Extract mapping 'SDE_ORA_ProductDimension_Derive' has been enabled for configuration of this Master Organization based on the configuration in the OLTP. By default, the organization ID (that is set by the \$\$MASTER_ORG parameter) is set to 204. This organization ID 204 needs to be changed based on the individual implementation of OLTP in your deployment.

Note: This ETL implementation supports the best practice prescribed by Oracle for the creation of Single Master Organization for defining the Product master. This ETL implementation does not support the multiple master Organizations if the same product is defined in multiple master organizations.

To set the Master Inventory Organization in the Product dimension extract

1. In DAC, go to the Design view, and select your custom container from the drop-down list.
2. Click the Tasks tab, and query for the SDE_ORA_ProductDimension_Derive task.
3. Specify or create the \$\$MASTER_ORG parameter with an appropriate value.
For example, \$\$MASTER_ORG=204.
4. Save your changes.

3.2.1.3 How to Map Oracle GL Natural Accounts to Group Account Numbers

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*.

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 values you can use. For a list of the Group Account Number domain values, see *Oracle Business Analytics Warehouse Data Model Reference*. The mappings to General Ledger Accounts Numbers are important for both Profitability analysis and General Ledger analysis (for example, Balance Sheets).

The logic for assigning the accounts is located in the `file_group_acct_codes_ora.csv` file. The table below shows an example configuration of the `file_group_acct_codes_ora.csv` file.

Table 3-6 Example Configuration of `file_group_acct_codes_ora.csv`

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 the table above, in the first row, all accounts within the account number range from 101010 to 101099 that have a Chart of Account (COA) ID equal to 1 are assigned to Current Asset. Each row maps all accounts within the specified account number range and within the given chart of account ID.

If you need to create a new group of account numbers, you can create new rows in the `file_group_acct_names_ora.csv` file. You can then assign GL accounts to the new group of account numbers in the `file_group_acct_codes_ora.csv` file.

You also need to add a new row in the `file_gpract_fstmt.csv` file. This file specifies the relationship between a group account number and a Financial Statement Item code. You must map the new group account number to one of the following Financial Statement Item codes:

- AP
- AR
- COGS
- REVENUE
- TAX
- OTHERS

These Financial Statement Item codes correspond to the following six base fact tables in the Financial Analytics product.

- AP base fact (`W_AP_XACT_F`)
- AR base fact (`W_AR_XACT_F`)

- Revenue base fact (W_GL_REVN_F)
- Cost of Goods Sold base fact (W_GL_COGS_F)
- Tax base fact (W_TAX_XACT_F)
- GL Journal base fact (W_GL_OTHER_F)

By mapping your GL accounts against the group account numbers and then associating the group account number to a Financial Statement Item code, you have indirectly associated the GL account numbers to Financial Statement Item codes as well.

Financial Statement Item codes are internal codes used by 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.

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 \$pmsserver\srcfiles folder (for example, \PowerCenter8.1.1\server\infa_shared\SrcFiles).
2. Edit the fields in the following table:

Field Name	Description
CHART OF ACCOUNTS ID	The ID of the GL chart of account.
FROM ACCT and TO ACCT	The natural account range. This is based on the natural account segment of your GL accounts.
GROUP_ACCT_NUM	This field denotes the nature of the Oracle General Ledger accounts. For example, cash account, payroll account, and so on. Refer to the file_group_acct_names.csv file for values you can use.

Note: It is important that you do not edit any other fields in the CSV files.

3. Save and close the CSV file.

3.2.1.4 How to Do Group Account Correction for Oracle Applications

Note: Refer to the section [Section 3.2.1.3, "How to Map Oracle GL Natural Accounts to Group Account Numbers"](#) for general concepts about group account number and Financial Statement Item code.

When a user mistakenly maps a GL natural account number to an incorrect group account number, incorrect accounting entries might be inserted into the fact table. For example, natural account 1210 is classified as belonging to 'AR' group account number when it should be classified as having 'AP' group account number. When this happens, the ETL program will get all the journal lines charging to account 100 and try to reconcile these journal lines against subledger accounting records in the AR fact table (W_AR_XACT_F). Since these journal lines did not come from AR, the ETL program will not be able to find the corresponding matching subledger accounting records for these 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 GL system charging against the AR accounts. This entire process is call GL reconciliation process.

In order to revert these 'Manual' entries in the AR fact, you will need to utilize the 'Group Account Number Cleanup' program provided in Oracle 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 matching subledger accounting records in the AP fact (W_AP_XACT_F); provided that you've re-assign the natural account 1210 to the 'AP' group account number in the file_group_acct_codes_ora.csv file.

To do group account correction

1. Correct the mapping of GL natural account to the group account in the input 'csv' file 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 Acct Num and after correcting the GL natural account to the group account, the CSV file would have the following values:

```
CHART OF ACCOUNTS ID = 101
FROM ACCT = 1210
TO ACCT = 1210
GROUP_ACCT_UM = AR
```

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.
 - c. Query for the 'Financials – General Ledger' subject area. Click the Configuration Tags subtab and verify which of the following Configuration Tags is marked as 'inactive':

- d. Click the Configuration Tags subtab, and verify 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 Cleanup' subject area and do the following:
 - If in the step above, 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 step above, 'Oracle – Extract GL Balance' is marked as 'Inactive', then 'Financials – Calculate GL Balance Reverse' should be active, that is, the check box should not be selected.
3. If you need to make any change in the step above, then you need to reassemble the 'Financials – Group Account Number Clean Up' subject area and then rebuild the appropriate execution plans, depending on which version of Oracle EBS you have:
 - Financials – Group Account Number Clean Up ORA1158
 - Financials – Group Account Number Clean Up ORA1159
 - Financials – Group Account Number Clean Up ORA11510
 - Financials – Group Account Number Clean Up R12
 4. Depending on the Oracle EBS version you have, run the corresponding Group Account Cleanup execution plan:
 - Financials – Group Account Number Clean Up ORA1158
 - Financials – Group Account Number Clean Up ORA1159
 - Financials – Group Account Number Clean Up ORA11510
 - Financials – Group Account Number Clean Up R12

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 topics:

- [Section 3.2.2.1, "How to Configure the Country Region and State Region Name"](#)
- [Section 3.2.2.2, "How to Configure the State Name"](#)
- [Section 3.2.2.3, "How to Configure the Country Name"](#)
- [Section 3.2.2.4, "How to Configure the Make-Buy Indicator"](#)

3.2.2.1 How to Configure the Country Region and State Region Name

For Oracle 11i, you can reconfigure the region, state, and country names. This configuration information applies only to plant, storage, and supplier locations. By default, the Region Name column (EXT_REGION_NAME) is populated using the same code value as the Region Code column (EXT_REGION_CODE). However, you can redefine the load mapping's source adapter mapplet to load a source-supplied region name instead of the code. If you want to reconfigure the load in this manner, you can load the region code and region name into the W_CODE_D table. For

information on loading codes and code names into the W_CODE_D table, see *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

When you have loaded the region code and region name into the W_CODE_D table, you can remove the expression in the source adapter that defines the Region Name column. By making the Region Name's expression blank, the SIL looks up the Region Name in the W_CODE_D table, using the supplied region code when the load occurs. The load mapping then inserts the region name and region code into the data warehouse table.

To configure the Country Region Name

1. In Informatica PowerCenter Designer, open the Configuration for SDE_ORA1158_adapter folder.
2. Open the maplet you want to edit.
The following is a list of all source adapter maplets that use the EXT_COUNTRY_REGION column:
 - mplt_SA_ORA_SupplierDimension
 - mplt_SA_ORA_BusinessLocationDimension_Plan
 - mplt_SA_ORA_BusinessLocationDimension_StorageLocation
3. Double-click the Expression transformation to open the Edit Transformations dialog, and click the Port tab to display the EXT_COUNTRY_REGION port.
4. Edit the condition by removing the assigned value if you want the lookup to occur.
5. Click Apply.
6. Validate the maplet, and save your changes to the repository.

To configure the State Region Name

1. In Informatica PowerCenter Designer, open the SDE_ORA<Ver>_adapter.
2. Open the maplet you want to edit.
The following is a list of all source adapter maplets that use the EXT_STATE_REGION column:
 - mplt_SA_ORA_SupplierDimension
 - mplt_SA_ORA_BusinessLocationDimension_Plan
 - mplt_SA_ORA_BusinessLocationDimension_StorageLocation
3. Double-click the Expression transformation to open the Edit Transformations dialog, and click the Port tab to display the EXT_STATE_REGION port.
4. Edit the condition by removing the assigned value if you want the lookup to occur.
5. Click Apply.
6. Validate the maplet, and save your changes to the repository.

3.2.2.2 How to Configure the State Name

For Oracle 11i, you can reconfigure the region, state, and country names that apply to the Supplier locations only. By default, the State Name column (EXT_STATE_NAME) is populated using the same code value as the State Code column (EXT_STATE_CODE). However, you can redefine the load mapping's source adapter maplet to load a source-supplied state name instead of the code. If you want to reconfigure the load in this manner, you can load the state code and state name into the W_CODE_D table.

For information on loading codes and code names into the W_CODE_D table, see *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

When you have loaded the state code and state name into the W_CODE_D table, you can remove the expression in the source adapter that defines the State Name column. By setting the State Name's expression to null, the SIL looks up the state name in the W_CODE_D table using the supplied state code, during the load process. The load mapping then inserts the state name and state code into the data warehouse table.

To configure the State Name

1. In Informatica PowerCenter Designer, open the SDE_ORA<Ver>_adapter.
2. Open one of the following mapplets:
 - mplt_SA_ORA_SupplierDimension
 - mplt_SA_ORA_BusinessLocationDimension_Plant
 - mplt_SA_ORA_BusinessLocationDimension_StorageLocation
3. Double-click the Expression transformation to open the Edit Transformations dialog, and click the Port tab to display the EXT_STATE_NAME port.
4. Edit the condition by removing the assigned value if you want the lookup to occur.
5. Click Apply.
6. Validate the mapplet and save your changes to the repository.

3.2.2.3 How to Configure the Country Name

For Oracle 11i, you can reconfigure the region, state, and country names that apply to supplier locations only. By default, the Country Name column (EXT_COUNTRY_NAME) is populated using the same code value as the Country Code column (EXT_COUNTRY_CODE). However, you can redefine the load mapping's source adapter mapplet to load a source-supplied country name instead of the code. If you want to reconfigure the load in this manner, you can load the country code and country name into the W_CODE_D table. For information on loading codes and code names into the W_CODE_D table, see *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

When you have loaded the country code and country name into the W_CODE_D table, you can remove the expression in the source adapter that defines the Country Name column. By setting the Country Name's expression to null, when the load occurs, the SIL looks up the country name in the W_CODE_D table, using the supplied country code. The load mapping then inserts the country name and country code into the data warehouse table.

To configure the Country Name

1. In Informatica PowerCenter Designer, open the SDE_ORA<Ver>_adapter.
2. Open one of the following mapplets:
 - mplt_SA_ORA_SupplierDimension
 - mplt_SA_ORA_BusinessLocationDimension_Plant
 - mplt_SA_ORA_BusinessLocationDimension_StorageLocation
3. Double-click the Expression transformation to open the Edit Transformations dialog, and click the Port tab to display the EXT_COUNTRY_NAME port.
4. Edit the condition by removing the assigned value if you want the lookup to occur.

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

3.2.2.4 How to Configure the Make-Buy Indicator

The Make-Buy indicator specifies whether a material that was used to manufacture a product was made in-house or bought from an outside vendor. By default, the indicator is set using the INP_PLANNING_MAKE_BUY_CODE. If the code is set to 1, then the indicator is set to M (for make). However, if the code is set to 2, then the indicator is set to B (for buy). Otherwise, the indicator is set to null.

Your organization may require different indicator codes. If so, you can modify the indicator logic by reconfiguring the condition in the 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 `SDE_ORA<Ver>_adapter`.
2. Open one of the following mapplets:
 - `mplt_SA_ORA_SupplierDimension`
 - `mplt_SA_ORA_BusinessLocationDimension_Plant`
 - `mplt_SA_ORA_BusinessLocationDimension_StorageLocation`
3. Double-click the Expression transformation to open the Edit Transformations dialog, and click the Port tab to display the `EXT_COUNTRY_NAME` port.
4. Edit the condition by removing the assigned value if you want the lookup to occur.
5. Click Apply.
6. 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

Not applicable to Oracle BI Applications release 7.9.5.1.

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, "Internal Organization Hierarchy Extraction Customization"](#)

3.3.2.1 Internal Organization Hierarchy Extraction Customization

Internal Organization Hierarchy extracts the hierarchy related to business units. The PeopleSoft Tree Name and SETID are the only settings that require modification.

Table 3–7 Internal Organization Hierarchy Mappings

Exec Sequence	Mapping Name	Needs Customization?
1	SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_Extract	Y
1	SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_GetGLBU	N
2	SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_GetHierarchyLevels	N
2	SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_DeriveRange	N
3	SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_Flatten	N
4	SDE_PSFT_InternalOrganizationDimension_CompanyHierarchy_Derive	N

As the above table illustrates, only the SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_Extract mapping needs to be modified (that is, there is no need to change or copy the other mappings). For example, if you want to extract a tree named 'BIZ_UNIT_HIER' with SETID 'XYZ', which uses BU_GL_UPDAT2_VW as its structure's Record Name, and BUSINESS_UNIT as its field name (that means it builds up hierarchy for BUSINESS_UNIT), do the following:

1. In Informatica PowerCenter Designer, create a new mapping by copying the existing SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_Extract mapping.

Name the new mapping SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy2_Extract.
2. Open the new mapping SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy2_Extract and edit as follows:
 - a. In SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy2_Extract, open the source definition PSFT_TREE_VERTICAL_FLATTENING.
 - b. Under the Properties tab, change the Tree Name to 'BIZ_UNIT_HIER', and Set ID to 'XYZ'.
3. Save the changes in the Informatica folder.
4. In Informatica PowerCenter Workflow Manager, add Corresponding Sessions and Workflows, as follows:
 - a. In Informatica PowerCenter Workflow Manager, locate the appropriate PeopleSoft folder SDE_PSFT_xx_adapter (for example, 'SDE_PSFT_89_adapter', 'SDE_PSFT_90_adapter').
 - b. Open Task Developer, and create new task for the mappings that you just created.
 - c. Using the parameters in the original sample sessions as a guide (refer to the parameter settings for SDE_PSFT_Stage_GLHierarchy_DepartmentHierarchy_Extract), modify the newly created session.
 - d. Save your changes.
 - e. In Informatica PowerCenter Workflow Manager, create a Workflow with the same name as the Session.

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"](#)

4.1 Overview of Oracle Procurement and Spend Analytics

Oracle Procurement and Spend Analytics comprises the following:

- Oracle Procurement and Spend Analytics (for more information, see [Section 4.1.1, "Oracle Procurement and Spend Analytics Module"](#)).
- Oracle Supplier Performance Analytics (for more information, see [Section 4.1.2, "Supplier Performance Analytics Module"](#)).

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 procurement and spend process, including comprehensive supplier performance analysis and supplier payables analysis. Through complete end-to-end insight into the factors that impact procurement and spend performance, organizations can significantly reduce costs, enhance profitability, increase customer satisfaction, and gain competitive advantage. Oracle Procurement and Spend Analytics also integrate with the other applications in Oracle BI Applications product line. They deliver this insight across the organization to increase the company's effectiveness in managing its customers, suppliers, and financial decisions.

4.1.1 Oracle Procurement and Spend Analytics Module

This module provides visibility into direct and indirect spending across the enterprise, payment, and employee expenses. The Oracle Procurement and Spend Analytics application comprises the following subject areas:

- **Total Spend:** This is a summary subject area that provides the ability to do comparative analysis and report on requested spend, committed spend and actual spend across suppliers, company, products, commodities and associated hierarchies for both direct and indirect spend (indirect spend being MRO and

employee expenses) in detail to allow complete visibility of spending across your organization.

- **Purchase Orders:** This is a detailed subject area that provides the ability to report on committed spend, and Purchase orders of the suppliers of an organization across suppliers, company, products, commodities and associated hierarchies at purchase order line level
- **Purchase Order Costs:** This is a detailed subject area that provides the ability to report on committed spend and purchase orders of the suppliers of an organization across suppliers, company, products, and commodities and associated hierarchies at cost center (distribution line) level.
- **Purchase Cycle Lines:** This is a summary subject area that provides the ability to report cycle time performance, such as requisition to purchase order lead time, purchase order to receipt lead time, P2P lead time of the suppliers of an organization.
- **Purchase Schedules:** This is a detailed subject area that provides the ability to report on purchase order shipments of an organization across suppliers, company, products, commodities and associated hierarchies at purchase schedule line level
- **Purchase Requisitions:** This is a detailed subject area that provides the ability to report on requested spend and purchase requisitions of the suppliers of an organization across suppliers, company, products, commodities and associated hierarchies at purchase requisition line level
- **Purchase Requisition Status:** This is a summary subject area that provides the ability to report on requisition status along the approval cycle of purchase requisitions of the suppliers of an organization. It's populated only by Universal adapter.
- **Purchase Receipts:** This is a detailed subject area that provides the ability to report on actual spend and purchase receipts of the suppliers of an organization across suppliers, company, location, products, commodities and associated hierarchies at purchase receipt line level
- **Employee Spend:** This is a detailed subject area that provides the ability to report on employee spend of an organization across employees, company, cost center and associated hierarchies. The Expenses subject area contains targeted metrics and reports that examine travel and expense costs in relationship to your organization's overall spending patterns. In contrast to analyzing direct spending patterns, where you may review purchasing, Expenses examines indirect spending—the cost of employee related expenses. It's populated only by Universal adapter.

4.1.2 Supplier Performance Analytics Module

Enables organizations to have a complete picture of the performance of their suppliers, including complete supplier scorecards, procurement cycle times, supplier price performance, delivery performance, product receipt quality, on-time payment ratings, payment activity and volume, and analysis of payments due and overdue.

The Supplier Performance Analytics application comprises the following subject areas:

- **Supplier Performance.** The Suppliers functional area contains targeted reports and metrics that allow you to analyze the timeliness, reliability, cost, and quality of goods provided by your suppliers. It helps you to understand how well suppliers are contributing to success of your organization, and to evaluate the price, quality, and delivery timing in procuring materials

- **Supplier AP Transactions:** This is a summary subject area that provides the ability to analyze payment performance and payment due analysis of the suppliers of an organization across suppliers, company, location, products, commodities and associated hierarchies. In addition to monitoring supplier performance, it is important to monitor organization's performance of making on time payments. This will help the Organizations to maintain better relationships with their best suppliers.

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

This section contains configuration steps that you need to perform on Oracle Procurement and Spend Analytics before you do a full data load. 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"](#)
- [Section 4.2.4, "Configuration Steps for Oracle Procurement and Spend Analytics for Universal"](#)
- [Section 4.2.5, "Configuration Steps for Controlling Your Data Set"](#)

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.

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

4.2.1.1 How to Deploy Stored Procedures

Stored procedures are a group of SQL statements that perform particular tasks on the database. For example, stored procedures can help to improve the performance of the database.

You can deploy stored procedures by copying the stored procedure files from your Oracle Business Intelligence installation and deploying them to the target data warehouse.

Note: Some sessions may fail if these procedures are not compiled in the database before running the workflows.

To deploy stored procedures

1. Navigate to the OracleBI\dwrep\Informatica\Stored_Procedure_Scripts folder.

2. Copy the source code in Compute_Bounds_Ora11i.sql file 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.

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"](#)

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 data coming from Oracle 11i applications from data from other applications such as SAP and PeopleSoft.

To configure the DAC parameter for purchase cycle line

1. In DAC, go to the Design view, then click the Tasks tab, and query for task SIL_PurchaseCycleLinesFact.
2. Click the Parameters tab in the lower pane. Change the values for parameter \$\$ORA_DATASOURCE_NUM_ID_LIST from (2,4,5,9) to the list of Data Source NUM ID you defined for your Oracle connections.
3. Repeat the same 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 modify or extend a seeded list of values, you must configure the CSV files for Oracle Procurement and Spend Analytics by mapping values from your source systems to the domain values.

This section explains how to extract the lists of values from your source system, which you then compare with the seeded values. If the lists of values are different from the seeded values, you need to follow the instructions for configuring the domain values and CSV worksheet files in this section.

4.2.2.2.1 List of Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics The table below lists the CSV worksheet files and the domain values for Oracle Procurement and Spend Analytics in the `$pmsserver\LkpFiles` folder.

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

Worksheet File Name	Description	Session
domainValues_PaymentType_ora12.csv	Lists the Payment Type and the corresponding domain values for the Oracle 11i application. For information about how to edit this file, see Section 4.2.2.2.2, "To configure domainValues_PaymentType_ora12.csv" .	SDE_ORA_TransactionTypeDimension_PO_Payment_Type
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 4.2.2.2.9, "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 4.2.2.2.3, "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 4.2.2.2.4, "To configure domainValues_Status_Purch_Cycle_ora11i.csv" .	SDE_ORA_StatusDimension_PurchaseCycle
domainValues_Xact_Types_PO_Line_Type_CODE1_ora11i.csv	List the Purchase Basis Type and the corresponding domain values for the Oracle EBS application. For information about how to edit this file, see Section 4.2.2.2.8, "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 4.2.2.2.7, "To configure domainValues_Xact_Types_PO_Line_Type_ora11i.csv" .	SDE_ORA_TransactionTypeDimension_PO_Line_Type
domainValues_Xact_Types_Purch_Orders_ora11i.csv	Lists the Purchase Order Transaction Type column and the corresponding domain values for the Oracle EBS application. For information about how to edit this file, see Section 4.2.2.2.5, "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 4.2.2.2.6, "To configure domainValues_Xact_Types_Purch_Requisitions_ora11i.csv" .	SDE_ORA_TransactionTypeDimension_PurchaseRequest

4.2.2.2.2 To configure domainValues_PaymentType_ora12.csv

This section explains how to configure domainValues_PaymentType_ora12.csv. This file is only applicable to Oracle R12.

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 \$pmsserver\lkpfiles folder, 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.

4.2.2.2.3 To configure domainValues_Status_Purch_Approve_ora11i.csv This section explains how to configure domainValues_Status_Purch_Approve_ora11i.csv.

1. Identify the Purchase Approval Status in your Oracle EBS source system by using the following SQL:

```
SELECT A.LOOKUP_CODE FROM FND_LOOKUP_VALUES A WHERE
A.LOOKUP_TYPE='AUTHORIZATION STATUS' AND A.LANGUAGE = 'US'
```

2. From 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.

4.2.2.2.4 To configure domainValues_Status_Purch_Cycle_ora11i.csv This section explains how to configure domainValues_Status_Purch_Cycle_ora11i.csv.

1. Identify the Purchase Cycle Status in your Oracle EBS source system by using the following SQL:

```
SELECT A.LOOKUP_CODE FROM FND_LOOKUP_VALUES A WHERE
A.LOOKUP_TYPE='DOCUMENT STATE' AND A.LANGUAGE = 'US'
```

2. From the \$pmsserver\lkpfiles folder, 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.

4.2.2.2.5 To configure domainValues_Xact_Types_Purch_Orders_ora11i.csv This section explains how to configure domainValues_Xact_Types_Purch_Orders_ora11i.csv.

1. Identify the Purchase Order Types in your Oracle EBS source system by using the following SQL:

```
SELECT DISTINCT PO_DOCUMENT_TYPES_ALL_TL.DOCUMENT_SUBTYPE
FROM PO_DOCUMENT_TYPES_ALL_TL

WHERE PO_DOCUMENT_TYPES_ALL_TL.LANGUAGE='US' AND PO_DOCUMENT_
TYPES_ALL_TL.DOCUMENT_TYPE_CODE IN ('PO', 'PA') AND PO_
DOCUMENT_TYPES_ALL_TL.DOCUMENT_SUBTYPE <> 'CONTRACT'

UNION SELECT 'COMPLEXWORK_ACTUAL' FROM DUAL

UNION SELECT 'COMPLEXWORK_FINANCING' FROM DUAL;
```

2. From the \$pmsserver\lcpfiles folder, 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.

4.2.2.2.6 To configure domainValues_Xact_Types_Purch_Requisitions_ora11i.csv This section explains how to configure domainValues_Xact_Types_Purch_Requisitions_ora11i.csv.

1. Identify the Purchase Requisition Type in your Oracle EBS source system by using the following SQL:

```
SELECT DISTINCT PO_DOCUMENT_TYPES_ALL_TL.DOCUMENT_SUBTYPE
FROM PO_DOCUMENT_TYPES_ALL_TL

WHERE PO_DOCUMENT_TYPES_ALL_TL.LANGUAGE='US' AND PO_DOCUMENT_
TYPES_ALL_TL.DOCUMENT_TYPE_CODE = 'REQUISITION'
```

2. From the \$pmsserver\lcpfiles folder, 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.

4.2.2.2.7 To configure domainValues_Xact_Types_PO_Line_Type_ora11i.csv This section explains how to configure domainValues_Xact_Types_PO_Line_Type_ora11i.csv.

1. Identify the Purchase Order Line Type in your Oracle EBS source system by using the following SQL:

```
SELECT DISTINCT PO_LINE_TYPES_V.ORDER_TYPE_LOOKUP_CODE FROM
PO_LINE_TYPES_V
```

2. From the \$pmsserver\lkpfiles folder, 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.

4.2.2.2.8 To configure domainValues_Xact_Types_PO_Line_Type_CODE1_ora11i.csv This section explains how to configure domainValues_Xact_Types_Code1_PO_Line_Type_ora11i.csv.

1. Identify the Purchase Order Line Purchase Basis in your Oracle EBS source system by using the following SQL:

```
SELECT DISTINCT PO_LINE_TYPES_V.PURCHASE_BASIS FROM PO_LINE_
TYPES_V
```

2. From the \$pmsserver\lkpfiles folder, 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.

4.2.2.2.9 To configure domainValues_ShipmentType_ora12.csv This section explains how to configure domainValues_ShipmentType_ora12.csv. 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 \$pmsserver\lkpfiles folder, 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.

4.2.3 Configuration Steps for Oracle Procurement and Spend Analytics for PeopleSoft

Not applicable to Oracle BI Applications release 7.9.5.1.

4.2.4 Configuration Steps for Oracle Procurement and Spend Analytics for Universal

Not applicable to Oracle BI Applications release 7.9.5.1.

4.2.5 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle Procurement and Spend Analytics.

- [Section 4.2.5.1, "Configuration Steps for Oracle Procurement and Spend Analytics for All Source Systems"](#)
- [Section 4.2.5.2, "Configuration Steps for Oracle Procurement and Spend Analytics for Oracle EBS"](#)
- [Section 4.2.5.3, "Configuration Steps for Oracle Procurement and Spend Analytics for PeopleSoft"](#)
- [Section 4.2.5.4, "Configuration Steps for Oracle Procurement and Spend Analytics for Universal"](#)

4.2.5.1 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 4.2.5.1.1, "About Configuring the Purchase Receipts Aggregate Table"](#)
- [Section 4.2.5.1.2, "How to Configure the Purchase Receipts Aggregate Table"](#)
- [Section 4.2.5.1.3, "About Configuring the Purchase Cycle Lines Aggregate Table"](#)
- [Section 4.2.5.1.4, "How to Configure the Purchase Cycle Lines Aggregate Table"](#)

4.2.5.1.1 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. The possible values for the parameter are:

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

The Purchase Receipt Lines aggregate table is fully loaded from the base table in the initial 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. This process is done in four steps:

1. Oracle Business Analytics Warehouse finds the records to be deleted in the base table since the last ETL run, and loads them into the `W_PURCH_RCPT_TMP` table. The measures in these records are multiplied by (-1). The mapping responsible for this task is suffixed with `_Derive_PreSoftDeleteImage`, and it is run before the records are deleted from the base table. The mapping is run in the source-specific workflow.
2. Oracle Business Analytics Warehouse finds the records to be updated in the base table since the last ETL run, and loads them into the `W_PURCH_RCPT_TMP` table. The measures in these records are multiplied by (-1). The mapping responsible for this task is suffixed with `_Derive_PreLoadImage`, and it is run before the records are updated in the base table. It is run in the source-specific workflow.
3. Oracle Business Analytics Warehouse finds the inserted or updated records in the base table since the last ETL run, and loads them into the `W_PURCH_RCPT_TMP` table, without changing their sign. The mapping responsible for this task is suffixed with `_Derive_PostLoadImage`, and it is run after the records are updated or inserted into the base table. It is run in the post load-processing workflow.
4. Oracle Business Analytics Warehouse aggregates the `W_PURCH_RCPT_TMP` table, and joins it with the `W_PURCH_RCPT_A` aggregate table to insert new or update existing buckets to the aggregate table. This step is part of the post load-processing workflow, and the mapping is suffixed with `_Derive`.

4.2.5.1.2 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 your custom container from the drop-down list.
2. Click the Tasks tab.
3. For each of the following tasks, click the Parameters subtab and create the specified parameter name and parameter value
 - `SIL_PurchaseReceiptAggregate_Derive_PreLoadImage $$TIME_GRAIN 'MONTH'`

- SIL_PurchaseReceiptAggregate_Derive_PreSoftDeleteImage \$\$TIME_GRAIN 'MONTH'
- PLP_PurchaseReceiptAggregate_Derive_PostLoadImage \$\$TIME_GRAIN 'MONTH'
- PLP_PurchaseReceiptAggregate_Load_Full \$\$GRAIN 'MONTH'

To configure the Purchase Receipts aggregate table for Universal Source

1. Using a text editor, open the file parameterfileDW.txt, located in OracleBI\DAC\Informatica\parameters\input.
2. Replace the default parameter values with your new values.
3. Save and close the file.

4.2.5.1.3 About Configuring the Purchase Cycle Lines Aggregate Table To aggregate the Purchase Cycle Lines table (W_PURCH_CYCLE_LINE_A), you need to configure the parameterfiledw.txt parameters file and the source system parameters file, and run the initial ETL workflow and then the incremental ETL workflow.

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 has a preconfigured value of Month. The possible values for parameters are:

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

The Purchase Cycle Lines aggregate table is fully loaded from the base table in the initial 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. This process is done in four steps:

1. Oracle Business Analytics Warehouse finds the records to be deleted in the base table since the last ETL run, and loads them into the W_PURCH_CYCLE_LINE_TMP table. The measures in these records are multiplied by (-1). The mapping responsible for this task is suffixed with _Derive_PreSoftDeleteImage, and it is run before the records are deleted from the base table. It is run in the source-specific workflow.
2. Oracle Business Analytics Warehouse finds the records to be updated in the base table since the last ETL run, and loads them into the W_PURCH_CYCLE_

LINE_TMP table. The measures in these records are multiplied by (-1). The mapping responsible for this task is suffixed with _Derive_PreLoadImage, and it is run before the records are updated in the base table. It is run in the source-specific workflow.

3. Oracle Business Analytics Warehouse finds the inserted or updated records in the base table since the last ETL run, and loads them into the W_PURCH_CYCLE_LINE_TMP table, without changing their sign. The mapping responsible for this task is suffixed with _Derive_PostLoadImage, and it is run after the records are updated or inserted into the base table. It is run in the post load-processing workflow.
4. Oracle Business Analytics Warehouse aggregates the W_PURCH_CYCLE_LINE_TMP table, and joins it with the W_PURCH_CYCLE_LINE_A aggregate table to insert new or update existing buckets to the aggregate table. This step is part of the post load-processing workflow, and the mapping is suffixed with _Derive.

4.2.5.1.4 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 your custom container from the drop-down list.
2. Click the Tasks tab.
3. For each of the following tasks, click the Parameters subtab and create the specified parameter name and parameter value
 - SIL_PurchaseCycleLinesAggregate_Derive_PreLoadImage \$\$TIME_GRAIN 'MONTH'
 - SIL_PurchaseCycleLinesAggregate_Derive_PreSoftDeleteImage \$\$TIME_GRAIN 'MONTH'
 - PLP_PurchaseCycleLinesAggregate_Derive_PostLoadImage \$\$TIME_GRAIN 'MONTH'
 - PLP_PurchaseCycleLinesAggregate_Load_Full \$\$GRAIN 'MONTH'

4.2.5.2 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 4.2.5.2.1, "How to Extract Particular Purchase Order Records"](#)

4.2.5.2.1 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 maplet. 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_ORA115<Ver>_Adaptor.
2. Open the MPLT_BC_ORA_PURCHASEORDERFACT mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations box, and select the Properties tab to display the SQL Query.
4. Double-click the value in the SQL Query to open the SQL Editor box and edit the statement.
5. 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 MPLT_BC_ORA_PURCHASESCHEDULELINESFACT mapplet.

4.2.5.3 Configuration Steps for Oracle Procurement and Spend Analytics for PeopleSoft

Not applicable to Oracle BI Applications release 7.9.5.1.

4.2.5.4 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 4.2.5.4.1, "About Configuring Oracle Procurement and Spend Analytics for Universal Source"](#)
- [Section 4.2.5.4.2, "How to Configure the Preferred Merchant Flag"](#)
- [Section 4.2.5.4.3, "How to Configure the Customer Billable Indicator"](#)
- [Section 4.2.5.4.4, "How to Configure the Receipts Indicator"](#)
- [Section 4.2.5.4.5, "How to Configure Expense Payment Types"](#)
- [Section 4.2.5.4.6, "How to Configure the Default Expense Distribution Percentage"](#)
- [Section 4.2.5.4.7, "How to Configure Lookup Dates for Currency Conversion"](#)

4.2.5.4.1 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.

4.2.5.4.2 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 Configuration for Universal Source folder.
2. Open the SDE_Universal_ExpenseFact mapping.
3. Select the Expression transformation to open the Edit Transformations box and select the Port tab.
4. Select the Add Port icon to add the PREF_MERCHANT_FLAG port.
5. Enter preferred merchant flag logic.
6. Validate the maplet and click OK to exit.
7. Save your changes to the repository.

4.2.5.4.3 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 Configuration for Universal Source folder.
2. Open the SDE_Universal_ExpenseFact mapping.
3. Select the Expression transformation to open the Edit Transformations box, and select the Port tab.
4. Add a port called CUST_BILLABLE_FLG = <insert your expression here>.
5. Validate the mapplet and click OK to exit.
6. Save your changes to the repository.

4.2.5.4.4 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_ExpenseFact mapping.
2. Open the Exp_FILE_EXPENSE Expression transformation and select the Port tab.
3. 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.
4. Re-map the RECEIPT_FLG port to the target and validate the mapping.
5. Save your changes to the repository.

4.2.5.4.5 How to Configure Expense Payment Types The Oracle Business Analytics Warehouse supports analysis on three types of payment—Reimbursable Expense (type E), expenses prepaid by your company (type P), and cash advance (type C). All of your organization's payment types must be mapped to one of these types described earlier; do this by modifying MPLT_SAF_EXPENSES.

To configure additional payment types

1. In Informatica PowerCenter Designer, open the SDE_Universal_ExpenseFact mapping.
2. Open the MPLT_SAF_EXPENSES source adapter mapplet.
3. Select the Expression transformation to open the Edit Transformations box, and select the Port tab to display the EXPENSE_PAY_TYPE_ID port.
4. Copy and paste this port, rename it EXPENSE_PAY_TYPE_ID.

5. Select O to make your new port an output port.
6. Add a decode logic in the expression to decode source-supplied values to the Oracle Business Analytics Warehouse payment type of your choice.
7. Validate your mapping, and save your changes to the repository.

4.2.5.4.6 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 Configuration for Universal Source folder.
2. Open the SDE_Universal_ExpenseFact mapping.
3. Select the Expression transformation to open the Edit Transformations box and select the Port tab.
4. Add a port named `DIST_PERCENTAGE = <expression that sets this to 100%>`.
5. Validate your mapplet.
6. Save your changes.

4.2.5.4.7 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 Configuration for Universal Source folder.
2. Open the SDE_Universal_ExpenseFact mapping.
3. Select the Expression transformation to open the Edit Transformations box, and select the Port tab to display the `EXT_XRATE_LKP_DATE` port.
The expression contains the exchange rate lookup date logic.
4. Select the expression in the `EXT_XRATE_LOOKUP_DATE` port to open the Expression Editor box and edit the expression.
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.

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"](#)

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 allows you to evaluate the efficiency of your cash outflows. The need for analysis is increasingly important because suppliers are becoming strategic business partners with the focus on increased efficiency for just in time, and quality purchasing relationships.

The default configuration for the Oracle Payables Analytics application is based on what is identified as the most- common level of detail, or granularity. However, you can configure or modify the extracts to best meet your business requirements.

- **Oracle Receivables Analytics.** The Oracle Receivables Analytics application provides information to support your credit and collection activities, and to monitor and identify potential, receivables problems.

The information found in the Oracle Receivables Analytics application pertains to data found exclusively in the Accounts Receivable (AR) account grouping of your financial statements and chart of accounts. Each day that your receivables are past the due date represents a significant, opportunity-cost to your company. Keeping a close eye on the trends, and clearing of AR is one way to assess the efficiency of your sales operations, the quality of your receivables, and the value of key customers.

The default configuration for the Oracle Receivables Analytics application is based on what is identified as the most-common level of detail or granularity. However, you may configure and modify the extracts to best meet your business requirements.

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 Universal"](#)
- [Section 5.2.5, "Configuration Steps for Controlling Your Data Set"](#)

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."](#)

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 domainValues_Xact_Types_DocTypes_ora12.csv for Oracle General Ledger and Profitability Analytics"](#)
- [Section 5.2.2.3, "How to Configure domainValues_Xact_Types_DocTypes_ora11i.csv for Oracle General Ledger and Profitability Analytics"](#)

- [Section 5.2.2.4, "How to Specify the Ledger or Set of Books for which GL Data is Extracted"](#)

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 5–2](#) lists the CSV worksheet files and the domain values for Financial Analytics and Oracle EBS in the \$pmsserver\LkpFiles folder.

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 domainValues_Xact_Types_DocTypes_ora12.csv for Oracle General Ledger and Profitability Analytics").	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 domainValues_Xact_Types_DocTypes_ora11i.csv for Oracle General Ledger and Profitability Analytics").	SDE_ORA_TransactionTypeDimension_GLCOGSDerive

5.2.2.2 How to Configure domainValues_Xact_Types_DocTypes_ora12.csv for Oracle General Ledger and Profitability Analytics

This section explains how to configure domainValues_Xact_Types_DocTypes_ora12.csv for Oracle General Ledger and Profitability Analytics, Release 12.

To configure domainValues_Xact_Types_DocTypes_psft.csv 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 \$pmsserver\lkpfiles folder.

Look for only the rows in the file which has xact_cat_code = 'COGS' or xact_cat_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_ORAxxx_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. Make the same change in Informatica PowerCenter Workflow Manager to the SQL Override in the full load session SDE_GL_COGSFACT_Full.

5.2.2.3 How to Configure domainValues_Xact_Types_DocTypes_ora11i.csv for Oracle General Ledger and Profitability Analytics

This section explains how to configure domainValues_Xact_Types_DocTypes_ora11i.csv for Oracle General Ledger and Profitability Analytics, Release 11i.

To configure domainValues_Xact_Types_DocTypes_ora11i.csv for Oracle General Ledger and Profitability Analytics:

1. Identify the entry types in 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 \$pmsserver\lkpfiles folder.

Look for only the rows in the file which has `xact_cat_code = 'COGS'`. From among these records, look for those with `W_XACT_TYPE_CODE = 'DELIVERY'` or `'RETURN'`. The column `XACT_TYPE_CODE` will contain the entry type codes from 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. In Informatica PowerCenter Designer, open the mapping `SDE_ORA_GLCOGSFact` in the folder `SDE_ORAxxx_Adaptor`.
6. Open the maplet `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. Make the same change in Informatica PowerCenter Workflow Manager to the full load sessions that correspond to the mapping `SDE_ORA_GLCOGSFact_Full`.

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 for.

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 set of books to extract the GL data for, do the following:

1. In DAC, go to the Design view, and select the custom container that you want to configure.
2. Click the Source System Parameters tab.
3. Query for the parameter `$$FILTER_BY_SET_OF_BOOKS_TYPE`.
4. Double-click the Value column to open the text box, and enter 'Y'.
Make sure that you put single quotes around the Y.
5. Save the changes.
6. Query for the parameter `$$SET_OF_BOOKS_ID_LIST`.
7. Double-click the Value column to open the text box, and enter the IDs of the set of books for which you want to extract GL data for.

Specify the list of set of book IDs separated by commas (do not enter single or double quotes).

For example, if you want to extract GL data for set of books with IDs: 101, 207, and 303, then set the value of this parameter to the following:

101, 207, 303

8. Save the changes.

You can also specify the Set of Books type you want to extract GL data for. In Oracle 11i, there are three set of books types (this is based on the column `GL_SETS_OF_BOOKS.mrc_sob_type_code` column):

- P (Parent)
- R (Reporting)
- N (None)

For Oracle 11i customers, to configure the types of set of books to extract the GL data, do the following:

1. In DAC, go to the Design view, and select your custom container from the drop-down list.
2. Click the Source System Parameters tab.
3. Query for the parameter `$$FILTER_BY_SET_OF_BOOKS_TYPE`.
4. Double-click the Value column to open the text box, and enter the value 'Y'.
Make sure that you put single quotes around the Y.
5. Save the changes.
6. Query for the parameter `$$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 your custom container from the drop-down list.
2. Click the Source System Parameters tab.
3. Query for the parameter `$$FILTER_BY_LEDGER_ID`.
4. Double-click the Value column to open the text box, and enter the value 'Y'.
Make sure that you put single quotes around the Y.
5. Save the changes.
6. Query for the parameter `$$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 column):

- 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, in go to the Design view, and select your source container from the drop-down list.
2. Click 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 that you 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.

5.2.3 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, "GL Account Dimension, Chartfields and Hierarchy for PeopleSoft"](#)
- [Section 5.2.3.5, "Group Account Configuration"](#)

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 \$pmsserver\LkpFiles folder.

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 \$pmsserver\lkpfiles folder.

Look for only the rows in the file that have xact_cat_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_PSFTxxx_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. Make the same change in Informatica PowerCenter Workflow Manager 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, make sure 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, then click the Source System Parameters tab.
2. 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_EXTRACT_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 GL Account Dimension, Chartfields and Hierarchy 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 <OracleBI>\dwrep\Informatica\SrcFiles folder.

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 chartfields currently supported for the PeopleSoft application are listed below.

Note: Values are case sensitive. You must specify the values exactly as shown below.

- 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

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

Hierarchy

Oracle Business Intelligence Application 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. Information is provided in the sections that follow on how to extract these trees and how to setup the repository file (RPD) to use the Oracle Business Analytics Warehouse's hierarchy.

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 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 following:

Table 5–3 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.

5.2.3.5 Group Account Configuration

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.

Note: It is critical that the General Ledger Account Numbers are mapped to the Group Account Numbers (or domain values) as the metrics in the General Ledger reporting layer uses these values.

For example, Cash account, Payroll account, and so on. For a list of the Group Account Number domain values, see Oracle Business Analytics Warehouse Fusion Edition Data Model Reference. The group account num configuration is used during data extraction as well as front-end reporting. For example the group account num 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 <OracleBI>\dwrep\Informatica\SrcFiles folder.

Table 5–4 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
AUS01	171101	173001	OTHER
AUS01	240100	240120	REVENUE
AUS01	251100	251120	TAX

In the Table above, 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.

5.2.4 Configuration Steps for Financial Analytics for Universal

This section contains configuration steps before you do a full data load that apply to Universal.

Not applicable to Oracle BI Applications release 7.9.5.1.

5.2.5 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle Financial Analytics. It contains the following topics:

- [Section 5.2.5.1, "Configuration Steps for Financial Analytics for All Source Systems"](#)
- [Section 5.2.5.2, "Configuration Steps for Financial Analytics for Oracle EBS"](#)
- [Section 5.2.5.3, "Configuration Steps for Financial Analytics for PeopleSoft"](#)
- [Section 5.2.5.4, "Configuration Steps for Financial Analytics Chain for Universal"](#)

5.2.5.1 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 5.2.5.1.1, "How to Include UnApplied Payments in AR Aging Calculations"](#)
- [Section 5.2.5.1.2, "How to Configure the AP/AR Aging Tables"](#)

5.2.5.1.1 How to Include UnApplied Payments in AR Aging Calculations 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, 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
```

4. Save the changes.

The next ETL will populate the Aging tables using UnApplied payments in the calculations.

5.2.5.1.2 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 below.

To configure the length of aging buckets

1. In the Design view, click the Tasks tab.
2. Query for the following tasks
 - PLP_APSnapshotInvoiceAging
 - PLP_ARSnapshotInvoiceAging
3. For each of these tasks, click on the Parameters tab and update the parameters with names starting with '\$\$BUCKET'.
4. Save your work.

These aging tables are snapshot tables. You can also control how many historic month end snapshots you want to keep in these tables (the default value is 24 months). You can increase or decrease the number by adjusting the \$\$HISTORY_MONTHS parameter.

5.2.5.2 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 5.2.5.2.1, "How GL Balances Are Populated in Oracle EBS"](#)
- [Section 5.2.5.2.2, "How to Configure Oracle Profitability Analytics Transaction Extracts"](#)
- [Section 5.2.5.2.3, "How to Configure Cost Of Goods Extract \(for Oracle EBS 11i\)"](#)
- [Section 5.2.5.2.4, "How to Configure AP Balance ID for Oracle Payables Analytics"](#)
- [Section 5.2.5.2.5, "How to Configure AR Balance ID for Oracle Receivables Analytics and Oracle General Ledger and Profitability Analytics"](#)
- [Section 5.2.5.2.6, "How to Configure the AR Adjustments Extract for Oracle Receivables Analytics"](#)
- [Section 5.2.5.2.7, "How to Configure the AR Schedules Extract"](#)
- [Section 5.2.5.2.8, "How to Configure the AR Cash Receipt Application Extract for Oracle Receivables Analytics"](#)
- [Section 5.2.5.2.9, "How to Configure the AR Credit-Memo Application Extract for Oracle Receivables Analytics"](#)
- [Section 5.2.5.2.10, "How to Set Up Drill Down in Oracle BI Answers from General Ledger to Subledger"](#)

5.2.5.2.1 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 the DAC Design view, click the Subject Areas tab, and query for the subject area 'Financials – General Ledger'. Click the Configuration Tags subtab, and query for the tag 'Oracle – Extract GL Balance'. Activate the tag by deselecting the Inactive check box.
 - b. Inactivate the tag 'Financials – Calculate GL Balance' by selecting the is Inactive check box.
 - c. Click Assemble to reassemble the subject area.
 - d. 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 the DAC Design view, click the Subject Areas tab, and query for the subject area 'Financials – General Ledger'. Click the Configuration Tags subtab, and query for the tag 'Financials – Calculate GL Balance'. Activate the tag by deselecting the Inactive check box.
 - b. Inactivate the tag 'Oracle – Extract GL Balance' by selecting the is Inactive check box.
 - c. Click Assemble to reassemble the subject area.
 - d. Rebuild the execution plans that contain this subject area.

5.2.5.2.2 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 General Ledger and Profitability Analytics Revenue

1. In Informatica PowerCenter Designer, open the appropriate Oracle Applications folder (for example, `SDE_ORA11510_Adaptor`).
2. In Mapplet Designer, open the `mp1t_BC_ORA_GLRevenueFact` mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations dialog, and click 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.

5.2.5.2.3 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_ORA11510_Adaptor`).
2. In Mapplet Designer, open `mp1t_BC_ORA_GLCOGSFact`.
3. Double-click the Source Qualifier to open the Edit Transformations dialog, and click 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 `mp1t_BC_ORA_GLCOGSFact_Primary`.
5. Double-click the Source Qualifier to open the Edit Transformations dialog, and click 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.
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.

5.2.5.2.4 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 || '~' || INP_ORG_ID
```

However, if you want to maintain your AP balance at a different grain, you can redefine the Balance ID value in the applicable 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
 - mplt_SA_ORA_APTransactionFact_PaymentSchedule
 - mplt_SA_ORA_APTransactionFact_DiffMan
-
-

1. In Informatica PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA11510_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. Click the Ports tab to edit the Balance ID definition in the EXT_NU_BALANCE_ID column.
5. Validate and save your changes to the repository.
6. Repeat steps 1 to 5 for each mapplet that is listed above.

5.2.5.2.5 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_ORA11510_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. Click the Ports tab to edit the Balance ID definition in the `EXT_NU_AR_BALANCE_ID` column.
5. Validate and save your changes to the repository.
6. Repeat steps 1 to 5 for each mapplet that is listed above.

5.2.5.2.6 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_ORA11510_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 click 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.

5.2.5.2.7 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_ORA11510_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.

5.2.5.2.8 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_ORA11510_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 click 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.

5.2.5.2.9 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 receipts

1. In Informatica PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE_ORA11510_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 click 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.

5.2.5.2.10 How to Set Up Drill Down in Oracle BI Answers from General Ledger to Subledger

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. This feature is available for AP if the source is Oracle EBS 11i and for all subledgers if the source is Oracle EBS R12.

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 subledger request you created earlier.

Note: For COGS and Revenue, the 'GL Journal ID' column is not exposed in any presentation catalogs. It is available in the business model layer of the RPD metadata under the logical tables 'Dim - GL COGS Details' and 'Dim - GL Revenue Details' respectively. As a workaround, you can create presentation catalogs to report of detail level transactions for COGS and Revenue and expose this column in the presentation catalog and use similar steps as above to setup a drilldown from GL to COGS/Revenue.

5.2.5.3 Configuration Steps for Financial Analytics for PeopleSoft

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

- [Section 5.2.5.3.1, "Overview of PeopleSoft Tree Data Structures"](#)
- [Section 5.2.5.3.2, "About PeopleSoft Trees in Oracle Financial Analytics"](#)
- [Section 5.2.5.3.3, "General Ledger Chartfield Hierarchy Extraction Customization"](#)
- [Section 5.2.5.3.4, "Internal Organization Hierarchy Extraction Customization"](#)

5.2.5.3.1 Overview of PeopleSoft Tree Data Structures For PeopleSoft, the Oracle Financial Analytics application sources 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.

5.2.5.3.2 About PeopleSoft Trees in Oracle Financial Analytics Oracle BI Applications provides a set of mechanisms to support various tree structures in PeopleSoft applications. [Table 5-5](#) shows the supported structure types, and [Table 5-6](#) shows the supported level types.

Table 5-5 PeopleSoft Tree Structure Types

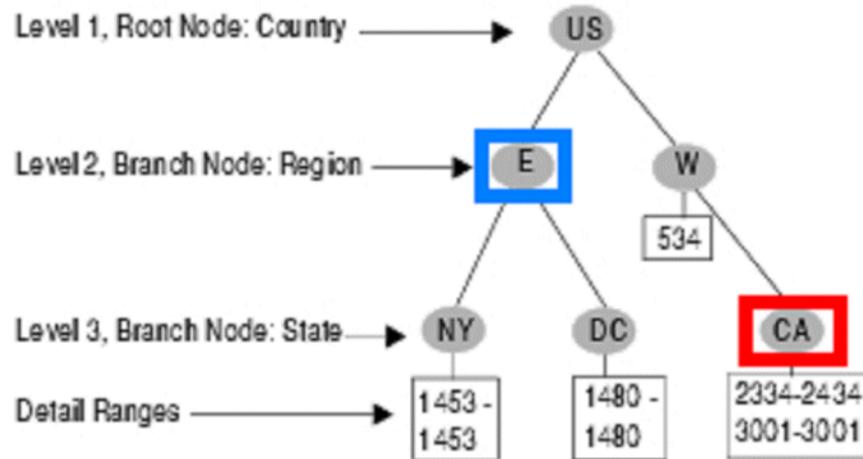
Structure Type	Support
Winter tree	Y
Detail tree	Y
Summary tree	Y

Table 5-6 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



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 5-7 will appear in the database.

Table 5-7 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 5-8.

Table 5-8 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 summary 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 5–9](#).

Table 5–9 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 5–10](#) lists the tables used during the GL Hierarchy extraction process. The sequence indicates the order in which these tables are populated.

Table 5–10 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 5–11](#) lists the tables used during the Internal Organization Hierarchy extraction process.

Table 5–11 Internal Organization Hierarchy

Sequence	Table Name
1	W_PSFT_INT_ORG_TREE_TMP
2	W_PSFT_INT_ORG_GLB_U_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.

Using DAC Configuration Tags to Activate and Inactivate Tasks

The DAC's Configuration Tag feature allows 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 the DAC Design view, select the PeopleSoft custom container from the drop-down list to the right of the Execute button.
2. Click 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, click 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 click the Execute tab.
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 the following:

- [Section 5.2.5.3.3, "General Ledger Chartfield Hierarchy Extraction Customization"](#)
- [Section 5.2.5.3.4, "Internal Organization Hierarchy Extraction Customization"](#)

5.2.5.3.3 General Ledger Chartfield Hierarchy Extraction Customization This section provides an example of how to customize mappings for a GL Chartfield 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 5–12](#).

Table 5–12 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 steps required to perform this extraction are as follows:

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, and the SetID to XYZ.
 - b. Open the Exp_TreeName_SetID Expression transformation, and change the values as follows:
 - Change the TREE_NAME column to DEPARTMENT_HIERARCHY.
 - Change the SETID column to XYZ.
 - Change the CHARTFIELD column 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_TABLE_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. Change the source table and Source Qualifier to DEPT_TBL.
 - 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_xx_Adaptor).
 - b. Open the Task Developer and create a new task for the mappings you just created. Modify the newly created sessions by referring to the parameters in the original sample sessions. (For an example, see [Table 5-13](#), which shows parameter configurations for SDE_PSFT_Stage_GLHierarchy_DepartmentHierarchy_Extract session; leave the other mappings set to the default value).

Table 5-13 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

Table 5–13 (Cont.) General Ledger Chartfield Hierarchy Mappings

Tab Name	Parameter Name	Value
Mapping	Connection SQ_PSFT_TREE_VERTICAL_FLATTENING - Application	\$DBCConnection_OLTP
Mapping	Connection W_PSFT_TREE_TMP1 - Rational	DBCConnection_OLAP
Mapping	Connection Lkp_Tree_ChartField - Rational	\$DBCConnection_OLTP
Mapping	SDE_PSFT_Stage_GLHierarchy_DepartmentHierarchy_Extract – Session \$Source Connection Value	\$DBCConnection_OLTP
Mapping	SDE_PSFT_Stage_GLHierarchy_DepartmentHierarchy_Extract – Session \$Target Connection Value	\$DBCConnection_OLAP

Note: The SDE_PSFT_Stage_GLHierarchy_GetDepartmentChartField parameters are similar to those shown in [Table 5–13](#).

- c. Save your changes.
 - d. In Informatica PowerCenter Workflow Manager, create a workflow with the same name as the session.
 - e. Drag the newly created session to the panel.
 - f. Connect the start with the session.
 - g. 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 click 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 on 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 click 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 the DAC's Design view, select the Task Group tab.
- In the top pane, select the appropriate task group.
- Click 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.
- Click 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 toolbar.
- 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 click the Execution Plans tab.

- Select the appropriate execution plan (for example, Financials_PeopleSoft 8.8).
- Click Build in the toolbar.

5.2.5.3.4 Internal Organization Hierarchy Extraction Customization This section provides an example of how to customize mappings for an Internal Organization Hierarchy extraction. Internal Organization Hierarchy only extracts the hierarchy related to the business unit. In this example, the extraction involves a tree named BIZ_UNIT_HIER with SetID XYZ.

The mappings that require customization for this example are shown in [Table 5–14](#).

Table 5–14 Internal Organization Hierarchy Mappings

Execution Sequence	Mapping Name	Needs Customization?
1	SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_Extract	Y
2	SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_GetGLBU	N
3	SDE_PSFT_InternalOrganizationDimension_CompanyHierarchy_GetHierarchyLevels	N
4	SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_DeriveRange	N
5	SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_Flatten	N
6	SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_Derive	N

The steps required to perform this extraction are as follows:

1. In Informatica PowerCenter Designer, create a new mapping by copying the existing mapping SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_Extract and renaming it as SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy2_Extract.
2. Open the mapping SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy2_Extract and edit the settings as follows:
 - a. Open the source definition PSFT_TREE_VERTICAL_FLATTENING. Under the 'Properties' tab, change the Tree Name to BIZ_UNIT_HIER, and the SetID to XYZ.
 - b. Open the Exp_PSFT_HIERARCHY Expression transformation, and change the values as follows:
 - Change the TREE_NAME column to BIZ_UNIT_HIER.
 - Change the SETID column to XYZ.
3. Save all your changes in the Informatica folder.
4. 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_xx_Adaptor).

- b. Open the Task Developer and create a new task for the mappings you just created. Modify the newly created sessions by referring to the parameters in the original sample sessions. (For an example, see [Table 5-15](#), which shows parameter configurations for SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy2_Extract session; leave the other mappings set to the default value).

Table 5-15 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_INT_ORG_TREE_TMP - Rational	\$DBConnection_OLAP
Mapping	Connection SQ_PSFT_TREE_VERTICAL_FLATTENING - Application	\$DBConnection_OLTP
Mapping	Connection Lkp_Tree_ChartField - Rational	\$DBConnection_OLTP
Mapping	SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy2_Extract – Session \$Source Connection Value	\$DBConnection_OLTP
Mapping	SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy2_Extract – Session \$Target Connection Value	\$DBConnection_OLAP

- c. Save your changes.
 - d. In Informatica PowerCenter Workflow Manager, create a workflow with the same name as the session.
 - e. Drag the newly created session to the panel.
 - f. Connect the start with the session.
5. Save and check-in all your changes in Informatica PowerCenter Designer and Informatica PowerCenter Workflow Manager.
 6. Reconfigure the DAC settings, as follows:
 - a. Log in to DAC, open your custom container, and click the Tasks tab.
 - b. Add the workflow that you just created as a new task.
 - SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy2_Extract

For information about what values to set task property values to, refer to the task:

- SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_Extract

- c. Right-click on the 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 click the Source Tables subtab and the Target Tables subtab.

- d. Add the newly created task 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. Internal Organization Hierarchy Extract has the following task groups:

- TASK_GROUP_Extract_InternalOrganizationDimensionHierarchy

- TASK_GROUP_Extract_InternalOrganizationDimensionHierarchy_Derive

For this example, you need to include the newly added task into the task group TASK_GROUP_Extract_InternalOrganizationDimensionHierarchy, as follows:

- In the DAC's Design view, select the Task Group tab.

- In the top pane, select the task group TASK_GROUP_Extract_InternalOrganizationDimensionHierarchy.

- Click the Child Tasks subtab, and then click Add/Remove in the bottom pane toolbar.

- Query for the task SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy2_Extract, and then click Add.

- Click OK, and then click Save.

- e. Add the task SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy2_Extract as a configuration tag in DAC:

- In the Design view, click Configuration Tags.

- Query for the tag PeopleSoft - Populate Business Unit Related Hierarchies.

- Click 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 toolbar.

- 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 click the Execution Plans tab.

- Select the appropriate execution plan (for example, Financials_PeopleSoft 8.8).

- Click Build in the toolbar.

5.2.5.4 Configuration Steps for Financial Analytics Chain for Universal

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

- [Section 5.2.5.4.1, "About the Customer Costs Lines and Product Costs Lines Tables for Oracle's Profitability Analytics"](#)
- [Section 5.2.5.4.2, "How to Configure the Customer Costs Lines and Product Costs Lines Tables for Oracle's Profitability Analytics"](#)
- [Section 5.2.5.4.3, "How to Setup Drill Down in Oracle BI Answers from General Ledger to Subledger"](#)

5.2.5.4.1 About the Customer Costs Lines and Product Costs Lines Tables for Oracle's 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. By default, for the Universal adapter, Oracle BI Applications does not capture the miscellaneous cost and expenses associated with generating revenue from a customer or from a product (for example, expenses such as marketing campaign expenses).

The Customer Costs Lines (W_CUSTOMER_COST_LINE_F) table stores cost details by customers. The total cost by customers 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 table also stores the source of allocations. For example, the Sales and Marketing costs are not be 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. This table stores the source cost centers and General Ledger accounts as well. The Customer Costs can be categorized by Sales, Marketing, Service, and Operating costs. These could be further broken down to Salaries, Events, and Promotions. 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. Apart from 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. The different cost lines such as Sales, Marketing, and Operating costs, have different details and many dimensions are included in this table. Some of the dimension keys are not be applicable to certain cost components. It is important that an organization identifies the dimensional keys that are used for Product Cost Analysis for various components.

The Product Costs Lines (W_PRODUCT_COST_LINE_F) table stores 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 table also stores 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. This table also stores 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 are also tracked in this table. Apart from 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. The different cost lines such as Sales, Marketing, and Operating costs, have different details and many dimensions are included in this table. Some of the dimension keys are not applicable to certain cost components. It is important that an organization identifies the dimensional keys that is used for Product Cost Analysis for various components.

5.2.5.4.2 How to Configure the Customer Costs Lines and Product Costs Lines Tables for Oracle's Profitability Analytics In Oracle's Profitability Analytics, the Customer Costs Lines and Product Costs Lines fact tables store the costing and expenses for the Profitability functional area. You need to use these tables with General Ledger Revenue and General Ledger COGS fact tables.

The General Ledger Revenue and General Ledger COGS fact tables are populated by the or Oracle 11i source system, but the Customer Costs Lines and Product Costs Lines fact tables are populated by the universal source system.

To load the Customer Costs Lines and Product Costs Lines tables

1. Open the customer_cost_line.csv file in the \$pmsserver\Sample Universal Source Files folder.
2. Insert a record into the file for each customer costing transaction you want to load into the Customer Cost fact table.

For the DATASOURCE_NUM_ID column, you need to provide the same source identifier value as in the file_parameters_ora11i.csv file.
3. Save the file in the \$pmsserver\srcfiles directory, then close the file.
4. Repeat steps 1 to 3 for the file_product_cost.csv file.

5.2.5.4.3 How to Setup Drill Down in Oracle BI Answers from General Ledger to Subledger

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 subledger request you created earlier.

5.2.5.4.4 How to Configure Budgets for Oracle Profitability Analytics If you are implementing Oracle Profitability Analytics and would like to view the budget data in your dashboards, you need to import budget data into your data warehouse, as described in this section.

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.

The table below shows the structure of the file_budget.csv file. The records in the file_acct_budget.csv will be loaded into W_BUDGET_D.

Table 5–16 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.
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	

The table below 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 5–17 Universal Source for Budget Fact (file_acct_budget.csv)

Column Name	Datatype	Size	Description
GL_ACCOUNT_ID	string	80	GL Account identifier. Populate with integration_id from w_gl_account_d.
PROJECT_ID	string	80	
PRODUCT_ID	string	80	Product identifier. Populate with integration_id from w_product_d.
COMPANY_ORG_ID	string	80	Company Org identifier. Populate with integration_id from w_int_org_d where company_flg = Y.
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_FIN_AREA_ORG_ID	string	80	Company Org identifier. Populate with integration_id from w_int_org_d where fin_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.
COST_CENTER_ID	string	80	Cost Center identifier. Populate with integration_id from w_cost_center_d.
PROFIT_CENTER_ID	string	80	Profit Center identifier. Populate with integration_id from w_profit_center_d.
BUDGET_ID	string	80	Populate with the value from integration_id in file_budget.csv
POSTED_ON_DT	string	14	A date on which this transaction can be reported.
PERIOD_BEGIN_DT	string	14	
PERIOD_END_DT	string	14	Populate with your budget period's end date. If your budget is monthly, populate with the month end date.
BUDGET_DOC_AMT	number	22	Budget amount in document currency.
BUDGET_LOC_AMT	number	22	Budget amount in local currency.
BUDGET_GRP_AMT	number	22	
DOC_CURR_CODE	string	30	Document currency code.
LOC_CURR_CODE	string	30	Local currency code.
GRP_CURR_CODE	string	30	
CREATED_BY_ID	string	80	ID of created by user. Populate with Integration_ID from w_user_d.
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.

Table 5–17 (Cont.) Universal Source for Budget Fact (file_acct_budget.csv)

Column Name	Datatype	Size	Description
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.
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	

Note: date columns are populated in the csv file as a number in YYYYMMDDHH24MISS format.

Use the following table 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 5–18 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.

Table 5–18 (Cont.) Populating the integration_id fields in Oracle EBS Source Systems

Field	How to populate
LEDGER_ID(w_ledger_d)	For Oracle 11i populate as set of book ID For Oracle R12 populate as ledger ID.

How to import budget data into your data warehouse

1. Populate the file_budget.csv and file_acct_budget.csv files with your budget data.
 The CSV files are located in the OracleBI\dwrep\Informatica\Sample Universal Source Files\ directory of the Oracle BI Applications installation machine.
 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.

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"](#)

6.1 Overview of Oracle Supply Chain and Order Management Analytics

The Oracle Supply Chain and Order Management Analytics application allows you to analyze:

- Bill of materials.
- Bookings.
- Financial and Operational Backlogs.
- Inventory held by an organization.
- Inventory movements in, out, and through manufacturing plants, distribution centers, or storage locations.
- Invoices.
- The movement of sales orders through different stages of the sales cycle.

The Oracle Supply Chain and Order Management Analytics application consists of orders, invoices, backlogs and inventory. Sales orders are the entry point for the sales process. Invoices are the exit point from the fulfillment process. Backlogs are points of congestion in your fulfillment process. This coverage includes insight into which items are booked, backlogged, and invoiced. This allows you to evaluate the sales performance of individual sales representatives or departments. Oracle Supply Chain and Order Management Analytics 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 following sources can populate Orders, Inventory and Revenue:

- Oracle 11i
- Oracle R12

- Universal source

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 PeopleSoft"](#)
- [Section 6.2.4, "Configuration Steps for Oracle Supply Chain and Order Management Analytics for Universal"](#)
- [Section 6.2.5, "Configuration Steps for Controlling Your Data Set"](#)

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.

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"](#)).

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 6.2.2.6, "How to Configure Invoice Status Domain Values"](#)
- [Section 6.2.2.7, "How to Configure Order Overall Status Domain Values"](#)
- [Section 6.2.2.8, "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 Quantity Types for Product Transactions"](#)

6.2.2.1 About Configuring Domain Values and CSV Worksheet Files for Oracle Supply Chain and Order Management Analytics

Table 6–1 below lists the CSV worksheet files and the domain values for Oracle Supply Chain and Order Management Analytics that are located in the \$pmsserver\LkpFiles folder.

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_ora11i.csv	Lists the method code column and the corresponding domain value for the application.	SDE_ORA_PaymentMethodDimension

Table 6–1 (Cont.) Domain Values and CSV Worksheet Files for Oracle Supply Chain and Order Management Analytics

Worksheet File Name	Description	Session
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 6.2.2.6, "How to Configure Invoice Status Domain Values" .	SDE_ORA_StatusDimension_SalesInvoiceLine
DomainValue_OrderOverallStatus_ora11i.csv	List the Order Status Code column and the corresponding domain values for the Oracle 11i or Oracle R12 application. For more information about updating the values in this file, see Section 6.2.2.7, "How to Configure Order Overall Status Domain Values" .	SDE_ORA_StatusDimension_SalesOrderLineCycle

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*.

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_InvoiceType_ora11i.csv file, located in the \$pmsserver\lkpfiles folder.
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. Identify the Pick Types in your Oracle 11i source system.
2. Using a text editor open the domainValues_PickType_ora11i.csv file, located in the \$pmserver\lkpfiles folder.
3. 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.

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.

6.2.2.4 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_OrderType_ora11i.csv file, located in \$pmserver\lkpfiles folder.
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 \$pmsserver\lkpfiles folder.
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 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 \$pmsserver\lkpfiles folder.
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.

6.2.2.7 How to Configure Order Overall Status Domain Values

This section explains how to configure Order Overall Status domain values using the domainValues_OrderOverallStatus_ora11i.csv file.

To configure Order Overall Status domain values

1. Identify the Order Overall Statuses in your Oracle 11i source system by using the following SQL:

```
SELECT DISTINCT FND_LOOKUP_VALUES.LOOKUP_CODE
FROM FND_LOOKUP_VALUES
WHERE FND_LOOKUP_VALUES.LOOKUP_TYPE = 'LINE_FLOW_STATUS'
      AND FND_LOOKUP_VALUES.LANGUAGE = 'US'
      AND FND_LOOKUP_VALUES.VIEW_APPLICATION_ID = 660
      AND FND_LOOKUP_VALUES.SECURITY_GROUP_ID = 0
ORDER BY 1;
```

2. Using a text editor, open the domainValues_OrderOverallStatus_ora11i.csv file, located in the \$pmserver\lkpfiles folder.
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 Pay Method Domain Values

This section explains how to configure Pay Method Status domain values using the domainValues_PayMethodCode_ora11i.csv file.

To configure Pay Method domain values

1. Identify the Pay Methods in your Oracle 11i source system by using the following SQL:

```
SELECT DISTINCT FND_LOOKUP_VALUES.LOOKUP_CODE
FROM FND_LOOKUP_VALUES
WHERE LOOKUP_TYPE = 'PAYMENT TYPE'
      AND VIEW_APPLICATION_ID = 660
      AND LANGUAGE = 'US'
      AND FND_LOOKUP_VALUES.SECURITY_GROUP_ID = 0
ORDER BY 1;
```

2. Using a text editor, open the domainValues_PayMethodCode_ora11i.csv file, located in the \$pmserver\lkpfiles folder.
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.

6.2.2.9 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 \$pmsserver\lkpfiles folder, 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 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, please 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_TYPE_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<Ver>_Adaptor.
2. Open the mplt_SA_ORA_ProductTransactionFact maplet.
3. Double-click the Expression transformation to open the Edit Transformations dialog, and click the Port tab to display the EXT_GR_QTY and EXT_DELIVERY_QTY port.
4. Edit the quantity types by substituting your desired condition for the prepackaged expression.
5. Click Apply.
6. Validate the maplet, and save your changes to the repository.

6.2.3 Configuration Steps for Oracle Supply Chain and Order Management Analytics for PeopleSoft

This section contains configuration steps required before you do a full data load that apply to PeopleSoft.

Not applicable to Oracle BI Applications release 7.9.5.1.

6.2.4 Configuration Steps for Oracle Supply Chain and Order Management Analytics for Universal

This section contains configuration steps required before you do a full data load that apply to Universal.

Not applicable to Oracle BI Applications release 7.9.5.1.

6.2.5 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 6.2.5.1, "Configuration Steps for Oracle Supply Chain and Order Management Analytics for All Source Systems"](#)
- [Section 6.2.5.2, "Configuration Steps for Oracle Supply Chain and Order Management Analytics for Oracle EBS"](#)
- [Section 6.2.5.3, "Configuration Steps for Oracle Supply Chain and Order Management Analytics for PeopleSoft"](#)
- [Section 6.2.5.4, "Configuration Steps for Oracle Supply Chain and Order Management Analytics for Universal"](#)

6.2.5.1 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 6.2.5.1.1, "Tracking Multiple Attribute Changes in Bookings"](#)
- [Section 6.2.5.1.2, "Process of Aggregating Oracle Supply Chain and Order Management Analytics Tables"](#)
- [Section 6.2.5.1.3, "About Tracking Multiple Products for Oracle Supply Chain and Order Management Analytics"](#)

- [Section 6.2.5.1.4, "Adding Dates to the Order Cycle Time Table for Post-Load Processing"](#)
- [Section 6.2.5.1.5, "About Configuring the Backlog Period Date for Oracle Supply Chain and Order Management Analytics"](#)
- [Section 6.2.5.1.6, "Example of How Backlog Data Is Stored in the Backlog History Table"](#)
- [Section 6.2.5.1.7, "About Configuring the Customer Status History Fact for Post-Load Processing In Oracle Supply Chain and Order Management Analytics"](#)
- [Section 6.2.5.1.8, "Configuring the Customer Status History Fact table"](#)
- [Section 6.2.5.1.9, "How to Configure the Customer Status History Fact table"](#)
- [Section 6.2.5.1.10, "About Configuring the Inventory Monthly Balance Table"](#)
- [Section 6.2.5.1.11, "How to Configure the Inventory Monthly Balance"](#)
- [Section 6.2.5.1.12, "About Configuring the Product Transaction Aggregate Table"](#)
- [Section 6.2.5.1.13, "How to Configure the Product Transaction Aggregate Table"](#)

6.2.5.1.1 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, then concatenate the technical name IDs in the VAR_BOOKING_ID column as follows:

```
TO_CHAR (INP_LINE_ID) || '~' || TO_CHAR (INP_INV_ITEM_ID) || '~' || TO_CHAR (INP_WAREHOUSE_ID) || '~' || TO_CHAR (INP_SALESREP_ID) || '~' || TO_CHAR (INP_CUSTOMER_ID)
```

To track dimensional attribute changes in bookings

1. In Informatica PowerCenter Designer, open the SDE_ORA115<ver>_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 VAR_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.

6.2.5.1.2 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:

- 'DAY'
- '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.

To configure the Sales Invoice Lines Aggregate Table

1. In DAC, go to the Design view, and select the appropriate custom container.
2. Click the Tasks tab.
3. For each of the following tasks, click the Parameters tab and specify an appropriate value in the Value field:
 - `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:

- `'DAY'`
- `'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 your custom container from the drop-down list.
2. Click the Tasks tab.
3. For each of the following tasks, click the Parameters tab and specify an appropriate value in the Value field:

- SIL_SalesOrderLinesAggregate_Derive_PreLoadImage
- SIL_SalesOrderLinesAggregate_Derive_PreSoftDeleteImage

6.2.5.1.3 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 6–2](#). The ORDLN_KEY_ID field contains the Line Item ID of the parent product in order to maintain the relationship between the parent and child products in a package. In this example, the ORDLN_KEY_ID field is Line_1 for each of the three child products (A1, A2, A3) that were sold as a part of the parent package, Parent A.

Table 6–2 Sales Order Table Columns With Parent/Child Relationships

Key_ID	SALES_ORDER_NUM	PRODUCT_ID	ORDHD_KEY_ID	ORDLN_KEY_ID	Relationship (Not a column in the table.)
Line_1	1000	Package	1000	Line_1	Parent A
Line_2	1000	Computer	1000	Line_1	Child A1
Line_3	1000	Scanner	1000	Line_1	Child A2
Line_4	1000	Printer	1000	Line_1	Child A3
Line_5	1000	Monitor	1000	Line_5	Parent B (no children)

In contrast, if each of the four items described in [Table 6–2](#) were bought individually, the ORDLN_KEY_ID would have a different Line Item ID for every row. In this case, the Sales Order Lines table would look like [Table 6–3](#).

Table 6–3 Sales Order Table Columns Without Parent/Child Relationships

Key_ID	SALES_ORDER_NUM	PRODUCT_ID	ORDHD_KEY_ID	ORDLN_KEY_ID	Relationship (Not a column in the table.)
Line_1	1000	Computer	1000	Line_1	None
Line_2	1000	Scanner	1000	Line_2	None
Line_3	1000	Printer	1000	Line_3	None
Line_4	1000	Monitor	1000	Line_4	None

6.2.5.1.4 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.
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. 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.

6.2.5.1.5 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_HISTORY_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.

6.2.5.1.6 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 6-4.

Table 6-4 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 6-5](#).

Table 6-5 Oracle 11i and Oracle R12: Backlog History Table Entry as of February 2, 2001

SALES_ORDER_NUM (Sales Order Number)	BACKLOG_DK (Backlog Date)	BACKLOG_PERIOD_DK (Backlog Period Date)	OPEN_QTY (Backlog Quantity)
1	02/01/2001	02/28/2001	Old value: 10 New value: 5

No further activity happens until March 1st. On March 1st, the remaining 5 items on financial backlog are invoiced and removed from financial backlog. In addition, a new sales order (Sales Order #2) comes in for 50 new items. All of the items are put on financial backlog.

Even though all items from Sales Order #1 are cleared from financial backlog, the last backlog row remains in the Backlog History table. The purpose in retaining the last row is to indicate that there was backlog for this particular order. The quantity, in this case 5 items, does not tell you how many items were initially on backlog, which was 10.

For the 50 new financial backlog items, there is a new entry into the Backlog History table. So, as of February 28, 2001, the Backlog History table looks like the [Table 6-6](#).

Table 6-6 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 6-7](#).

Table 6-7 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 6-7](#), 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 6–8](#) provides a list of all applicable mappings and their corresponding Expression transformations that you must modify.

Table 6–8 Oracle 11i and Oracle R12: Backlog History Applicable Mappings and Expression Transformations

Mapping	Expression Transformation
PLP_SalesBacklogLinesfact_ LoadOrderLines	EXP_SALES_ORNLNS_BACKLOG
PLP_SalesBacklogLinesfact_ LoadScheduleLines	EXP_SALES_SCHLNS_BACKLOG

The backlog history period is monthly by default. The default SQL statement in the Expression transformation for the port BACKLOG_PERIOD_DK is:

```
TO_DECIMAL (TO_CHAR (LAST_DAY (CALENDAR_DATE) , 'YYYYMMDD' ) )
```

You can edit the backlog period date so that you can capture a more detailed backlog history with the following procedure. Possible periods include daily (CAL_DAY_DT), weekly (CAL_WEEK_DT), monthly (CAL_MONTH_DT), and quarterly (CAL_QTR_DT).

6.2.5.1.7 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.

6.2.5.1.8 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 6.2.5.1.9, "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 targeting 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 6.2.5.1.9, "How to Configure the Customer Status History Fact table"](#).

6.2.5.1.9 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 6.2.5.1.8, "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.
2. Click 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.

2. Click the Tasks tab and query for the following two tasks:
 - PLP_CustomerStatusHistoryFact_New_Customers_Load
 - PLP_CustomerStatusHistoryFact_Status_Revalidate
3. For each task, click 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.

6.2.5.1.10 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
- KEEP_PERIOD
- NUM_OF_PERIOD

The GRAIN parameter has a preconfigured value of Month. The possible values for the GRAIN parameter are:

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

The KEEP_PERIOD parameter has a preconfigured value of Month. Values for the KEEP_PERIOD parameter include:

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

The NUM_OF_PERIOD parameter has a preconfigured value of 3. The value for the NUM_OF_PERIOD parameter is a positive integer, for example, 1, 2, 3, and so on.

6.2.5.1.11 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. Click the Tasks tab.

3. For each of the following tasks, click 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, please make sure that you adjust the NUM_OF_PERIOD values to reflect your data volume and data recency requirement.

6.2.5.1.12 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).

6.2.5.1.13 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. Click the Tasks tab.
3. For each of the following tasks, click the Parameters subtab and create the specified parameter name and parameter value
 - PLP_ProductTransactionAggregate \$\$REFRESH_PERIOD 'MONTH'
 - PLP_ProductTransactionAggregate \$\$GRAIN 'MONTH'
 - PLP_ProductTransactionAggregate \$\$NUM_OF_PERIOD 3
 - PLP_ProductTransactionAggregate_Full \$\$GRAIN 'MONTH'

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_ProductTransactionAggregate_Full` 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.

6.2.5.2 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:

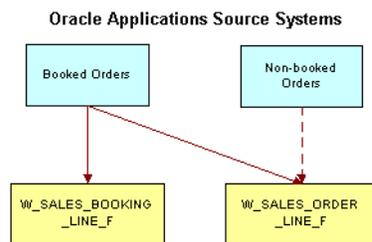
- [Section 6.2.5.2.1, "How to Configure Sales Order Lines Data Storage"](#)
- [Section 6.2.5.2.3, "How to Configure Sales Schedule Lines Data Storage"](#)
- [Section 6.2.5.2.3, "How to Configure Sales Schedule Lines Data Storage"](#)
- [Section 6.2.5.2.4, "About the Handling of Booked and Nonbooked Orders in the Sales Schedule Lines Table"](#)
- [Section 6.2.5.2.5, "About Loading Bookings at the Schedule Line Level"](#)
- [Section 6.2.5.2.6, "How to Configure Early and Late Tolerances for Shipments"](#)
- [Section 6.2.5.2.7, "How to Configure Sales Invoice Lines Data Storage"](#)
- [Section 6.2.5.2.8, "How to Configure the Sales Invoice Extract"](#)
- [Section 6.2.5.2.9, "How to Configure the Order Lines Execution Plan and OTB ETL"](#)
- [Section 6.2.5.2.10, "How to Configure Procurement and Spend Analytics for Oracle Supply Chain and Order Management Analytics"](#)
- [Section 6.2.5.2.11, "How to Configure Oracle Financial Analytics for Oracle Supply Chain and Order Management Analytics"](#)
- [Section 6.2.5.2.12, "About Tracking Attribute Changes in Bookings"](#)
- [Section 6.2.5.2.13, "About Viewing the Data Warehouse Changes by Salesperson ID"](#)
- [Section 6.2.5.2.14, "How to Configure Different Types of Backlog Calculations"](#)
- [Section 6.2.5.2.15, "Adding Closed Orders to Backlog Calculations"](#)
- [Section 6.2.5.2.16, "How to Configure the Bill of Materials Explosion Options"](#)
- [Section 6.2.5.2.17, "How to Configure the Left Bound and Right Bound Calculation Option"](#)

6.2.5.2.1 How to Configure Sales Order Lines Data Storage Sales order lines are the itemized lines that make up a sales order. This information is stored in the `W_SALES_ORDER_LINE_F` table. This topic describes how to modify the type of information stored in this table.

6.2.5.2.2 About the Handling of Booked and Nonbooked Orders in the Order Lines and Bookings Table By default, only booked orders are extracted from the Oracle source system as shown in [Figure 6-1](#). Therefore, all orders loaded into the Sales Order Lines and Bookings tables are booked.

However, if you want to load nonbooked orders into the Sales Order Lines table, you have to configure the extract so that it does not filter out nonbooked orders. In Oracle 11i and Oracle R12, the `OE_ORDER_LINES_ALL.BOOKED_FLAG = Y` condition indicates that an order is booked; therefore, this statement is used to filter out nonbooked orders. To load all orders, including nonbooked orders, remove the filter condition from the WHERE clause in the `SDE_ORA_SalesOrderLinesFact` and `SDE_ORA_SalesOrderLinesFact_Primary` mappings.

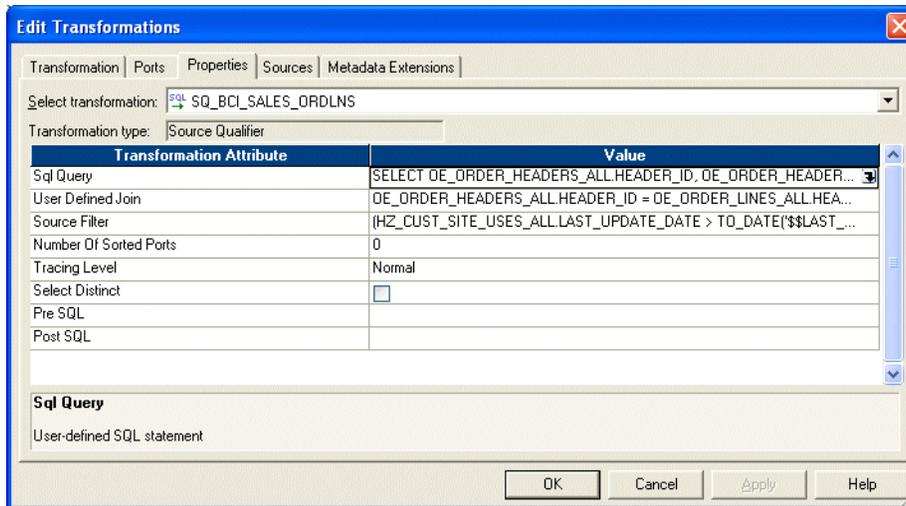
Figure 6-1 Handling Booked and Nonbooked Orders



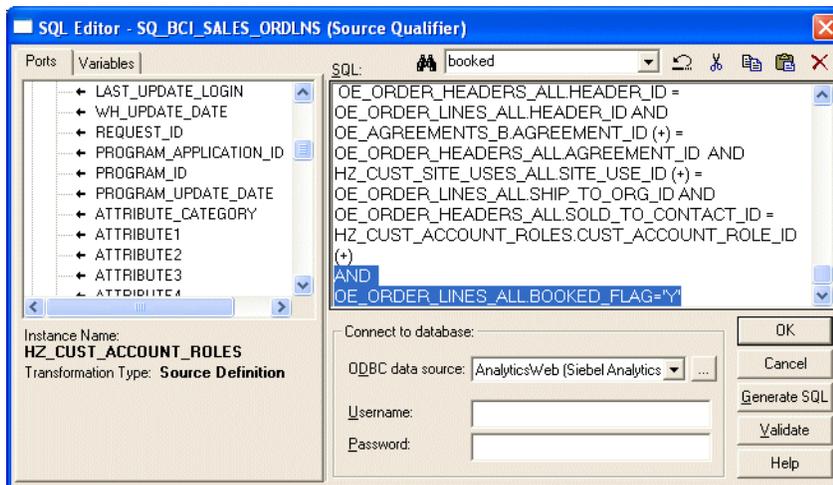
By default, only booked orders are loaded into the Sales Order Lines (`W_SALES_ORDER_LINES_F`) and Sales Booking Lines (`W_SALES_BOOKING_LINE_F`) tables. However, you can also load non-booked orders in Sales Order Lines (`W_SALES_ORDERS_LINES_F`).

To include nonbooked orders in the Sales Order Lines tables

1. In Informatica PowerCenter Designer, open the `SDE_ORA115<ver>_Adaptor` or `SDE_ORAR12_Adaptor` folder.
2. Open the `mp1t_BC_ORA_SalesOrderFact` maplet in the Maplet Designer.
3. Double-click the `SQ_BCI_SALES_ORDLNS` Source Qualifier to open the Edit Transformations box.
4. Click the Properties tab.



5. For both the Sql Query Transformation Attribute and the User Defined Join Transformation Attribute, do the following:
 - a. Select the down arrow in the Value field to display the SQL Editor box.



- b. In the SQL box, remove the line 'AND OE_ORDER_LINES_ALL.BOOKED_FLAG='Y''.
 - c. Select OK to save the changes.
6. Validate and save your changes to the repository.
7. Repeat steps 3 - 5 for the SDE_ORA_SalesOrderLinesFact_Primary mapping.

6.2.5.2.3 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.

6.2.5.2.4 About the Handling of Booked and Nonbooked Orders in the Sales Schedule Lines

Table By default, all orders loaded into the Sales Schedule Lines are booked.

However, if you want to load nonbooked orders into the Sales Schedule Lines table, you have to configure the extract so that it does not filter out nonbooked orders. In Oracle 11i and Oracle R12, the `OE_ORDER_LINES_ALL.BOOKED_FLAG = Y` condition indicates that an order is booked; therefore, this statement is used to filter out nonbooked orders. To load all orders, including nonbooked orders, remove the filter condition from the WHERE clause in the `SDE_ORA_SalesScheduleLinesFact` and `SDE_ORA_SalesScheduleLineLines_Fact_Primary` mappings.

To include nonbooked orders in the Sales Schedule Lines tables

1. In Informatica PowerCenter Designer, open the `SDE_ORA115<ver>_Adaptor` or `SDE_ORAR12_Adaptor` folder.
2. Open the `mpI_t_BC_ORA_SalesScheduleLinesFact` mapplet in the Mapplet Designer.
3. Double-click the `SQ_BCI_SALES_ORDLNS` Source Qualifier to open the Edit Transformations box.
4. Click the Properties tab.
5. For both the Sql Query Transformation Attribute and the User Defined Join Transformation Attribute, do the following:
 - a. Select the down arrow in the Value field to display the SQL Editor box.
 - b. In the SQL box, remove the line `'AND OE_ORDER_LINES_ALL.BOOKED_FLAG='Y'`.
 - c. Select OK to save the changes.
6. Validate and save your changes to the repository.
7. Repeat steps 3 - 5 for the `SDE_ORA_SalesScheduleLinesFact_Primary` mapping.

6.2.5.2.5 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 the figure below.

Figure 6–2 Sales Order Lines and Bookings Table



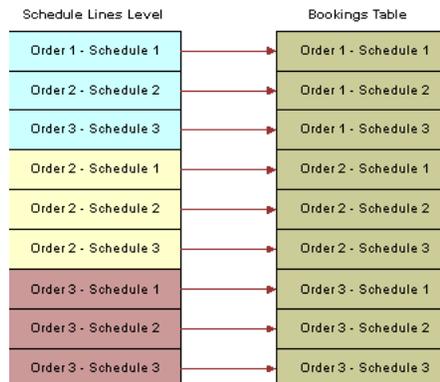
There are two subject areas in the `SDE_ORA115<ver>_Adaptor` or `SDE_ORAR12_Adaptor` containers:

- 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 the figure below. 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

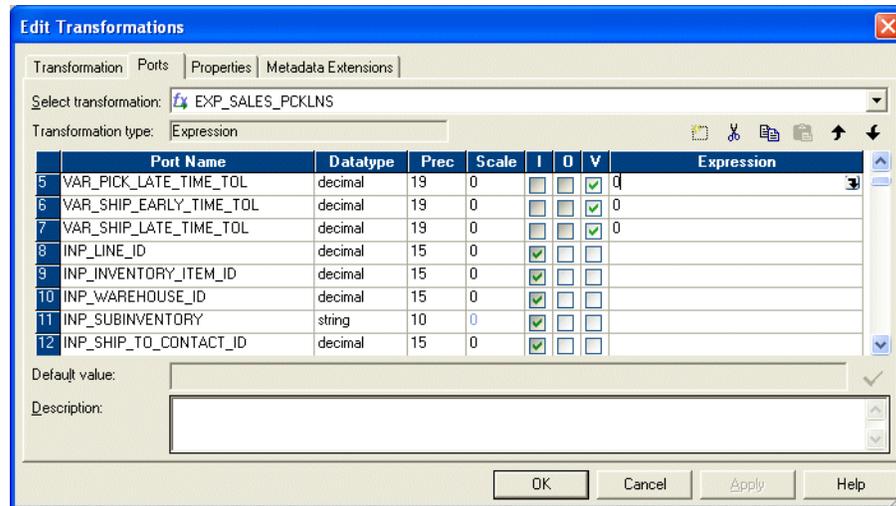


6.2.5.2.6 How to Configure Early and Late Tolerances for Shipments You configure the definition of early and late shipments by editing the `EXP_SALES_PCKLNS` expression in the `mplt_SA_ORA_SalesPickLinesFact` mapplet. The `mplt_SA_ORA_SalesPickLinesFact` mapplet is used by the `SDE_ORASalesPickLinesFact` 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. In Informatica PowerCenter Designer, open the `SDE_ORA115<ver>_Adaptor` or `SDE_ORAR12_Adaptor` folder.
2. Open the `mplt_SA_ORA_SalesPickLinesFact` mapplet in the Mapplet Designer.
3. Double-click the `EXP_SALES_PCKLNS` expression to open the Edit Transformation box.
4. Click the Ports tab.



5. Edit the expression for the ports that you want to modify.

For example:

- If you want to allow two days after the scheduled pick date before you flag the pick as late, set the Expression value for the VAR_PICK_LATE_TIME_TOL port to 2.
- To set the number of days before a pick is flagged as early, set the Expression value the VAR_PICK_EARLY_TIME_TOL port.
- To set the number of days before a pick is flagged as late, set the Expression value for the VAR_PICK_LATE_TIME_TOL port.
- If you want to change the shipping tolerances, set the Expression values for the shipping ports (for example, VAR_SHIP_LATE_TIME_TOL, VAR_SHIP_EARLY_TIME_TOL).

6. Validate and save your changes to the repository.

6.2.5.2.7 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.

6.2.5.2.8 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<ver>_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. Click 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 `mp1t_BC_ORA_SalesInvoiceLinesFact_Primary` maplet.

6.2.5.2.9 How to Configure the Order Lines Execution Plan and OTB ETL The Backlogs and Cycle Lines (OTB 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 OTB ETL and Execution Plan as follows:

To configure the Order Lines Execution Plan and OTB ETL

1. In Informatica PowerCenter Designer, open the PLP folder.
2. Open the `PLP_SalesCycleLinesFact` maplet in the Maplet Designer.
3. Double-click the `SQ_W_SALES_ORDER_LINE_F` Source Qualifier to open the Edit Transformations box.
4. Click 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, replace the SQL text 'X.TOTAL_SHIPPED_QTY' with 'PICKLINE.TOTAL_SHIPPED_QTY'.
7. In the SQL box, replace the SQL text 'X.TOTAL_INVOICED_QTY' with 'IVCLINE.TOTAL_INVOICE_QTY'.
8. Validate and save your changes to the repository.
9. In DAC, do the following: go to the Design view, click the Configuration Tags tab, and select the Sales PLP Optional Tasks.
 - a. Go to the Design view, and select the appropriate custom container.
 - b. Click the Configuration Tags tab.
 - c. Query for the tag 'Sales PLP Optional Tasks'.
 - d. Click the Subject Areas subtab.
 - e. Activate the appropriate subject areas by deselecting the Inactive check box.
10. Open the Source Qualifier in `SDE_ORA_SalesPickLinesFact`.
11. 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`.

6.2.5.2.10 How to Configure Procurement and Spend Analytics for Oracle Supply Chain and Order Management Analytics

The Oracle Supply Chain and Order Management Analytics application uses tables that are also used in Oracle's Procurement and Spend Analytics Family of Products.

For Oracle 11i and Oracle R12, you need to use the following configuration steps for Procurement and Spend Analytics to configure Oracle Supply Chain and Order Management Analytics:

- [Section 3.2.2.1, "How to Configure the Country Region and State Region Name"](#)
- [Section 3.2.2.2, "How to Configure the State Name"](#)
- [Section 3.2.2.3, "How to Configure the Country Name"](#)
- [Section 3.2.2.4, "How to Configure the Make-Buy Indicator"](#)

6.2.5.2.11 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 5.2.5.2.5, "How to Configure AR Balance ID for Oracle Receivables Analytics and Oracle General Ledger and Profitability Analytics"](#)
- [Section 5.2.5.2.6, "How to Configure the AR Adjustments Extract for Oracle Receivables Analytics"](#)
- [Section 5.2.5.2.7, "How to Configure the AR Schedules Extract"](#)
- [Section 5.2.5.2.8, "How to Configure the AR Cash Receipt Application Extract for Oracle Receivables Analytics"](#)
- [Section 5.2.5.2.9, "How to Configure the AR Credit-Memo Application Extract for Oracle Receivables Analytics"](#)

6.2.5.2.12 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 maplet (mplt_SA_ORA_SalesOrderLinesFact). The following section describes what happens if you modify the Booking ID to include the sales representative.

6.2.5.2.13 About Viewing the Data Warehouse Changes by Salesperson ID Assume you want to track changes to the sales representative for bookings and debookings. 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 maplet in the Maplet Designer.
2. Double click the MAPI_SALES_ORDLNS transformation to open the Edit Transformation box.
3. Click 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 6–9](#).

Table 6–9 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 6–9](#) is entered into the IA Bookings table (W_SALES_BOOKING_LINE_F) as shown in [Table 6–10](#).

Table 6–10 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 6–11](#).

Table 6–11 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 6–12](#).

Table 6–12 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

6.2.5.2.14 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.

6.2.5.2.15 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<ver>_Adaptor or SDE_ORAR12_Adaptor folder.
2. Open the mpIt_BC_ORA_SalesOrderLinesFact mapplet in the Mapplet Designer.
3. Double click on 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 mpIt_BC_ORA_SalesScheduleLinesFact mapplet in the Mapplet Designer.
7. Double click on 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.

6.2.5.2.16 How to Configure the Bill of Materials Explosion Options The Bill of Materials (BOM) functional area allows you to determine the profit margin of the components that comprise the finished goods. BOM allows 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.

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_BomItemFact mapping, the bompexpl.exploder_userexit stored procedure explodes the BOM structure. The table below lists the variables for the bompexpl.exploder_userexit stored procedure.

Table 6–13 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
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<Ver>_Adaptor.
2. Open the SDE_ORA_BomItemFact mapping.
3. Double-click the EXP_BOMPEXPL_EXPLODER_USEREXIT Expression transformation to open the Edit Transformations dialog, and click the Port tab to display the EXPLODE_OPTION port.
4. Change the value to 1, and click Apply.
5. Validate the mapping, and save your changes to the repository.
6. Go to Mapplet Designer, and open mplt_BC_ORA_BOMHeaderDimension.
7. Double-click the SQL qualifier SQ_BOM_INVENTORY_COMPONENTS to open the Edit Transformations dialog, and click on Properties tab, open value for SQL Query.
8. 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
```

- Click Apply, and Validate the mapping and save your changes to the repository.

To configure the BOM explosion to the Current and Future option

- In Informatica PowerCenter Designer, open the SDE_ORA<Ver>_Adaptor.
- Open the SDE_ORA_BOMItemFact mapping.
- Double-click the EXP_BOMPEXPL_EXPLODER_USEREXIT Expression transformation to open the Edit Transformations dialog, and click the Port tab to display the EXPLODE_OPTION port.
- Change the value to 3, and click Apply.
- Validate the mapping, and save your changes to the repository.
- Go to Mapplet Designer, and open mplt_BC_ORA_BOMHeaderDimension.
- Double-click the SQL qualifier SQ_BOM_INVENTORY_COMPONENTS to open the Edit Transformations dialog, and click on Properties tab, open value for SQL Query.
- 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
    
```

9. Click Apply, and Validate the mapping and save your changes to the repository.

To configure the BOM type

1. In Informatica PowerCenter Designer, open the SDE_ORA<Ver>_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 click on Properties tab, open value for SQL Query.
4. Modify `BOM_ITEM_TYPE` section in Where statement.

Change the number to your BOM type. For example, change the number to 3 for a Planning BOM type.

```

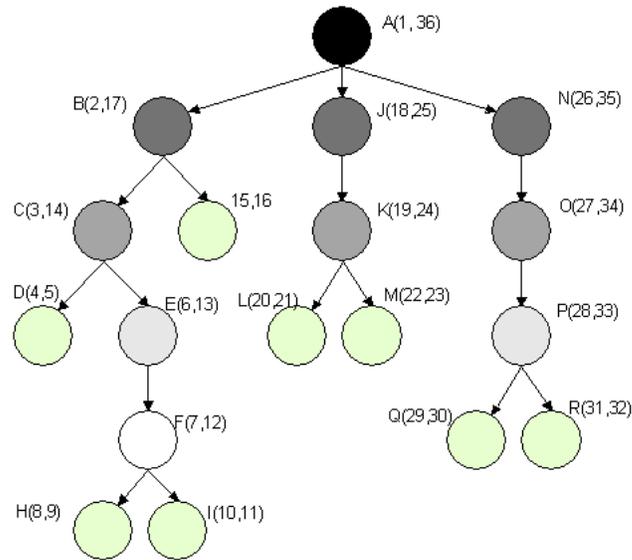
Where INV.BOM_ITEM_TYPE = 3 AND
M.BOM_ITEM_TYPE = 3 AND
    
```

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.

6.2.5.2.17 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.

The figure below 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` (for more information, see [Section 4.2.1.1, "How to Deploy Stored Procedures"](#)).

To configure the left bound and right bound calculation option

1. In Informatica PowerCenter Designer, open the `SDE_ORA<Ver>_Adaptor`.
2. Open the `SIL_BOMItemFact` mapping.
3. Double-click the `COMPUTE_BOUNDS` stored procedure transformation to open the Edit Transformations dialog, and click the Properties tab.
4. Change the value for Transformation Attributes Call Text to `compute_bounds_ora11i(1)`.
5. Click Apply.
6. Validate the mapping, and save your changes to the repository.

6.2.5.3 Configuration Steps for Oracle Supply Chain and Order Management Analytics for PeopleSoft

This section contains configuration steps that apply to PeopleSoft.

Not applicable to Oracle BI Applications release 7.9.5.1.

6.2.5.4 Configuration Steps for Oracle Supply Chain and Order Management Analytics for Universal

This section contains configuration steps that apply to Universal.

Not applicable to Oracle BI Applications release 7.9.5.1.

Configuring Oracle Human Resources Analytics

This section describes how to configure Oracle Human Resources Analytics. It contains the following topics:

- [Section 7.1, "Overview of Oracle Human Resources Analytics"](#)
- [Section 7.2, "Configuration Required Before A Full Load for Oracle HR Analytics"](#)

Note: All references to Oracle 11i also apply to Oracle R12.

7.1 Overview of Oracle Human Resources Analytics

Oracle Human Resources contains information for HR operations, employee events, and payroll.

The Oracle HR Analytics application has the following functional areas:

- **Compensation.** HR Compensation allows you to analyze the salaries, benefits, and rewards that comprise your employee compensation plan. The metrics provided as part of the application allow you to measure several areas of performance and perform a variety of comparative analyses at various levels of granularity.

It provides your company with employee payroll information that can be vital to success in today's economy. Over or under-compensating employees can both have serious effects on your company's ability to maintain a competitive edge. The HR Compensation area provides the information your HR Management department needs to manage compensation costs, such as identifying emerging trends within the organization, or within specific areas of compensation, and evaluating the effectiveness of the level of compensation as an incentive.

- **Human Resource Performance.** The information stored in the Human Resource Performance area allows you to measure several areas of performance, including contribution and productivity, HR effectiveness, and trends analytics.
- **Retention.** Under the Retention functional area you can find the events that are the hallmarks of employees' professional life cycle. These events include their hiring information, their promotional opportunities realized and not realized, the quality of the employees' job performance as measured by performance ranking, their length of service, and the reasons for termination, both voluntary and involuntary. Monitoring retention rates within departments is useful in determining potential problem areas that may want to be addressed by senior management.

- **U.S. Statutory Compliance.** The U.S. Statutory Compliance functional area stores information that helps Human Resources departments prepare government-required reports.
- **Workforce Profile.** The Workforce Profile functional area provides you with the tools to separate sensitive from non-sensitive information, and to restrict access to sensitive data. Sensitive information includes such data as ethnicity, age, native language, marital status, and performance ratings. Non-sensitive information includes information such as job title, work location, and position status.

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 Oracle HR Analytics for PeopleSoft"](#)
- [Section 7.2.4, "Configuration Steps for Oracle HR Analytics for Universal"](#)
- [Section 7.2.5, "Configuration Steps for Controlling Your Data Set"](#)

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.

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"](#)).

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, "About Domain Values and CSV Worksheet Files for Oracle HR Analytics"](#)
- [Section 7.2.2.2, "How to Configure the domainValues_Employee_Ethnic_Group_Code_ora<ver>.csv"](#)
- [Section 7.2.2.3, "How to Configure the domainValues_Employee_Sex_MF_ora<ver>.csv"](#)
- [Section 7.2.2.4, "How to Configure the domainValues_Employee_Veteran_Status_Code_ora<ver>.csv"](#)
- [Section 7.2.2.5, "How to Configure the domainValues_Employment_Cat_ora<ver>.csv"](#)
- [Section 7.2.2.6, "How to Configure the domainValues_Employment_Exempt_Flg_ora<ver>.csv"](#)
- [Section 7.2.2.7, "How to Configure the domainValues_Employment_Full_Time_Flg_ora<ver>.csv"](#)

- Section 7.2.2.8, "How to Configure the domainValues_Employment_Status_ora<ver>.csv"
- Section 7.2.2.9, "How to Configure the domainValues_EventTypes_ora<ver>.csv"
- Section 7.2.2.10, "How to Configure the domainValues_HRPosition_Active_Pos_Flg_ora<ver>.csv"
- Section 7.2.2.11, "How to Configure the domainValues_HRPosition_Exempt_Flg_ora<ver>.csv"
- Section 7.2.2.12, "How to Configure the domainValues_Job_Eeo_Cat_Code_ora<ver>.csv"
- Section 7.2.2.13, "How to Configure the domainValues_Job_Flsa_Stat_Code_ora<ver>.csv"
- Section 7.2.2.14, "How to Configure the domainValues_Pay_Type_Grp_Code_ora<ver>.csv"
- Section 7.2.2.15, "How to Configure the domainValues_Pay_Type_Flg_ora<ver>.csv"
- Section 7.2.2.16, "How to Configure Address Types for HR Profile"
- Section 7.2.2.17, "How to Configure Phone Types for HR Profile"
- Section 7.2.2.18, "How to Configure Education Degree Codes for Employee Dimension"
- Section 7.2.2.19, "About Configuring Key Flexfields"
- Section 7.2.2.20, "How to Configure the Key Flexfields for the Job Dimension"
- Section 7.2.2.21, "How to Configure the Key Flexfields for the HR Position Dimension"
- Section 7.2.2.22, "How to Configure the Key Flexfields for the Pay Grade Dimension"
- Section 7.2.2.23, "How to Configure Multi-Segmented Flexfields"
- Section 7.2.2.24, "How to Configure Flags for the Pay Type Dimension"
- Section 7.2.2.25, "How to Configure Classification Names for Payroll"

7.2.2.1 About Domain Values and CSV Worksheet Files for Oracle HR Analytics

You configure Oracle HR Analytics by mapping domain values to columns in the CSV files located in the `$pmserver\server\infa_shared\LkpFiles` folder. For example, `D:\Informatika\PowerCenter8.1.1\server\infa_shared\Lkpfiles`.

For more information on configuring domain values with CSV worksheet files, see .

For more information on domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

Note: Incorrect mappings result in inaccurate calculations of Oracle Business Intelligence metrics.

Table 7–1 lists the CSV worksheet files and the domain values for Oracle HR Analytics in the `$pmserver\server\infa_shared\LkpFiles` folder.

Table 7-1 Domain Values and CSV Worksheet Files for Oracle HR Analytics

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_ Employee_Ethnic_ Group_Code_ ora<ver>.csv	W_EMPLOYEE_D.W_ ETHNIC_GRP_CODE	Lists the Ethnic codes and their corresponding domain values of 'Ethnic Group Code' for the Oracle 11i Application.	SDE_ORA_ EmployeeDimension, SDE_ORA_ EmployeeDimension_ Full
domainValues_ Employee_Sex_MF_ ora<ver>.csv	W_EMPLOYEE_D.W_ SEX_MF_CODE	Lists the Sex codes and their corresponding domain values of 'Sex Code' for the Oracle 11i Application.	SDE_ORA_ EmployeeDimension, SDE_ORA_ EmployeeDimension_ Full
domainValues_ Employee_Veteran_ Status_Code_ ora<ver>.csv	W_EMPLOYEE_D.W_ VETERAN_STATUS_ CODE	Lists the Veteran codes and their corresponding domain values of 'Veteran Status Code' for the Oracle 11i Application.	SDE_ORA_ EmployeeDimension, SDE_ORA_ EmployeeDimension_ Full
domainValues_ Employment_Cat_ ora<ver>.csv	W_EMPLOYMENT_ D.W_EMPLOYEE_ CAT_CODE	Lists the User Person Types and their corresponding domain values of 'Employment Category Code' for the Oracle 11i Application.	SDE_ORA_ Employment Dimension, SDE_ ORA_Employment Dimension_Full
domainValues_ Employment_ Exempt_Flg_ ora<ver>.csv	W_EMPLOYMENT_ D.W_EXEMPT_FLG	Lists the FLSA Statuses and their corresponding domain values of 'Exempt Flag' for the Oracle 11i Application.	SDE_ORA_ Employment Dimension, SDE_ ORA_Employment Dimension_Full
domainValues_ Employment_Full_ Time_Flg_ ora<ver>.csv	W_EMPLOYMENT_ D.W_FULL_TIME_ FLG	Lists the Employment Categories and their corresponding domain values of 'Full Time Flag' for the Oracle 11i Application.	SDE_ORA_ Employment Dimension, SDE_ ORA_Employment Dimension_Full
domainValues_ Employment_Status_ ora<ver>.csv	W_EMPLOYMENT_ D.W_ EMPLOYMENT_ STAT_CODE	Lists the Per System Statuses and their corresponding domain values of 'Employment Status' for the Oracle 11i Application.	SDE_ORA_ Employment Dimension, SDE_ ORA_Employment Dimension_Full

Table 7-1 (Cont.) Domain Values and CSV Worksheet Files for Oracle HR Analytics

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_EventTypes_ora<ver>.csv	W_EVENT_TYPE_ D.W_EVENT_CLASS, W_EVENT_TYPE_ D.W_EVENT_GRP_ CODE, W_EVENT_ TYPE_D.W_EVENT_ REASON_CODE, W_ EVENT_SUBG_CODE	Lists the Event Types, Event Codes and Meanings and their corresponding domain values of 'Event Group', 'Event Sub-Group' and 'Event Reason' for the Oracle 11i Application.	SDE_ORA_ EventTypeDimension _AbsenceAttendance, SDE_ORA_ EventTypeDimension _AbsenceAttendance_ Full, SDE_ORA_ EventTypeDimension _AdditionalEvents_ FromFile, SDE_ORA_ EventTypeDimension _OtherHREvents, SDE_ORA_ EventTypeDimension _OtherHREvents_Full
domainValues_HRPosition_Active_Pos_Flg_ora<ver>.csv	W_HR_POSITION_ D.W_ACTIVE_ POSITION_FLG	Lists the Position Statuses and their corresponding domain values of 'Active Position Flag' for the Oracle 11i Application.	SDE_ORA_ HRPositionDimension , SDE_ORA_ HRPositionDimension
domainValues_HRPosition_Exempt_Flg_ora<ver>.csv	W_HR_POSITION_ D.W_EXEMPT_FLG	Lists the FLSA Statuses and their corresponding domain values of 'Exempt Flag' for the Oracle 11i Application.	SDE_ORA_ HRPositionDimension , SDE_ORA_ HRPositionDimension
domainValues_Job_Eeo_Cat_Code_ora<ver>.csv	W_JOB_D.W_EEO_ JOB_CAT_CODE	Lists the EEO Job Categories and their corresponding domain values of 'EEO Job Category' for the Oracle 11i Application.	SDE_ORA_ JobDimension, SDE_ ORA_JobDimension_ Full
domainValues_Job_Flsa_Stat_Code_ora<ver>.csv	W_JOB_D.W_FLSA_ STAT_CODE	Lists the FLSA Statuses and their corresponding domain values of 'FLSA Status Code' for the Oracle 11i Application.	SDE_ORA_ JobDimension, SDE_ ORA_JobDimension_ Full
domainValues_Pay_Type_Grp_Code_ora<ver>.csv	W_PAY_TYPE_D.W_ PAY_TYPE_GRP_ CODE	Lists the Classification Names, Element Names and their corresponding domain values of 'Pay Type Group Code' for the Oracle 11i Application.	SDE_ORA_ PayTypeDimension, SDE_ORA_ PayTypeDimension_ Full

Table 7-1 (Cont.) Domain Values and CSV Worksheet Files for Oracle HR Analytics

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_Pay_Type_Flg_ora<ver>.csv	W_PAY_TYPE_D.W_PAY_TYPE_FLG	Lists the Costing Debit or Credit values and their corresponding domain values of 'Pay type Flag' for the Oracle 11i Application.	SDE_ORA_PayTypeDimension, SDE_ORA_PayTypeDimension_Full
domainValues_Employee_Sex_MF_ora<ver>.csv	W_EMPLOYEE_D.W_SEX_MF_CODE	Lists the Sex codes and their corresponding domain values of 'Sex Code' for the Oracle 11i Application.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employee_Veteran_Status_Code_ora<ver>.csv	W_EMPLOYEE_D.W_VETERAN_STATUS_CODE	Lists the Veteran codes and their corresponding domain values of 'Veteran Status Code' for the Oracle 11i Application.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employment_Cat_ora<ver>.csv	W_EMPLOYMENT_D.W_EMPLOYEE_CAT_CODE	Lists the User Person Types and their corresponding domain values of 'Employment Category Code' for the Oracle 11i Application.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full
domainValues_Employment_Exempt_Flg_ora<ver>.csv	W_EMPLOYMENT_D.W_EXEMPT_FLG	Lists the FLSA Statuses and their corresponding domain values of 'Exempt Flag' for the Oracle 11i Application.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full
domainValues_Employment_Full_Time_Flg_ora<ver>.csv	W_EMPLOYMENT_D.W_FULL_TIME_FLG	Lists the Employment Categories and their corresponding domain values of 'Full Time Flag' for the Oracle 11i Application.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full
domainValues_Employment_Status_ora<ver>.csv	W_EMPLOYMENT_D.W_EMPLOYMENT_STAT_CODE	Lists the Per System Statuses and their corresponding domain values of 'Employment Status' for the Oracle 11i Application.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full

Table 7-1 (Cont.) Domain Values and CSV Worksheet Files for Oracle HR Analytics

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_EventTypes_ora<ver>.csv	W_EVENT_TYPE_ D.W_EVENT_CLASS, W_EVENT_TYPE_ D.W_EVENT_GRP_ CODE, W_EVENT_ TYPE_D.W_EVENT_ REASON_CODE, W_ EVENT_SUBG_CODE	Lists the Event Types, Event Codes and Meanings and their corresponding domain values of 'Event Group', 'Event Sub-Group' and 'Event Reason' for the Oracle 11i Application.	SDE_ORA_ EventTypeDimension _AbsenceAttendance, SDE_ORA_ EventTypeDimension _AbsenceAttendance_ Full, SDE_ORA_ EventTypeDimension _AdditionalEvents_ FromFile, SDE_ORA_ EventTypeDimension _OtherHREvents, SDE_ORA_ EventTypeDimension _OtherHREvents_Full
domainValues_HRPosition_Active_Pos_Flg_ora<ver>.csv	W_HR_POSITION_ D.W_ACTIVE_ POSITION_FLG	Lists the Position Statuses and their corresponding domain values of 'Active Position Flag' for the Oracle 11i Application.	SDE_ORA_ HRPositionDimension , SDE_ORA_ HRPositionDimension
domainValues_HRPosition_Exempt_Flg_ora<ver>.csv	W_HR_POSITION_ D.W_EXEMPT_FLG	Lists the FLSA Statuses and their corresponding domain values of 'Exempt Flag' for the Oracle 11i Application.	SDE_ORA_ HRPositionDimension , SDE_ORA_ HRPositionDimension
domainValues_Job_Eeo_Cat_Code_ora<ver>.csv	W_JOB_D.W_EEO_ JOB_CAT_CODE	Lists the EEO Job Categories and their corresponding domain values of 'EEO Job Category' for the Oracle 11i Application.	SDE_ORA_ JobDimension, SDE_ ORA_JobDimension_ Full
domainValues_Job_Flsa_Stat_Code_ora<ver>.csv	W_JOB_D.W_FLSA_ STAT_CODE	Lists the FLSA Statuses and their corresponding domain values of 'FLSA Status Code' for the Oracle 11i Application.	SDE_ORA_ JobDimension, SDE_ ORA_JobDimension_ Full
domainValues_Pay_Type_Grp_Code_ora<ver>.csv	W_PAY_TYPE_D.W_ PAY_TYPE_GRP_ CODE	Lists the Classification Names, Element Names and their corresponding domain values of 'Pay Type Group Code' for the Oracle 11i Application.	SDE_ORA_ PayTypeDimension, SDE_ORA_ PayTypeDimension_ Full

Table 7-1 (Cont.) Domain Values and CSV Worksheet Files for Oracle HR Analytics

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_Pay_Type_Flg_ora<ver>.csv	W_PAY_TYPE_D.W_PAY_TYPE_FLG	Lists the Costing Debit or Credit values and their corresponding domain values of 'Pay type Flag' for the Oracle 11i Application.	SDE_ORA_PayTypeDimension, SDE_ORA_PayTypeDimension_Full

Note: When editing CSV files, make sure 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.

7.2.2.2 How to Configure the domainValues_Employee_Ethnic_Group_Code_ora<ver>.csv

This section explains how to configure the domainValues_Employee_Ethnic_Group_Code_ora<ver>.csv.

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', 'BA', 'BC', 'BO', 'C', 'I', 'O',
'P', 'W')
ORDER BY 1
```

2. Using a text editor, open the domainValues_Employee_Ethnic_Group_Code_ora<ver>.csv file, located in the \$pmsserver\lkpfiles folder.
3. Copy the PER_INFORMATION1 to the ETHNIC_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each Ethnic Code to one domain value.
5. Save and close the file.

7.2.2.3 How to Configure the domainValues_Employee_Sex_MF_ora<ver>.csv

This section explains how to configure the domainValues_Employee_Sex_MF_ora<ver>.csv.

1. Identify the Sex Codes in your Oracle source system by using the following SQL:

```
SELECT DISTINCT SEX FROM PER_ALL_PEOPLE_F ORDER BY 1
```

2. Using a text editor, open the domainValues_Employee_Sex_MF_ora<ver>.csv file, located in the \$pmsserver\lkpfiles folder.

3. Copy the SEX column to the SEX column in the file. The data must be copied starting from the 6th line.
4. Map each Sex Code to one domain value.
5. Save and close the file.

7.2.2.4 How to Configure the domainValues_Employee_Veteran_Status_Code_ora<ver>.csv

This section explains how to configure the domainValues_Employee_Veteran_Status_Code_ora<ver>.csv

1. Identify the Veteran Status Codes in your Oracle source system by using the following SQL:

```
SELECT DISTINCT PER_INFORMATION5 FROM PER_ALL_PEOPLE_F
WHERE PER_INFORMATION5 in ('NOTVET', 'OTEDV', 'VET', 'VETDIS', 'VIETVET',
'VIETVETDIS')
ORDER BY 1
```

2. Using a text editor, open the domainValues_Employee_Veteran_Status_Code_ora<ver>.csv file, located in the \$pmserver\lkpfiles folder.
3. Copy the PER_INFORMATION5 column to the VETERAN_STATUS_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each Veteran Status Code to one domain value.
5. Save and close the file.

7.2.2.5 How to Configure the domainValues_Employment_Cat_ora<ver>.csv

This section explains how to configure the domainValues_Employment_Cat_ora<ver>.csv.

1. Identify the User Person Types in your Oracle source system by using the following SQL:

```
SELECT DISTINCT SYSTEM_PERSON_TYPE, USER_PERSON_TYPE FROM
PER_PERSON_TYPES
WHERE SYSTEM_PERSON_TYPE IN
('EMP', 'OTHER', 'EMP_APL', 'EX_EMP', 'EX_EMP_APL', 'RETIREE', 'PRTN')
ORDER BY 1,2
```

2. Using a text editor, open the domainValues_Employment_Cat_ora<ver>.csv file, located in the \$pmserver\lkpfiles folder.
3. Copy the User Person Types to the USER_PERSON_TYPE column in the file. The data must be copied starting from the 6th line.
4. Map each User Person Type (results of the SQL query) to one of the delivered W_EMPLOYEE_CATEGORY_CODE domain values in the delivered example csv file.

You can map more than one User Person Type to the same W_EMPLOYEE_CATEGORY_CODE by adding in new rows. For example:

Contractor	CONTRACTOR	CONTRACTOR
Contingent Worker	CONTRACTOR	CONTRACTOR

System Person Types are also extracted with User Person Type to help you map the domain values. Do not copy the System Person types in the CSV file.

5. Save and close the file.

7.2.2.6 How to Configure the domainValues_Employment_Exempt_Flg_ora<ver>.csv

This section explains how to configure the domainValues_Employment_Exempt_Flg_ora<ver>.csv

1. Identify the FLSA Statuses in your Oracle source system by using the following SQL:

```
SELECT DISTINCT JOB_INFORMATION3 FROM PER_JOBS ORDER BY 1
```

2. Using a text editor, open the domainValues_Employment_Exempt_Flg_ora<ver>.csv file, located in the \$pmsserver\lkpfiles folder.
3. Copy the JOB_INFORMATION3 to the FLSA_STATUS_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each FLSA_STATUS_CODE to one domain value.
5. Save and close the file.

7.2.2.7 How to Configure the domainValues_Employment_Full_Time_Flg_ora<ver>.csv

This section explains how to configure the domainValues_Employment_Full_Time_Flg_ora<ver>.csv.

1. Identify the Employment Categories in your Oracle source system by using the following SQL:

```
SELECT DISTINCT EMPLOYMENT_CATEGORY FROM PER_ALL_ASSIGNMENTS_F ORDER BY 1
```

2. Using a text editor, open the domainValues_Employment_Full_Time_Flg_ora<ver>.csv file, located in the \$pmsserver\lkpfiles folder.
3. Copy the Employment Category to the EMPLOYMENT_CATEGORY column in the file. The data must be copied starting from the 6th line.
4. Map each EMPLOYMENT_CATEGORY to one domain value.
5. Save and close the file.

7.2.2.8 How to Configure the domainValues_Employment_Status_ora<ver>.csv

This section explains how to configure the domainValues_Employment_Status_ora<ver>.csv

1. Identify the Per System Statuses in your Oracle source system by using the following SQL:

```
SELECT DISTINCT PER_SYSTEM_STATUS FROM PER_ASSIGNMENT_STATUS_TYPES  
ORDER BY 1
```

2. Using a text editor, open the domainValues_Employment_Status_ora<ver>.csv file, located in the \$pmsserver\lkpfiles folder.
3. Copy the PER_SYSTEM_STATUS to the PER_SYSTEM_STATUS column in the file. The data must be copied starting from the 6th line.
4. Map each PER_SYSTEM_STATUS to one domain value.
5. Save and close the file.

7.2.2.9 How to Configure the domainValues_EventTypes_ora<ver>.csv

This section explains how to configure the domainValues_EventTypes_ora<ver>.csv

1. Identify the Event Types in your Oracle source system by using the following SQL:

```
SELECT DISTINCT LOOKUP_TYPE, LOOKUP_CODE, MEANING
FROM FND_LOOKUP_VALUES
WHERE LOOKUP_TYPE IN
('EMP_ASSIGN_REASON',
'LEAV_REAS',
'PROPOSAL_REASON')
ORDER BY 1, 2, 3
```

2. Using a text editor, open the domainValues_EventTypes_ora<ver>.csv file, located in the \$pmsserver\lkpfiles folder.
3. Copy the Lookup Type, Lookup Code and Meaning to the LOOKUP_TYPE, LOOKUP_CODE, and MEANING columns in the file respectively. The data must be copied starting from the 6th line. Use commas to separate the entries.
4. Map each Event Type (LOOKUP_CODE) to one domain value for each of the 3 domain columns — W_EVENT_GRP_CODE, W_EVENT_SUBG_CODE, and W_EVENT_REASON_CODE. Event Category (LOOKUP_TYPE) and Event Description (MEANING) are also extracted with Event Type to help you map the domain values.
5. Save and close the file.

7.2.2.10 How to Configure the domainValues_HRPosition_Active_Pos_Flg_ora<ver>.csv

This section explains how to configure the domainValues_HRPosition_Active_Pos_Flg_ora<ver>.csv.

1. Identify the Position Statuses in your Oracle source system by using the following SQL:

```
SELECT DISTINCT STATUS FROM HR_ALL_POSITIONS_F ORDER BY 1
```

2. Using a text editor, open the domainValues_HRPosition_Active_Pos_Flg_ora<ver>.csv file, located in the \$pmsserver\lkpfiles folder.
3. Copy the STATUS to the STATUS column in the file. The data must be copied starting from the 6th line.
4. Map each position STATUS to one domain value.
5. Save and close the file.

7.2.2.11 How to Configure the domainValues_HRPosition_Exempt_Flg_ora<ver>.csv

This section explains how to configure the domainValues_HRPosition_Exempt_Flg_ora<ver>.csv.

1. Identify the FLSA Statuses in your Oracle source system by using the following SQL:

```
SELECT DISTINCT JOB_INFORMATION3 FROM PER_JOBS
ORDER BY 1
```

2. Using a text editor, open the domainValues_HRPosition_Exempt_Flg_ora<ver>.csv file, located in the \$pmsserver\lkpfiles folder.

3. Copy the JOB_INFORMATION3 to the FLSA_STATUS_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each FLSA_STATUS_CODE to one domain value.
5. Save and close the file.

7.2.2.12 How to Configure the domainValues_Job_Eeo_Cat_Code_ora<ver>.csv

This section explains how to configure the domainValues_Job_Eeo_Cat_Code_ora<ver>.csv

1. Identify the EEO Job Categories in your Oracle source system by using the following SQL:

```
SELECT DISTINCT JOB_INFORMATION1 FROM PER_JOBS
ORDER BY 1
```

2. Using a text editor, open the domainValues_Job_Eeo_Cat_Code_ora<ver>.csv file, located in the \$pmsserver\lkpfiles folder.
3. Copy the JOB_INFORMATION1 to the EEO_JOB_CAT_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each EEO_JOB_CAT_CODE to one domain value.
5. Save and close the file.

7.2.2.13 How to Configure the domainValues_Job_Flsa_Stat_Code_ora<ver>.csv

This section explains how to configure the domainValues_Job_Flsa_Stat_Code_ora<ver>.csv.

1. Identify the FLSA Statuses in your Oracle source system by using the following SQL:

```
SELECT DISTINCT JOB_INFORMATION3 FROM PER_JOBS
ORDER BY 1
```

2. Using a text editor, open the domainValues_Job_Flsa_Stat_Code_ora<ver>.csv file, located in the \$pmsserver\lkpfiles folder.
3. Copy the JOB_INFORMATION3 to the FLSA_STAT_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each FLSA_STAT_CODE to one domain value.
5. Save and close the file.

7.2.2.14 How to Configure the domainValues_Pay_Type_Grp_Code_ora<ver>.csv

This section explains how to configure the domainValues_Pay_Type_Grp_Code_ora<ver>.csv.

1. Identify the Pay Elements in your Oracle source system by using the following SQL:

```
SELECT DISTINCT CLASSIFICATION_NAME, ELEMENT_NAME
FROM
PAY_ELEMENT_TYPES_F,
PAY_ELEMENT_CLASSIFICATIONS
WHERE
PAY_ELEMENT_CLASSIFICATIONS.CLASSIFICATION_ID = PAY_ELEMENT_TYPES_
F.CLASSIFICATION_ID AND
CLASSIFICATION_NAME NOT LIKE '%Information%' AND
```

```
CLASSIFICATION_NAME NOT LIKE '%Employer%' AND
CLASSIFICATION_NAME NOT LIKE '%Balance%'
ORDER BY 1, 2
```

2. Using a text editor, open the domainValues_Pay_Type_Grp_Code_ora<ver>.csv file, located in the \$pmsserver\lkpfiles folder.
3. Copy the Classification Name and Element Name to the CLASSIFICATION_NAME and ELEMENT_NAME columns in the file respectively. The data must be copied starting from the 6th line. Use commas to separate the entries.

Note: Do not change lines one to four in the domainValues_Pay_Type_Grp_Code_ora<ver>.csv file.

4. Map each Element Name to one domain value. Classification Names are also extracted with Element Names to help you map the domain values. If the element is not related to Payroll Pay Check, you can map the element to 'OTHER'.
5. Save and close the file.

7.2.2.15 How to Configure the domainValues_Pay_Type_Flg_ora<ver>.csv

This section explains how to configure the domainValues_Pay_Type_Flg_ora<ver>.csv.

1. Identify the Costing (Debit or Credit) in your Oracle source system by using the following SQL:

```
SELECT DISTINCT COSTING_DEBIT_OR_CREDIT FROM PAY_ELEMENT_CLASSIFICATIONS
ORDER BY 1
```

2. Using a text editor, open the domainValues_Pay_Type_Flg_ora<ver>.csv file, located in the \$pmsserver\lkpfiles folder.
3. Copy the COSTING_DEBIT_OR_CREDIT to the COSTING_DEBIT_OR_CREDIT column in the file. The data must be copied starting from the 6th line.
4. Map each _DEBIT_OR_CREDIT to one domain value.
5. Save and close the file.

7.2.2.16 How to Configure Address Types for HR Profile

There are three address fields in the Employee dimension table:

- Permanent address
- Mail address
- Work address

For each of these, we use only the primary ones. In addition, the following logic is used to determine the various types of addresses:

- Permanent: Address Type = 'H' (This is also the home address, in other words)
- Mail: Address Type = 'M'. If this is not available, use Permanent address (which can be the best alternate for mailing address).
- Work: Address Type = 'Default'. If this is not available, use Permanent address (which can be the best alternate for mailing address).

You can modify this logic if required. For example, if your system tracks work address with a special address type 'W', then you should be able to modify the existing logic.

Or, if you do not want to assume that the mail address (M) is same as the home address (H), you may want to remove the null evaluation check there.

To configure Address Type

1. In Informatica PowerCenter Designer, open the SDE_ORA1158_Adaptor folder.
2. In Mapplet Designer, open the mplt_SA_ORA_EmployeeDimension mapplet.
3. Locate the Expression transformation Exp_SA_Employees_Validate and scroll down the ports to get to the variables 'HOME_ADDRESS_ID_VAR', 'MAIL_ADDRESS_ID_VAR' and 'DEFAULT_ADDRESS_ID_VAR'.

For example, if you have a specific address type for work addresses, (for example 'W'), you would modify the expression for the variable port DEFAULT_ADDRESS_ID_VAR as follows:

From: :LKP.Lkp_Address_Id(INP_PERSON_ID, INP_EFFECTIVE_END_DATE, 'Y', 'Default', INP_DATASOURCE_NUM_ID)

To: :LKP.Lkp_Address_Id(INP_PERSON_ID, INP_EFFECTIVE_END_DATE, 'Y', 'W', INP_DATASOURCE_NUM_ID)

The output Address ID values are the next three ports 'PERM_ADDRESS_ID', 'MAIL_ADDRESS_ID' and 'WORK_ADDRESS_ID'. If you do not want to assume that the mail address (M) is same as the home address (H) in the event of mail address not being available, then you would modify the logic in the outgoing port 'MAIL_ADDRESS_ID' as follows:

From: IIF(ISNULL(MAIL_ADDRESS_ID_VAR), HOME_ADDRESS_ID_VAR, MAIL_ADDRESS_ID_VAR)

To: MAIL_ADDRESS_ID_VAR

4. Validate and save changes to the repository. If you are using the version controlling for the Informatica Repository, you will have to check in your changes as well.

7.2.2.17 How to Configure Phone Types for HR Profile

There are four phone related fields in the Employee dimension table, as follows:

- Fax
- Work phone
- Pager
- Mobile phone

The following logic is used to arrive at the various types of addresses:

- Fax: Phone Type = 'WF' (Work Fax)
- Work Phone: Phone Type = 'W1' (First work phone, if there are more than one)
- Pager: Phone Type = 'P' (Pager)
- Mobile: Phone Type = 'M' (Mobile)

You can modify this logic if required. For example, if your system tracks the primary work phone with a special phone type 'WP', instead of W1, then you should be able to modify the existing logic. The same applies for other phone types as well.

To configure Phone Type

1. In Informatica PowerCenter Designer, open the SDE_ORA1158_Adaptor folder.

2. In Mapplet Designer, open the `mplt_SA_ORA_EmployeeDimension` mapplet.
3. Locate the Expression transformation `Exp_SA_Employee` and scroll down the ports to get to the output ports 'EXT_FAX_PH_NUM', 'EXT_WORK_PHONE', 'EXT_PAGER_NUM' and 'EXT_MOBILE_NUM'.

For example, if you have a specific phone type for the primary work phone, 'WP', you would modify the expression for the output port `EXT_WORK_PHONE` as follows:

From: `:LKP.Lkp_Phone_Number(INP_PERSON_ID, 'W1', INP_EFFECTIVE_END_DATE, INP_DATASOURCE_NUM_ID)`

To: `:LKP.Lkp_Phone_Number(INP_PERSON_ID, 'WP', INP_EFFECTIVE_END_DATE, INP_DATASOURCE_NUM_ID)`

4. Validate and save changes to the repository. If you are using the version controlling for the Informatica Repository, you will have to check in your changes as well.

7.2.2.18 How to Configure Education Degree Codes for Employee Dimension

This section explains how to configure the categories that are used to evaluate the highest education degree code for an employee.

The 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 your custom container from the drop-down list.
2. Click the Tasks tab, and select the `SDE_ORA_EmployeeDimension` task.
3. Click 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 please login to Oracle E-Business Suite Instance using SQL*Plus and execute the following SQL:

```
SELECT lookup_code, meaning FROM hr_lookups WHERE lookup_type = 'PER_CATEGORIES'
```

Select from the returned `lookup_code` column values and decide which ones are used to identify Education Degrees.

7.2.2.19 About Configuring Key Flexfields

This section explains how to configure Key Flexfields in Oracle EBS applications.

In Oracle EBS Application module, a Flexfield is a field that allows a user to customize the Oracle applications. Each Flexfield in Oracle in turn consists of sub-fields called segments. Each segment is assigned a value set. The value set consists of values.

The main purpose of using Flexfields in Oracle EBS Applications is to customize the applications according to the business environment and practices. Oracle EBS Application allows a user to validate values that are entered in these flexfields. A user can change the structure of a Flexfield depending on the data in the application.

Since Flexfield configurations vary from implementation to implementation, in order to accommodate all kinds of flexfield configurations done at the Oracle EBS Applications end, the attributes sourced from Flexfield columns is parameterized.

To configure Oracle HR Analytics to support flexfields, do the following:

- [Section 7.2.2.20, "How to Configure the Key Flexfields for the Job Dimension"](#)
- [Section 7.2.2.21, "How to Configure the Key Flexfields for the HR Position Dimension"](#)
- [Section 7.2.2.22, "How to Configure the Key Flexfields for the Pay Grade Dimension"](#)
- [Section 7.2.2.23, "How to Configure Multi-Segmented Flexfields"](#)

7.2.2.20 How to Configure the Key Flexfields for the Job Dimension

This section is about configuring Job Code, Job Name and the Job Family Code columns.

To configure the Key Flexfields for the Job Dimension

1. In DAC, go to the Design view, and select your custom container from the drop-down list.
2. Click the Tasks tab, and select the SDE_ORA_JobDimension task.
3. Specify or create the following parameters with an appropriate value.
 - \$\$JOBCODE_FLXFLD_SEGMENT_COL = the Column Name of Job Code.
 - \$\$JOBFAMILYCODE_FLXFLD_SEGMENT_COL = the Column name of Job Group Code.
4. In the Tasks tab, and select the SDE_ORA_CodeDimension_Job task.
5. Specify or create the following parameters with an appropriate value.
 - \$\$JOBCODE_FLXFLD_SEGMENT_COL = the Column Name of Job Code.
 - \$\$JOBNAME_FLXFLD_SEGMENT_COL = the Column Name of the Job Name.
6. Save your changes.

To find out what columns are defined in the Flexfield, do the following:

1. Login with Application Developer Responsibility.
2. Under the Key Flexfield, click on Segments.
3. Query for the Application Human Resources and locate the Flexfield Title relevant to Job.
4. Click on the Segments corresponding to the Flexfield that is being used in order to find the columns being used for the Job Code and Job Name.

7.2.2.21 How to Configure the Key Flexfields for the HR Position Dimension

This section is about configuring the Position Number column.

To configure the Key Flexfields for the HR Position Dimension

1. In DAC, go to the Design view, and select your custom container from the drop-down list.
2. Click the Tasks tab, and select the SDE_ORA_HRPositionDimension task.
3. Specify or create the following parameter with an appropriate value.
 - \$\$POSITION_NUM_FLXFLD_SEGMENT_COL = the Column Name of Position Num.
4. In the Tasks tab, and select the SDE_ORA_HRPositionDimension_Full task.
5. Specify or create the following parameter with an appropriate value.
 - \$\$POSITION_NUM_FLXFLD_SEGMENT_COL = the Column Name of Position Num.
6. Save your changes.

To find out what columns are defined in the Flexfield, do the following:

1. Login with Application Developer Responsibility.
2. Under the Key Flexfield, click on Segments.
3. Query for the Application Human Resources and locate the Flexfield Title relevant to Position.
4. Click on the Segments corresponding to the Flexfield that is being used in order to find the columns being used for the Position Num.

7.2.2.22 How to Configure the Key Flexfields for the Pay Grade Dimension

This section explains how to configure the Pay Level Name column.

To configure the Key Flexfields for the Pay Grade Dimension

1. In DAC, go to the Design view, and select your custom container from the drop-down list.
2. Click the Tasks tab, and select the SDE_ORA_CodeDimension_Pay_Level task.
3. Specify or create the following parameter with an appropriate value.
 - \$\$PAYLVLNAME_FLXFLD_SEGMENT_COL = the Column Name for the Pay level Name.
4. Save your changes.

To find out what columns are defined in the Flexfield, do the following:

1. Login with Application Developer Responsibility.
2. Under the Key Flexfield, click on Segments.
3. Query for the Application Human Resources and locate the Flexfield Title relevant to Job.
4. Click on the Segments corresponding to the Flexfield that is being used in order to find the columns being used for the Job Code and Job Name.

7.2.2.23 How to Configure Multi-Segmented Flexfields

This section explains how to configure HR Analytics for multi-segmented Flexfields for any column.

In case your Flexfield configuration for a given business column involves more than one segment column, or in case you want to report a business column as a concatenation of more than one segment column separated by a character (for example, a period '.'), then the corresponding parameter can be used to provide the information correctly. The example of configuring Job Name flexfield is provided below to illustrate the process.

When Oracle BI Applications is installed out-of-the-box, the Informatica parameter for the task 'SDE_ORA_CodeDimension_Job' is set to the following:

```
$$JOBNAME_FLXFLD_SEGMENT_COL=SEGMENT6
```

If the Job Name in your deployment is configured using SEGMENT3 together with SEGMENT6 (or if you want to report Job Name as a concatenation of these two with 'dot' as the separator), you should set the Informatica parameter value as:

```
$$JOBNAME_FLXFLD_SEGMENT_COL=SEGMENT3 || SEGMENT6
```

Or, if you want a period (that is '.') in between, set the Informatica parameter value as:

```
$$JOBNAME_FLXFLD_SEGMENT_COL=SEGMENT3 || '.' || SEGMENT6
```

If your OLTP table already stores the concatenated value in some other column (like NAME), you could use it directly as follows:

```
$$JOBNAME_FLXFLD_SEGMENT_COL=NAME
```

Note: You must make sure you take a look at the SQL code (in Informatica PowerCenter Designer Source Qualifier object) before setting the parameter values. For example, before configuring the values for the Informatica parameter \$\$JOBNAME_FLXFLD_SEGMENT_COL (as in the above example), you should know which table the data will be pulled in. If necessary, use the following steps to obtain OLTP table names.

To obtain the OLTP table names:

1. In DAC, go to the Design view, and select your custom container from the drop-down list.
2. Click the Tasks tab, and select the SDE_ORA_JobDimension and SDE_ORA_JobDimension tasks (using the example job name flexfield example).
3. Note the value of the following parameter (.
 - \$\$JOBNAME_FLXFLD_SEGMENT_COL
 In this case, the maplet name that you need to derive is mplt_BC_ORA_Codes_Job.
4. Launch Informatica PowerCenter Designer and log on to the repository and navigate to the appropriate folder.
5. Open the maplet found in Step#2, open the Source Qualifier object, and see the SQL override. Get the physical OLTP table name from here. In this case, the table name is 'PER_JOB_DEFINITIONS'.

Once you know the OLTP table names, you can work out the column names to use in the parameter file (SEGMENT3 or SEGMENT6 or NAME and so on) that really exist in the table PER_JOB_DEFINITIONS.

7.2.2.24 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, or Regular Compensation Earning type or a Pension Compensation Earning type. These Flag columns are chosen to contribute towards slowly changing dimension of type 2, and hence should be configured correctly.

Each of these three Flag columns are evaluated based on the Incoming Classification Name using a Flat File lookup. This Flat file needs to be configured to see the accurate values of the Pay Type Flags in the DataWarehouse.

To Configure Flags for the Pay Type Dimension

1. Open the file_comp_pension_taxable_flg_ora<ver>.csv from Informatica Server\LkpFiles directory.
2. Add all possible Classification Names that are being used in your business.
3. Corresponding to the Classification Name, add the relevant values for the Pay Type Flags. Make sure to use either 'Y' or 'N'.

In Oracle HR Analytics, these flags apply to Earnings only. Therefore, you should only need to configure Classification Names related to Earnings. For other areas, the default Informatica mapping values should be acceptable.

7.2.2.25 How to Configure Classification Names for Payroll

This section explains how to configure classification names for Payroll.

The Aggregated items are loaded into the Payroll Fact table with a DETAIL_FLG = 'N', apart from the line items. The Aggregated Line items that are loaded are: TOTAL_GROSS, NET_PAY, TOTAL_DEDUCTIONS and TOTAL_TAXES.

The following parameters need to be configured to get accurate results. Each Aggregated Line item is computed by grouping by elements that belong to a certain Classification name set.

Table 7-2 Classification Name Parameters

Parameter Name	Description
\$\$GROSS_PAY_CLASSIFICATION_NAME	Add all the Classification Names that add up to the Total Gross Pay.
\$\$TOTAL_DEDUCTION_CLASSIFICATION_NAME	Add all the Classification Names that correspond to any deduction.
\$\$TOTAL_TAXES_CLASSIFICATION_NAME	Add all the Classification Names that correspond to various Taxes.
\$\$TOTAL_GROSS_FOR_NETPAY_EARNINGS	Add all the Earnings Classification Names that are realized in Net pay. Note that some Classification Names are considered for Gross pay, but do not feature in the Net Pay (for example, Imputed Income). This is the major difference between this parameter and the \$\$GROSS_PAY_CLASSIFICATION_NAME parameter.

Table 7–2 (Cont.) Classification Name Parameters

Parameter Name	Description
\$\$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 following the steps below.

To configure the classification name parameters:

1. In DAC, go to the Design view, and select your custom container from the drop-down list.
2. Click the Tasks tab, and select the task SDE_ORA_PayrollFact_Agg_Items.
3. Click the Parameters subtab, and specify the following parameters with an appropriate value.
 - \$\$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='Involuntary Deductions','Pre-Tax Deductions','Voluntary Deductions','Tax Deductions'
4. Save your changes.

To find out the possible Classification Names existing in your system, please run the following SQL against the OLTP Database:

```
SELECT CLASSIFICATION_NAME FROM PAY_ELEMENT_CLASSIFICATIONS
WHERE LEGISLATION_CODE LIKE 'US'
```

The Classification Name of Elements can be obtained using the following SQL:

Note: Enter the Element Type IDs of the elements for which you need the Classification Name within the quotes. If there are multiple elements, separate them using commas:

```
SELECT E.ELEMENT_TYPE_ID, E.ELEMENT_NAME, C.CLASSIFICATION_NAME
FROM PAY_ELEMENT_TYPES_F E, PAY_ELEMENT_CLASSIFICATIONS C
WHERE E.CLASSIFICATION_ID = C.CLASSIFICATION_ID AND E.ELEMENT_TYPE_ID IN ( )
```

7.2.3 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 7.2.3.1, "Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft HCM"](#)

- Section 7.2.3.2, "How to Configure the domainValues_Employee_Ethnic_Group_Code_psft.csv"
- Section 7.2.3.3, "How to Configure the domainValues_Employee_Sex_MF_psft.csv"
- Section 7.2.3.4, "How to Configure the domainValues_Employee_Veteran_Status_Code_psft.csv"
- Section 7.2.3.5, "How to Configure the domainValues_Employment_Category_psft.csv"
- Section 7.2.3.6, "How to Configure the domainValues_Employment_Exempt_Flag_psft.csv"
- Section 7.2.3.7, "How to Configure the domainValues_Employment_Full_Time_Flag_psft.csv"
- Section 7.2.3.8, "How to Configure the domainValues_Employment_Status_psft.csv"
- Section 7.2.3.9, "How to Configure the domainValues_Emp_EventType_Grp_Code_psft.csv"
- Section 7.2.3.10, "How to Configure the domainValues_Emp_EventType_SubG_Code_psft.csv"
- Section 7.2.3.11, "How to Configure the domainValues_HRPosition_Active_Pos_Flag_psft.csv"
- Section 7.2.3.12, "How to Configure the domainValues_HRPosition_Exempt_Flag_psft.csv"
- Section 7.2.3.13, "How to Configure the domainValues_Job_Eeo_Cat_Code_psft.csv"
- Section 7.2.3.14, "How to Configure the domainValues_Job_Flsa_Stat_Code_psft.csv"
- Section 7.2.3.15, "How to Configure the domainValues_Pay_Type_Grp_Code_psft.csv"
- Section 7.2.3.16, "How to Configure the domainValues_Emp_EventType_EventReason_Code_psft.csv"
- Section 7.2.3.17, "How to Configure the domainValues_Emp_EventType_EventClass_Code_psft.csv"
- Section 7.2.3.18, "How to Configure the file_pension_comp_flg.csv"
- Section 7.2.3.19, "How to Configure Compensation Flag and Pension Compensation Flag for Pay Type Dimension"
- Section 7.2.3.20, "How to Configure the Performance Review Type ETL Runtime Parameter"
- Section 7.2.3.21, "How to Configure Event Types using domainValues_EventTypes_psft.csv"
- Section 7.2.3.22, "How to Configure Department Trees for Oracle HR Analytics"

7.2.3.1 Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft HCM

The table below lists the CSV worksheet files and the domain values for PeopleSoft HR Analytics in \$pmsserver\LkpFiles folder located on Informatica Server box.

Table 7–3 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
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
domainValues_Employment_Exempt_Flg_psft.csv	W_EMPLOYMENT_D.W_EXEMPT_FLG	Lists the FLSA Statuses and corresponding domain values of 'Exempt Flag'.	SDE_PSFT_EmploymentDimension
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_Emp_EventType_Grp_Code_psft.csv	W_EVENT_TYPE_D.W_EVENT_CLASS,W_EVENT_TYPE_D.W_EVENT_GRP_CODE,W_EVENT_TYPE_D.W_EVENT_REASON_CODE	Lists the Event Types, Event Codes and Meanings and corresponding domain values of 'Event Group' and Event Reason'.	SDE_PSFT_EventTypeDimension_Stage1 SDE_PSFT_EventTypeDimension_Stage2

Table 7–3 (Cont.) Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft HCM

Domain Value	Table	Column Description	Session
domainValues_ Emp_EventType_ SubG_Code_ psft.csv	W_EVENT_ SUBG_CODE	Lists the Event Types, Event Codes and Meanings and corresponding domain values of 'Event Sub-Group'	SDE_PSFT_ EventDimension_Stage1 SDE_PSFT_ EventDimension_Stage2
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_ HRPosition_ Exempt_Flg_ psft.csv	W_HR_ POSITION_D.W_ EXEMPT_FLG	Lists the FLSA Statuses and corresponding domain values of 'Exempt Flag'.	SDE_PSFT_ HRPositionDimension
domainValues_ Job_Eeo_Cat_ Code_psft.csv	W_JOB_D.W_ EEO_JOB_CAT_ CODE	Lists the EEO Job Categories and corresponding domain values of 'EEO Job Category'.	SDE_PSFT_JobDimension
domainValues_ Job_Flsa_Stat_ Code_psft.csv	W_JOB_D.W_ FLSA_STAT_ CODE	Lists the FLSA Statuses and Corresponding domain values of the 'FLSA Status Code'	SDE_PSFT_JobDimension
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_ Pay_Type_Flg_ psft.csv	W_PAY_TYPE_ D.W_PAY_TYPE_ FLG	Lists the Costing Debit/Credit values and corresponding domain values of 'Pay type Flag'.	SDE_PSFT_PayTypeDimension_ Earnings, SDE_PSFT_ PayTypeDimension_ FederalTaxes, SDE_PSFT_ PayTypeDimension_ Deductions, SDE_PSFT_ PayTypeDimension_ Total, SDE_PSFT_ PayTypeDimension_ LocalTaxes1, SDE_PSFT_PayTypeDimension_ LocalTaxes2, SDE_PSFT_ PayTypeDimension_ StateTaxes1, SDE_PSFT_PayTypeDimension_ StateTaxes2

Table 7-3 (Cont.) Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft HCM

Domain Value	Table	Column Description	Session
domainValues_ Emp_EventType_ EventClass_ Code_psft.csv	NA	Lists the event type class codes.	SDE_PSFT_ EventTypeDimension_Stage2
domainValues_ Emp_EventType_ EventReason_ Code_psft.csv	NA	Lists the event type reason codes.	SDE_PSFT_ EventTypeDimension_Stage2

7.2.3.2 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 ETHNIC_GRP_CD NOT LIKE
ORDER BY 1
```

2. Open the domainValues_Employee_Ethnic_Group_Code_psft.csv file text editor in the \$pmsserver\lkpfiles folder.
3. Copy the FIELDVALUE to the ETHNIC_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each Ethnic Code to one domain value.
5. Save and close the file.

7.2.3.3 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. Open the domainValues_Employee_Sex_MF_psft.csv file located in \$pmsserver\lkpfiles folder on Informatica Server host using text editor.
3. Copy the FIELDVALUE column to the SEX column in the file. The data must be copied starting from the 6th line.
4. Map each Sex Code to one domain value.
5. Save and close the file.

7.2.3.4 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. Open the domainValues_Employee_Veteran_Status_Code_psft.csv file located in \$pmserver\lkpfiles folder on Informatica Server host using text editor.
3. Copy the FIELDVALUE column to the VETERAN_STATUS_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each Veteran Status Code to one domain value.
5. Save and close the file.

7.2.3.5 How to Configure the domainValues_Employment_Category_psft.csv

1. Identify the User Person Types in your PeopleSoft source system by using the following SQL:

```
SELECT DISTINCT LTRIM (RTRIM (EMPL_CLASS)) EMPL_CLASS
FROM PS_JOB
ORDER BY 1
```

2. Open the domainValues_Employment_Category_psft.csv file located in \$pmserver\lkpfiles directory placed on Informatica Server host using text editor.
3. Copy the EMPL_CLASS values to the EMP_CAT_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each User Person Type to one domain value.
5. Save and close the file.

7.2.3.6 How to Configure the domainValues_Employment_Exempt_Flg_psft.csv

1. Identify the FLSA 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 = 'FLSA_STATUS'
ORDER BY 1
```

2. Open the domainValues_Employment_Exempt_Flg_psft.csv file located in \$pmserver\lkpfiles directory placed on Informatica Server host using text editor.

3. Copy the FIELDVALUE to the FLSA_STATUS column in the file. The data must be copied starting from the 6th line.
4. Map each FLSA_STATUS to one domain value.
5. Save and close the file.

7.2.3.7 How to Configure the domainValues_Employment_Full_Time_Flg_psft.csv

1. Identify the Employment Categories 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 = 'FULL_PART_TIME'
ORDER BY 1
```

2. Open the domainValues_Employment_Full_Time_Flg_psft.csv file located in \$pmsserver\lcpfiles directory placed on Informatica Server host using text editor.
3. Copy the FIELDVALUE to the FULL_PART_TIME column in the file. The data must be copied starting from the 6th line.
4. Map each EMPLOYMENT_CATEGORY to one domain value.
5. Save and close the file.

7.2.3.8 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. Open the domainValues_Employment_Full_Time_Flg_psft.csv file located in \$pmsserver\lcpfiles directory placed on Informatica Server host using text editor.
3. Copy the FIELDVALUE to the FULL_PART_TIME column in the file. The data must be copied starting from the 6th line.
4. Map each EMPLOYMENT_CATEGORY to one domain value.
5. Save and close the file.

7.2.3.9 How to Configure the domainValues_Emp_EventType_Grp_Code_psft.csv

1. Identify the Event Types Group 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 = 'ACTION'
ORDER BY 1
```

2. Open the domainValues_EMP_EventType_Grp_Code_psft.csv file editor in \$pmsserver\lkpfiles folder using text.
3. Copy the FIELDVALUE to the LOOKUP_TYPE_CODE. The data must be copied starting from the 6th line. Use commas to separate the entries.
4. Map each Event Type (LOOKUP_CODE) to one domain value for each domain column.
5. Save and close the file.

7.2.3.10 How to Configure the domainValues_Emp_EventType_SubG_Code_psft.csv

1. Identify the Event Types Subgroup Codes in your PeopleSoft source system by using the following SQL:

MSSQL Server Syntax:

```
SELECT ACTION + ':' + ACTION_REASON SUBG_CODE, DESCR FROM PS_
ACTN_REASON_TBL ORDER BY 1
```

Oracle Syntax:

```
SELECT ACTION || ':' || ACTION_REASON SUBG_CODE, DESCR FROM PS_
ACTN_REASON_TBL ORDER BY 1
```

2. Open the domainValues_EMP_EventType_SubG_Code_psft.csv file editor in \$pmsserver\lkpfiles folder using text.
3. Copy the SUBG_CODE to the EVENT_REASON. The data must be copied starting from the 6th line. Use commas to separate the entries.
4. Map each Event Type (EVENT_REASON) to one domain value for each domain column.
5. Save and close the file.

7.2.3.11 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. Open the domainValues_HRPosition_Active_Pos_Flg_psft.csv file using text editor in \$pmsserver\lkpfiles directory.
3. Copy the FIELDVALUE to the EFF_STATUS column in the file. The data must be copied starting from the 6th line.

4. Map each position EFF_STATUS to one domain value.
5. Save and close the file.

7.2.3.12 How to Configure the domainValues_HRPosition_Exempt_Flg_psft.csv

1. Identify the FLSA 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 = 'FLSA_STATUS'
ORDER BY 1
```

2. Open the domainValues_HRPosition_Exempt_Flg_psft.csv file using text editor in \$pmsserver\lkpfiles directory.
3. Copy the FIELDVALUE to the FLSA_STATUS column in the file. The data must be copied starting from the 6th line.
4. Map each FLSA_STATUS to one domain value.
5. Save and close the file.

7.2.3.13 How to Configure the domainValues_Job_Eeo_Cat_Code_psft.csv

1. Identify the EEO Job Categories in your PeopleSoft source system by using the following SQL:

```
SELECT 'EEO1CODE' CODE_TYPE,
LTRIM (RTRIM (EEO1CODE))
FROM PS_JOBCODE_TBL
UNION
SELECT 'EEO4CODE' CODE_TYPE,
LTRIM (RTRIM (EEO4CODE))
FROM PS_JOBCODE_TBL
UNION
SELECT 'EEO5CODE' CODE_TYPE,
LTRIM (RTRIM (EEO5CODE))
FROM PS_JOBCODE_TBL
UNION
SELECT DISTINCT 'EEO6CODE' CODE_TYPE,
LTRIM (RTRIM (EEO6CODE))
FROM PS_JOBCODE_TBL
```

2. Open the domainValues_Job_Eeo_Cat_Code_psft.csv file in \$pmsserver\lkpfiles directory using text editor.
3. Copy the CODE_TYPE to the EEO_JOB_CAT_CODE_TYPE column in the file. The data must be copied starting from the 6th line.

4. Map each pair of EEO_JOB_CAT_CODE_TYPE and EEO_JOB_CAT_CODE_VAL to one domain value.
5. Save and close the file.

7.2.3.14 How to Configure the domainValues_Job_Flsa_Stat_Code_psft.csv

1. Identify the FLSA 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 = 'FLSA_STATUS'
ORDER BY 1
```

2. Open the domainValues_Job_Flsa_Stat_Code_psft.csv file in \$pmserver\lkpfiles directory using text editor.
3. Copy the JOB_INFORMATION3 to the FLSA_STAT_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each FLSA_STAT_CODE to one domain value.
5. Save and close the file.

7.2.3.15 How to Configure the domainValues_Pay_Type_Grp_Code_psft.csv

The table below shows pay type sources and their associated group codes.

Table 7-4 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

1. To configure the domainValues_Pay_Type_Grp_Code_psft.csv
2. Open the domainValues_Pay_Type_Grp_Code_psft.csv file using text editor in the \$pmserver\lkpfiles directory.
3. Copy the Classification Name and Element Name to the PAYTYPE_SOURCE. The data must be edited starting from the 6th line. Use commas to separate the entries.
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.

7.2.3.16 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 \$pmsserver\lkpfiles folder.
3. Copy EVENT_REASON to the EVENT_REASON column in the file. The data must be copied starting from the 6th line.
4. Map each EVENT_REASON to one domain value.
5. Save and close the file.

7.2.3.17 How to Configure the domainValues_Emp_EventType_EventClass_Code_psft.csv

1. The shipped domain value for PeopleSoft ETL source system is set:
 - EVENT_CLASS: W_EVENT_CLASS
 - DEFAULT: HR_EVENTS
2. Using a text editor, open the domainValues_Emp_EventType_EventClass_Code_psft.csv file, located in the \$pmsserver\lkpfiles folder.
3. Modify EVENT_CLASS column in the file. The data must be copied starting from the 6th line.
4. Map each EVENT_CLASS to one domain value.
5. Save and close the file.

7.2.3.18 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 \$pmsserver\lkpfiles folder.
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
```

7.2.3.19 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 using in a text editor.
2. Enter the COMP_FLG and PENSION_COMP_FLG value against the correct Earnings Code.
3. Save the file.

7.2.3.20 How to Configure the Performance Review Type ETL Runtime Parameter

As a part of the ETL configuration runtime parameter, value for Performance Review Type must be specified in DAC. This parameter will be used to extract performance ranking information and related metrics for the Employee Snapshot Fact table.

To configure the Performance Review runtime ETL parameter

1. In DAC, go to the Design view, and select the appropriate custom container.
2. Click the Tasks tab, and select the task 'SDE_PSFT_EmployeeDailySnapshotFact_Reviews_Accruals'.
3. Click the Parameters subtab.
4. Query for the parameter \$\$PERFORMANCE_REVIEW_TYPE, and enter the appropriate value in the Value field.

You can use the SQL statement below to find out the Performance Review Types for review process in the PeopleSoft source system:

```
SELECT EP_REVIEW_TYPE, EP_REVW_DESCR FROM PS_EP_REVW_TYP_TBL
```

7.2.3.21 How to Configure Event Types using domainValues_EventTypes_psft.csv

To configure Event Types using the domainValues_EventTypes_psft.csv:

1. Identify the Actions and Action reasons in your PeopleSoft source system using the following SQL:

```
SELECT A.ACTION,A.ACTION_REASON
FROM PS_ACTN_REASON_TBL A, (SELECT ACTION,ACTION_REASON, MAX(EFFDT) EFFDT
FROM PS_ACTN_REASON_TBL
GROUP BY ACTION,ACTION_REASON) B
WHERE A.ACTION=B.ACTION AND
A.ACTION_REASON = B.ACTION_REASON AND
A.EFFDT = B.EFFDT
```

2. Using a text editor, open the file domainValues_EventTypes_psft.csv, located in \$pmserver\lkpfiles.
3. Copy the Action and Action Reason from the SQL result starting from the sixth line.

Use commas to separate the entries.

4. Map each combination of Action and Action reason to one domain value for each of the domain value columns (that is, W_EVENT_GRP_CODE, W_EVENT_SUBG_CODE, W_EVENT_REASON_CODE and E_EVENT_CLASS).
5. Save and close the file.

7.2.3.22 How to Configure Department Trees for Oracle HR Analytics

Oracle HR Analytics 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

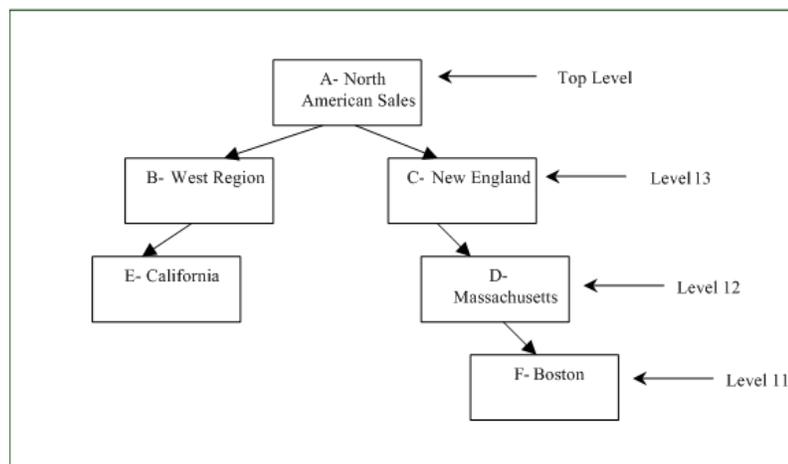
PeopleSoft departments and the associated department trees are supported in Oracle HR Analytics as the organization dimension (W_INT_ORG_D) and the flattened organizational hierarchy structure (W_INT_ORG_DH).

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 7-1 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 7-5 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 7-6 How PS_DEPT_TBL Populates W_INT_ORG_DH

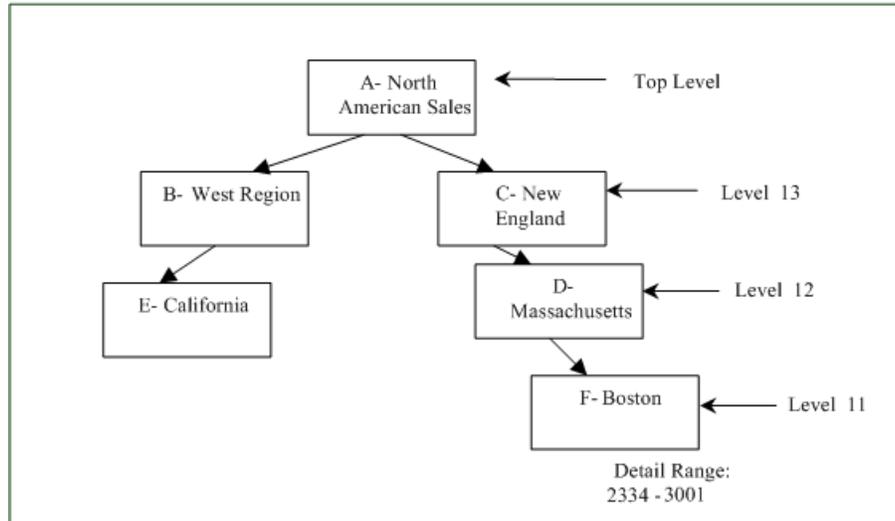
ORG_WID	ORG_HIER11_NUM	ORG_HIER11_NAME	ORG_HIER12_NUM	ORG_HIER12_NAME	ORG_HIER13_NUM	ORG_HIER13_NAME	ORG_TOP_NUM	ORG_TOP_NAME	HIER_ARCH_Y_NAME	W_HIER_ARCH_Y_CLAS S	FIXED_HIER_LEVEL	HR_ORG_FLAG
1	A	North American Sales	A	North American Sales	A	North American Sales	A	North American Sales	Share ~NA Sales	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
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 7-2 How a Summer Tree Is Flattened



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

Table 7-7 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 7-8 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	10	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_DEH.CURRENT_VER_HIER_FLG
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
Employee Organization Hierarchy 13	W_INT_ORG_DH.HIER13_NUM
Employee Organization Hierarchy 14	W_INT_ORG_DH.HIER14_NUM

The table below 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_DHW

The following temporary tables are used to handle the extraction and load of the trees:

- Sequence 1: W_PSFT_INT_ORG_DEPT_DH_TMP

- Sequence 2: W_PSFT_INT_ORG_TREE_TMP
- Sequence 3: W_PSFT_INT_ORG_VERT_DH_TMP
- Sequence 4: W_PSFT_INT_ORG_DTLRGE_DH_TMP
- Sequence 5: W_PSFT_INT_ORG_FLAT_DH_TMP

Department Tree Flattening Process ETL Parameters

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 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 tree SetID and Tree Name in DAC

1. In DAC, go to the Design view.
2. Click the Tasks tab, and select the SDE_PSFT_Stage_InternalOrganizationDimension_Hierarchy_Extract.
3. Click the Parameters subtab.
4. For the parameter \$\$TREE_SETID_NAME_LIST, enter the appropriate values for SETID and TREE_NAME.

Note: The \$\$TREE_SETID_NAME_LIST parameter is in the format '`<setid>~<tree_name>`'.

A single quotation mark is required. If you have multiple trees, separate them using a comma, for example, 'SHARE~DEPT1', 'US~DEPT1'.

7.2.4 Configuration Steps for Oracle HR Analytics for Universal

This section contains configuration steps required before you do a full data load that apply to Universal.

Not applicable to Oracle BI Applications release 7.9.5.1.

7.2.5 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle HR Analytics.

- [Section 7.2.5.1, "Configuration Steps for Oracle HR Analytics for All Source Systems"](#)
- [Section 7.2.5.2, "Configuration Steps for Oracle HR Analytics for Oracle EBS"](#)
- [Section 7.2.5.3, "Configuration Steps for Oracle HR Analytics for PeopleSoft"](#)
- [Section 7.2.5.4, "Configuration Steps for Oracle HR Analytics for Universal"](#)

7.2.5.1 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 7.2.5.1.1, "How to Aggregate the Payroll Table for Oracle HR Analytics"](#)
- [Section 7.2.5.1.2, "About Configuring the Employee Snapshot Fact tables"](#)

- [Section 7.2.5.1.3, "How to Configure the Rolling Period for Storing Employee Daily Snapshot Data"](#)
- [Section 7.2.5.1.4, "How to Configure the Granularity of the Employee Monthly Snapshot Data"](#)

7.2.5.1.1 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 the DAC (Data Administration Console).

To load the Payroll aggregate table (W_PAYROLL_A)

1. In DAC, go to the Design view.
2. Click the Tasks tab, and select the Task PLP_PayrollAggregate_Load.
3. Click 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.

7.2.5.1.2 About Configuring the Employee Snapshot Fact tables The Employee Daily Snapshot Fact table stores employee-related metrics for all employees against their primary job function. You can configure the two fact tables that store Employee

Snapshot data, namely W_EMPLOYEE_DAILY_SNP_F and W_EMPLOYEE_MONTHLY_SNP_F.

You can configure the rolling period and the granularity for the Employee Snapshot data process.

Rolling Period

The Employee Daily Snapshot Fact table is loaded everyday, the date being identified by the snapshot date column. For example, if your organization has 1000 employees, this table would load 1000 records everyday. If an employee's contract is terminated on a particular day, the following day's snapshot data would not have that record (that is, 999 records would be uploaded). Continuing this example, if you run ETL for 60 days, you will have 60 snapshot data sets. However, the number of days worth of snapshot data you want to store can be configured. In fact, you can configure to store only for a 'rolling period of' some days, example 30. This is the out of box setting. If you start your ETL on January 1 and keep running for 40 days, you will have snapshot data sets from Jan 20 until Feb 10. If you configure to store only seven rolling days, you will have snapshot data sets from Feb 3 until Feb 10, and so on.

The ETL mapping responsible for maintaining the rolling period choice you have indicated is 'PLP_EmployeeDailySnapshot_Trim', which uses two Informatica parameters:

- **\$\$KEEP_PERIOD**
Indicates your choice of the period type based on which you will be providing the rolling periods. The Valid values for this parameter are 'DAY', 'WEEK', 'MONTH', 'QUARTER', and 'YEAR' (values should include single quotes as indicated).
- **\$\$NUM_OF_PERIOD**
Indicates your choice of how many days/weeks/months/years (set by \$\$KEEP_PERIOD) data you want to preserve in a rolling fashion. The valid value for this parameter is any integer.

For example, if you choose to preserve a rolling period of 45 Days, you would set \$\$KEEP_PERIOD='DAY' and \$\$NUM_OF_PERIOD=45.

Granularity

The Employee Monthly Snapshot Fact table stores the end-of-month snapshot data for all employees in your organization. Out of the box, this table stores end-of-month snapshot data, but this is configurable to store end-of-week, end-of-quarter or even end-of-year data. If you configure it to store end-of-month, as it is installed out-of-the-box, then you will have one snapshot data set for a given month, the set being refreshed everyday.

The out-of-box setting for the W_EMPLOYEE_MONTHLY_SNP_F table is to store end-of-month snapshot data set, on set for each calendar month. However, this can be configured. If you like to have this table store end-of-week, or end-of-quarter or even end-of-year data, your would need to configure the correct Informatica parameter definition in the DAC. This parameter definition is as follows:

- **\$\$GRAIN**: Indicates the chosen granularity of the snapshot fact table that stores data at a interval higher than the base snapshot fact table. Valid values for this parameter are 'WEEK', 'MONTH', 'QUARTER', 'YEAR' (values should include single quotes as indicated).

Notes:

- Do not set the value of \$\$GRAIN to 'DAY'.

- In the reporting logical metadata (RPD), the table W_EMPLOYEE_MONTHLY_SNP_F is joined with the Calendar Month dimension table - both in the physical layer as well as the logical layer. If you are change the \$\$GRAIN out-of-box value from 'MONTH' to anything else (like 'QUARTER'), you would need to modify the join in the RPD as well to the appropriate date dimension.
- The physical name of the table can still remain as W_EMPLOYEE_MONTHLY_SNP_F. You can change the logical name (or alias) appropriately. For example, if your \$\$GRAIN is 'QUARTER', you might change it to quarterly snapshot.

To configure the Employee Snapshot Fact tables, perform the following tasks:

- [Section 7.2.5.1.3, "How to Configure the Rolling Period for Storing Employee Daily Snapshot Data"](#)
- [Section 7.2.5.1.4, "How to Configure the Granularity of the Employee Monthly Snapshot Data"](#)

7.2.5.1.3 How to Configure the Rolling Period for Storing Employee Daily Snapshot Data For more information about the rolling period for the Employee Snapshot data, see [Section 7.2.5.1.2, "About Configuring the Employee Snapshot Fact tables."](#)

To configure the rolling period for storing Employee Daily Snapshot data

1. Log in to the DAC repository, go to the Design view, and select your custom container from the drop-down list.
2. Click on Tasks in the right pane and query for the task name = PLP_EmployeeDailySnapshot_Trim.
3. In the pane below, click on the Parameters subtab.
4. Set the values of the two parameters \$\$KEEP_PERIOD and \$\$NUM_OF_PERIOD.
Note: Although the \$\$NUM_OF_PERIOD requires an integer value, do not change the data type from 'text'.

7.2.5.1.4 How to Configure the Granularity of the Employee Monthly Snapshot Data For more information about granularity in the Employee Snapshot data, see [Section 7.2.5.1.2, "About Configuring the Employee Snapshot Fact tables"](#).

To configure the granularity of the Employee Snapshot data

1. In the DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Click on the Tasks tab, and query for the task PLP_EmployeeMonthlySnapshot.
3. Click the Parameters subtab.
4. Query for the \$\$GRAIN parameter, and enter an appropriate value in the Value field.

7.2.5.2 Configuration Steps for Oracle HR Analytics for Oracle EBS

This section contains configuration steps that apply to Oracle EBS.

Not applicable to Oracle BI Applications release 7.9.5.1.

7.2.5.3 Configuration Steps for Oracle HR Analytics for PeopleSoft

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

- [Section 7.2.5.3.1, "How to Tune Performance for the Payroll Table for PeopleSoft HCM HR Analytics"](#)

7.2.5.3.1 How to Tune Performance for the Payroll Table for PeopleSoft HCM HR Analytics The Payroll mappings designed 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 requirement, there are two ways you can achieve better overall performance in the Payroll Analytics. These techniques are as follows:

- Inactivate Adjustments and Reversals, if not required, for a given pay period
- Defer calculating Adjustments and Reversals for some time, until needed.

To inactivate 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 inactivate those tasks in the DAC.

1. In the DAC, go to the Design view, and select the appropriate custom container.
2. Select the Tasks tab.
3. Deselect 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. Inactivate 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.

7.2.5.4 Configuration Steps for Oracle HR Analytics for Universal

Not applicable to Oracle BI Applications release 7.9.5.1.

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, "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"](#)
- [Section 8.2.2, "Configuration Steps for Controlling Your Data Set"](#)

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 to 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 OracleBI\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.

8.2.2 Configuration Steps for Controlling Your Data Set

This sections contains additional configuration steps for Oracle Sales Analytics. It contains the following topics:

- [Section 8.2.2.1, "How to Configure the Snapshot Frequency in the Pipeline Fact"](#)

8.2.2.1 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 your custom container from the drop-down list.
2. Click the Tasks tab, and query for the SIL_PipelineFact task.
3. Click the Parameters subtab, and create a parameter called \$\$GRAIN, and enter an appropriate parameter value.

8.3 Configuring Usage Accelerator

This section explains how to configure Usage Accelerator. It contains the following topics:

- [Section 8.3.1, "What is Usage Accelerator?"](#)
- [Section 8.3.2, "About the Usage Accelerator Execution Plan"](#)
- [Section 8.3.3, "Using Usage Accelerator with Siebel 7.5.3 Applications"](#)

8.3.1 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.

8.3.1.1 Configuring Usage Accelerator

This section provides an overview to configuring Usage Accelerator.

8.3.1.1.1 Tracking of Targeted Users In order to optimize the storage space required for the application data, Usage Accelerator enables you to choose which users you want to measure. Usage Accelerator determines the users that you want to measure through the use of a new responsibility called 'Usage Accelerator – Tracking'. This new responsibility will need to be associated to each user in the operational Siebel application.

8.3.1.1.2 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

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 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	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
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	Opportunity Updates	Opportunity Updates
Usage Accelerator – FINS Sales Manager	No Position Based Security	User Adoption	Active Users

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	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
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	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

8.3.1.2 About Deployment Date

Usage Accelerator module needs to 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. Please note that if you set the Deployment Date earlier than the current calendar date, metrics that rely on the Last Updated Date (e.g. # of Updated Records, # of Logins, # of Sync's, etc.) of a record may be inaccurate. For this reason we recommend setting the Deployment Date to the current Calendar Date.

8.3.1.3 About Metric Definitions and Variables

This section explains Usage Accelerator metric definitions and variables.

8.3.1.3.1 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 to 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_NUMSYNC.
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

8.3.1.3.2 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

8.3.1.3.3 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
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

8.3.1.3.4 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.

Table 8–6 (Cont.) Coverage Metrics

Metric Category	Description	Variables
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 (FIN\$)	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.

8.3.2 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*'.

8.3.3 Using Usage Accelerator with Siebel 7.5.3 Applications

This section explains how to install and use Usage Accelerator with Siebel 7.5.3 Applications.

Note: This section only applies if you are planning to use Usage Accelerator with Siebel CRM 7.5.3 applications.

8.3.3.1 Usage Accelerator Application Integration with Siebel 7.5.3 CRM Application

The Usage Accelerator installation process consists of installing Usage Accelerator and importing seed data into the transactional database. You install Usage Accelerator using the standard Oracle Business Intelligence installer. The installation options that are available to you depend on the license key you enter.

To integrate Usage Accelerator

1. Import repository objects to enable embedded Analytics.

Use Siebel tools to imports 'SRW_753_UA_Integration_objects.sif' file into tools repository.

2. Lock the following projects:
 - Analytics
 - ERM
 - Siebel Field Service
 - Siebel Marketing Enterprise
 - Siebel Channel

- Siebel Sales Enterprise
- Siebel Service Enterprise
- Siebel Universal Agent

Note: Do not apply the schema in the transactional database after importing these objects in tools repository. You might see some tables created in this step. These tables would have been applied in the transactional database.

3. Recompile the srf file, as follows:
 - a. In Windows Explorer, navigate to \tools\objects\enu, and create a copy of siebel.srf under a different name.
 - b. In Oracle's Siebel Tools, navigate to Tools, then Compile Projects, and select the Locked Projects radio button.
 - c. Under the Siebel Repository, select the new SRF file you created in Step a, and click Compile.

To see the Usage Accelerator screen in your application, the new SRF file must replace the siebel.srf file you were using.

4. Confirm the Analytics Server name to run Embedded Analytics.
 - a. Log in to the Siebel application as the administrator.
 - b. Navigate to Site Map, then Administration - Integration, then WI Symbolic URL List.
 - c. In the Administration - Integration screen, from the visibility filter, select Host Administration.
 - d. In the Host Administration list, locate the record Name=[AnalyticsServerName].
 - e. Change [AnalyticsServerName] to the name of the machine hosting your Analytics Server.
5. Import language-independent Siebel seed data, as follows:
 - a. Copy the appropriate seed data files:

Table 8-7 Seed data files

File name	From	To
seed_753_UA.dat	OracleBI\dwrep	Local machine
seed_753_<xxx>_UA.inp where xxx corresponds to the database platform you are using	OracleBI\dwrep	Local machine

- b. Open a command line, and go to the same directory where you copied the .dat and .inp files.
 - c. Replace each of the connection parameters with the values appropriate to your database environment, and then run the appropriate import command from the following table:

Table 8–8 Import commands

Database Platform	Run Import Command
Oracle	<code>\$SIEBELSERVERROOT\bin\dataimp /u \$UserName /p Oracle \$Password /c "\$ODBCDataSource" /d \$DatabaseOwner /f seed_753_UA.dat /w y /q 100 /h Log /x f /i seed_753_ora_UA.inp /lseed_753_ora_UA.log</code>
DB2UDB	<code>\$SIEBELSERVERROOT\bin\dataimp /u \$UserName /p \$Password /c "\$ODBCDataSource" /d \$DatabaseOwner /f seed_753_UA.dat /w y /q 100 /h Log /x f /i seed_753_db2_UA.inp /lseed_753_db2_UA.log</code>
MSSQL	<code>\$SIEBELSERVERROOT\bin\dataimp /u \$UserName /p \$Password /c "\$ODBCDataSource" /d \$DatabaseOwner /f seed_753_UA.dat /w y /q 100 /h Log /x f /i seed_753_mssql_UA.inp /lseed_753_mssql_UA.log</code>

- Associate the new Usage Accelerator responsibilities with the users whom you want to see the Usage Accelerator screen.

The Usage Accelerator responsibilities are as follows:

- Usage Accelerator Tracking
- Usage Accelerator - Sales Rep
- Usage Accelerator - Sales Manager
- Usage Accelerator - Sales Executive
- Usage Accelerator – Administrator

Note: The first responsibility has to be associated with any transaction application user whom you want to be tracked in Usage Accelerator reports.

8.3.3.2 Usage Accelerator-Analytics Adapter Files

The contents of each of the files included in the OracleBI\dwrep folder are shown in the table below.

Table 8–9 Usage Accelerator Analytics Adapter Files

File	Description
seed_753_UA.dat	Siebel Seed Data file containing language- independent seed data records for the following tables: S_WI_SYMURL, S_WE_SYMURL_ARG, S_RESP, S_APP_VIEW, S_APP_VIEW_RESP.
seed_753_db2_UA.inp seed_753_mssql_UA.inp seed_753_ora_UA.inp	Import file used to control the import of the seed_753_UA.dat file to a particular database platform.

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 `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.

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 \$PMRoot\SrcFiles folder.
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 make sure 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 needs to 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 make sure 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 `\OracleBI\dwrep\Informatica\SrcFiles` folder.
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_bnmrk_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 below:
 - 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 the section entitled "Customizing the Oracle Business Analytics Warehouse," in *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.
- 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 needs to be leveled in the repository (RPD) at the grand-total level of that dimension. This dimension ID also needs to 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_bchmrk_tgt.csv` file, located in the `\OracleBI\dwrep\Informatica\SrcFiles` folder.
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 needs to be created for each fact table and a mapping needs to be created that can load this table.

For example in order to delete records from W_ACD_EVENT_F, a new staging table needs to 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 needs to 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. The table below 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. The following table 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 Configuration for Post Load Processing 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 , click the Subject Areas tab and query for the 'Service – Rep Activity' subject area.
2. With this subject area selected, click on 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 \OracleBI\dwrep\Informatica\SrcFiles folder.
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, in the Design view, select the Universal container
2. Click the Subject Areas tab, and query for the 'Service - Contact Center Performance' subject area
3. With this subject area selected, click the Configuration Tags tab.
4. Deselect 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 \OracleBI\dwrep\Informatica\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, in the Design view, select the Universal container.
2. Click the Tasks tab, and query for the SDE_Universal task.
3. In the lower pane, click the Parameters subtab.
4. Change the value of the parameter \$\$READ_INTEGRATION_ID_FROM_FILE to FALSE.

The following table 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_ContactCenterPerformance Fact

Table 9–8 (Cont.) Expressions and Task Names

Table Name	INTEGRATION_ID Expression	Task Name
W_CNTCT_CNTR_ BNCHMRK_TGT_F	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	SDE_Universal_ ContactCenterBenchmark TargetFact
W_CNTCT_CNTR_PERF_ REP_HOUR_A	CNTCT_CNTR_LOC_ ID ' ~' CNTCT_REP_ ID ' ~' CNTCT_DT	SDE_Universal_ ContactCenterPerformance RepHourAggregate
W_REP_ACTIVITY_F	CNTCT_REP_ ID ' ~' ACTIVITY_TYPE_ ID ' ~' ACTIVITY_START_ DT ' ~' ACTIVITY_ START_LDT	SDE_Universal_ RepActivityFact
W_REP_ACTIVITY_HOUR_ A	CNTCT_REP_ ID ' ~' ACTIVITY_TYPE_ ID ' ~' ACTIVITY_DT	SDE_Universal_ RepActivityHourAggregate

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-.....-MENU n where MENU m 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 given below.

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 analytic 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 on 'Format Column' to open the Column Properties dialog.
3. Go to Data format.
4. Check 'Override Default data format'.
5. Select Treat Number as 'Custom' and give Custom Numeric Format as "0#:##:##".

9.2.20 Configuration Steps for Controlling Your Data Set

Not applicable to Oracle BI Applications release 7.9.5.1.

Part III

Configuring the Oracle BI Repository

Part III contains instructions for configuring the Oracle BI Repository. It includes the following section: [Chapter 10, "Configuring the Oracle BI Repository."](#)

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:

- [Section 10.1, "Modifying the Oracle BI Repository for Siebel Applications version 6.3 and version 7.5"](#)
- [Section 10.2, "Configuring the Oracle BI Repository Connections"](#)
- [Section 10.3, "Setting up Date-Specific Metrics"](#)
- [Section 10.4, "Setting Up Additional Time Series Metrics for Oracle Business Analytics Warehouse"](#)
- [Section 10.5, "Setting Up Additional Dimension Tables for Oracle Business Analytics Warehouse"](#)
- [Section 10.6, "About the Period Ago Keys for Oracle Business Analytics Warehouse"](#)
- [Section 10.7, "About Oracle BI Time Repository Variables"](#)
- [Section 10.8, "About User Authentication"](#)
- [Section 10.9, "About the Security or Visibility Configuration"](#)
- [Section 10.10, "About the Group Variable"](#)
- [Section 10.11, "About Configuring Usage Tracking for Oracle Business Analytics Warehouse"](#)
- [Section 10.12, "About the Incremental Deployment of the Oracle BI Applications Repository"](#)

Note: When you access the OracleBIAnalyticsApps.rpd repository in Oracle BI Administration Tool, you must log on with the username and password Administrator\SADMIN.

Note: In this section, the Oracle Business Analytics Warehouse is also referred to as the Oracle Data Warehouse. These two terms refer to the same data warehouse.

10.1 Modifying the Oracle BI Repository for Siebel Applications version 6.3 and version 7.5

Before you can enable Siebel Applications version 6.3 and version 7.5, you must modify the Oracle BI Repository.

To modify the Oracle BI Repository for Siebel applications version 6.3 and version 7.5

1. Open the file RepositoryDifferences75_63.xls in the \OracleBI\dwrep folder.
2. Using the Oracle BI Administration Tool, publish the information in the following tabs:
 - ITBlocks
 - LogicalCols
 - FinCols

10.2 Configuring the Oracle BI Repository Connections

Oracle BI Repository uses four databases defined in the Physical layer, as follows:

- Oracle Business Analytics Warehouse
- Oracle EBS OLTP
- PeopleSoft OLTP
- Siebel OLTP

You need to configure the Oracle BI Repository variables and connection pools to connect to your databases, as follows:

- For information about configuring connection pools, see [Section 10.2.2, "How to Configure the Oracle BI Repository Connections"](#).
- For information about configuring repository variables, see [Section 10.2.3, "How to Configure Oracle BI Repository Variables"](#).

10.2.1 About the Predefined Connection Pools in the Oracle Business Analytics Warehouse

The Oracle Business Analytics Warehouse physical database has two predefined connection pools:

- **Oracle Data Warehouse Connection Pool.** The Oracle Business Analytics Warehouse Connection Pool is the main connection pool in the Oracle BI Repository. You need to configure this connection pool to connect to your physical data warehouse. The connection is used by the session initialization blocks. You can use this connection pool to set up a dynamic data source name.
- **Oracle Data Warehouse Repository Initblocks Connection Pool.** You need to configure the Oracle Data Warehouse Repository Initblocks Connection Pool to connect to the your physical data warehouse. The connection is used by the repository level initialization blocks. Repository level initialization blocks cannot be configured to use the dynamic data source name.

You can also set up dynamic data source names, which allow an Administrator to set one instance of Analytics server to connect to different data warehouses depending on the user. For more information about how to set up dynamic data

source names, see [Section 10.2.1.1, "How to Configure Dynamic Data Source Names"](#).

You also need to configure the following Static variables:

- **OLAP_DSN.** The value of the OLAP_DSN static variable is set to the data source name for the warehouse database.
- **OLAP_USER.** The value of the OLAP_USER static variable is set to the database user name for the warehouse database.
- **OLAPTBO.** The value of the OLAPTBO static variable is set to the database table owner for the data warehouse database.

The PeopleSoft OLTP, Siebel OLTP, and Oracle EBS OLTP databases have two predefined connection pools each. The actual databases in the RPD will depend on the modules licensed by the customer. The connection pools and their functions within each database are listed below.

- Oracle EBS OLTP:
 - **Oracle EBS OLTP DBAuth Connection Pool.** The Oracle EBS OLTP DBAuth Connection Pool is used if database authentication is required.
 - **Oracle EBS OLTP Connection Pool.** The Oracle EBS OLTP Connection Pool is used to connect to the Oracle EBS OLTP system.

You also need to configure the following Static variables:

- ORA_EBS_OLTP_DSN. The value of the ORA_EBS_OLTP_DSN static variable is set to the data source name for the Oracle EBS OLTP database.
- ORA_EBS_OLTP_USER. The value of the ORA_EBS_OLTP_USER static variable is set to the database user name for the Oracle EBS OLTP database.
- Siebel OLTP
 - **Siebel OLTP DBAuth Connection Pool.** The Siebel OLTP DBAuth Connection Pool is used if database authentication is required.
 - **Siebel OLTP Connection Pool.** The Siebel OLTP Connection Pool is used to connect to the Siebel OLTP system.

You also need to configure the following Static variables:

- OLTP_DSN. The value of the OLTP_DSN static variable is set to the data source name for the Siebel OLTP database.
- OLTP_USER. The value of the OLTP_USER static variable is set to the database user name for the Siebel OLTP database.
- PeopleSoft OLTP
 - **PeopleSoft OLTP DBAuth Connection Pool.** The PeopleSoft OLTP DBAuth Connection Pool is used if database authentication is required.
 - **PeopleSoft OLTP Connection Pool.** The PeopleSoft OLTP Connection Pool is used to connect to the PeopleSoft OLTP system.

You also need to configure the following Static variables:

- OLTP_DSN. The value of the OLTP_DSN static variable is set to the data source name for the PeopleSoft OLTP database.
- OLTP_USER. The value of the OLTP_USER static variable is set to the database user name for the PeopleSoft OLTP database.

10.2.1.1 How to Configure Dynamic Data Source Names

This section explains how to create and configure dynamic data source names.

Dynamic data source names allow the Administrator to set one instance of Analytics server to connect to different data warehouses depending on the user. For this you need to have your user authentication based on an external system (like LDAP), and add the following to your repository:

1. Create new session variables: Session_OLAP_DSN and Session_OLAP_USER
2. Create a Session Init Block which uses 'Oracle Data Warehouse Repository Initblocks Connection Pool' to populate these session variables based on the user login.
3. Add this Initialization Block to the Execution Precedence list of the Authorization Initialization block.
4. Modify the values of Data Source Name and User Name fields in 'Oracle Data Warehouse Connection Pool' to be VALUEOF(Session_OLAP_DSN) and VALUEOF(Session_OLAP_USER) respectively.

For the Password field, you should have the user password the same as the user name.

5. Update the field password with the same value as of User Name.

10.2.2 How to Configure the Oracle BI Repository Connections

The section explains how to configure the Oracle BI connection physical databases and connection pools to connect to your database by doing the following:

- Configuring the Oracle Data Warehouse, Oracle EBS OLTP, PeopleSoft OLTP, and Siebel OLTP physical databases.
- Configuring the Oracle BI Repository connection pools.

To configure the Oracle Data Warehouse, Oracle EBS OLTP, PeopleSoft OLTP, or Siebel OLTP physical databases

1. Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd file in the \$SAHome\OracleBI\Server\Repository folder.
2. In the Physical pane, double-click the Oracle Data Warehouse object.
3. In the Database list, click your database type.
4. Save the repository.
5. Click Yes to Check Global Consistency.
6. Repeat steps 1-5 for the Oracle EBS OLTP, PeopleSoft OLTP, and Siebel OLTP data warehouse objects
7. Click OK when the Warnings are displayed.

To configure the Oracle BI Repository connection pools

1. Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd file in the \$SAHome\OracleBI\Server\Repository folder.
2. In the Physical pane:
 - a. Double-click the Oracle Data Warehouse Connection Pool.
 - b. Type the database source name in the Data Source name box.

- c. Type your database User ID in the User name box.
- d. Type your password in the Password box.
3. Repeat Steps a. to d. above for the other connection pools listed above.
4. Save the repository.
5. Click Yes to Check Global Consistency.
6. Click OK when the Warnings are displayed.

10.2.3 How to Configure Oracle BI Repository Variables

The section explains how to configure the Oracle BI Repository variables.

To configure the Oracle BI Repository variables

1. Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd file in the \$SAHome\OracleBI\Server\Repository folder.
2. On the Manage menu, click Variables.
3. In the Variables Manager dialog, click Static.
4. Edit the OLAP_DSN, OLAP_USER, OLAPTBO, OLTP_DSN, OLTP_USER, ORA_EBS_OLTP_DSN, ORA_EBS_OLTP_USER variables, and close the Variables Manager dialog.
5. On the Manage menu, click Security
6. In the Security Manager dialog, click Users, and then:
 - a. Double-click on Administrator user, and type a new password.
 - b. Double-click on SADMIN, and type a new password.
 - c. Save and close the Security Manager.
7. Save the repository.
8. Click Yes to Check Global Consistency.
9. Click OK when the Warnings are displayed.

10.3 Setting up Date-Specific Metrics

The time dimension in the Oracle BI Repository for Oracle Business Analytics Warehouse is a standard or canonical time dimension that links to the important time role in each star schema. The Physical table alias used as a canonical time dimension is W_DAY_D_Common.

If a fact table contains a distinct set of metrics that needs to be reported by different dates, the metadata is organized so that each metric is reported by its causal date.

For example, the Invoice fact table has three metrics called Invoice Amount, Fulfill Amount, and Paid Amount, and each of these metrics need to be reported by the corresponding date—Invoice Date, Fulfill Date, and Payment Date. Additional dates in a fact table that a metric could be queried by are known as Secondary dates. These are available to the end users inside a detailed presentation folder. The detailed presentation folder is typically called the Details folder.

In [Table 10-1](#) each of the metrics reflect the activity related to that event for the entire period, for example, Invoice Amount by Invoice Date, Fulfill Amount by Fulfill date, and Payment Amount by Payment Date.

Table 10–1 Invoice Fact Table Example

Date	Invoice Amount	Fulfill Amount	Payment Amount
January	4000	5000	4500

To implement date-specific metrics

- Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd.
The OracleBIAnalyticsApps.rpd file is located in the \OracleBI\server\Repository folder.
- Right-click Oracle Data Warehouse in the Physical layer, and create a new physical alias for the fact table.
- 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.
- Create a new logical table source in the logical fact table that maps the metrics for the physical fact alias.
The grain of the fact table is the same as the base fact table.

Note: You need to map each metric to one logical table source at the Detail Level.

10.4 Setting Up Additional Time Series Metrics for Oracle Business Analytics Warehouse

The Oracle BI Repository provides a framework to add Period Ago metrics. The Oracle Business Analytics Warehouse is preconfigured with pre-mapped period ago metrics; however, you can map other metrics by using the following procedure.

To set up additional time series metrics

- Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd.
- Right-click Oracle Data Warehouse in the Physical layer, and create a new Period Ago physical alias table.
- Create additional tables in the Physical layer for each Period Ago alias required.
For example, Quarter Ago, Year Ago, and so on.
These aliases need to have the same joins as the base fact table, except for the date join, which you can change in the next step. Setting up this alias is easier to accomplish by copying the base table.
- Change the join to the date dimension (W_DAY_D) to use the appropriate Period Ago Key.
- Map the Period Ago metrics in the logical table using the new fact alias by creating a new logical table source under the fact table.
- Set the content pane levels for the period ago logical table source, to specify the level of the source data.
These settings are the same as the base fact table.
- Save and close the OracleBIAnalyticsApps.rpd file.

10.5 Setting Up Additional Dimension Tables for Oracle Business Analytics Warehouse

Oracle Business Analytics Warehouse is preconfigured to map dimension tables required for analysis. The Physical layer in the Oracle BI Repository provides several other dimension table keys that can be used for certain specific analysis. If you need to set up any of the additional dimensions tables to the Physical layer, perform the following procedure.

To set up additional dimension tables

1. Validate that the dimension table key is resolved appropriately for the data source that you are using.

Note: Dimension tables do not apply to every source system.

2. Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd.
3. Add a dimension table alias in the Physical layer.
4. Join the dimension table alias to the fact table alias using the appropriate keys.
5. Save and close the OracleBIAnalyticsApps.rpd file.

10.6 About the Period Ago Keys for Oracle Business Analytics Warehouse

The Period Ago Key fields are used to set up the time series metrics like Year Ago, Quarter Ago, and so on. The Period Ago Key fields represent metrics for a prior period, for example, Quarter Ago Revenue, Year Ago Revenue, and so on. Oracle Business Analytics Warehouse is preconfigured with a set of fields in the W_DAY_D table. These fields are:

- MONTH_AGO_WID
- QUARTER_AGO_WID
- TRIMESTER_AGO_WID
- WEEK_AGO_WID
- YEAR_AGO_WID

These fields are used in joins to Oracle Business Analytics Warehouse fact tables to achieve the period ago metrics. The joins in Oracle Business Analytics Warehouse uses the Period Ago fields in the W_DAY_D table.

10.7 About Oracle BI Time Repository Variables

The Oracle BI Repository is preconfigured with variables that are used for both reporting and internal usage.

[Table 10-2](#) lists the Oracle BI repository date variables and their descriptions.

Table 10-2 Oracle BI *Repository Date Variables*

Variable Name	Description
CAL_MONTH_YEAR_AGO	Returns the value of Previous Year Month in the YYYY/MM format.

Table 10–2 (Cont.) Oracle BI Repository Date Variables

Variable Name	Description
CURRENT_BALANCE_DK_ AP	Returns the value of the last date key for the available Accounts Payable balance. It is used in Accounts Payable Account Balance Computation.
CURRENT_BALANCE_DK_ AR	Returns the value of the last date key for the available Accounts Receivables balance. It is used in Accounts Receivable Account Balance Computation.
CURRENT_BALANCE_DK_ GL	Returns the value of the last date key for the available General Ledger balance. It is used in General Ledger Account Balance Computation.
CURRENT_DAY	Returns the value of Current Date in the MM/DD/YYYY format.
CURRENT_FSCL_MONTH	Returns the value of Current Fiscal Month in the YYYY/MM format.
CURRENT_FSCL_ QUARTER	Returns the value of Current Quarter in the YYYY Q n format.
CURRENT_FSCL_WEEK	Returns the value of Current Fiscal Week in the YYYY Week nn format.
CURRENT_FSCL_YEAR	Returns the value of Current Fiscal Year in the FYYYYY format.
CURRENT_JULIAN_DAY_ NUM	Returns the value of Current Julian Date Number.
CURRENT_MONTH	Returns the value of Current Month in the YYYY/MM format.
CURRENT_QTR	Returns the value of Current Quarter in YYYY Q n format.
CURRENT_WEEK	Returns the value of Current Week in the YYYY Week nn format.
CURRENT_YEAR	Returns the value of Current Year in the YYYY format.
FSCL_MONTH_YEAR_ AGO	Returns the value of Previous Year Fiscal Month in YYYY/MM format.
FSCL_QTR_YEAR_AGO	Returns the value of Previous Year Quarter in YYYY Q n format.
NEXT_FSCL_MONTH	Returns the value of Next Fiscal Month in the YYYY / MM format.
NEXT_FSCL_QUARTER	Returns the value of Next Quarter in the YYYY Q n.
NEXT_FSCL_WEEK	Returns the value of Next Fiscal Week in the YYYY Weeknn format.
NEXT_FSCL_YEAR	Returns the value of Next Fiscal Year in the FYYYYY format.
NEXT_MONTH	Returns the value of Next Month in the YYYY / MM format.
NEXT_QUARTER	Returns the value of Next Quarter in the YYYY Q n.
NEXT_WEEK	Returns the value of Next Week in the YYYY Weeknn format.
NEXT_YEAR	Returns the value of Next Year in the YYYY format.
PREVIOUS_FSCL_MONTH	Returns the value of Previous Fiscal Month in the YYYY/MM format.
PREVIOUS_FSCL_ QUARTER	Returns the value of Previous Quarter in the YYYY Q n format.
PREVIOUS_FSCL_WEEK	Returns the value of Previous Fiscal Week in the YYYY Weeknn format.

Table 10–2 (Cont.) Oracle BI Repository Date Variables

Variable Name	Description
PREVIOUS_FSCL_YEAR	Returns the value of Previous Fiscal Year in the FYYYYY format.
PREVIOUS_MONTH	Returns the value of Previous Month in the YYYY/MM format.
PREVIOUS_QUARTER	Returns the value of Previous Quarter in the YYYY Q n.
PREVIOUS_WEEK	Returns the value of Previous Week in the YYYY Weeknn format.
PREVIOUS_YEAR	Returns the value of Previous Year in the YYYY format.
REF_JULIAN_DATE	Stores the start date of the Julian calendar and should not be changed.
REF_JULIAN_DATE_NUM	Stores the Julian number for the start of the Julian calendar and should not be changed.
TIME_OFFSET	Returns the difference between the current date and a given number of days value. It is primarily used for testing to simulate an earlier or later date. You could set the variable to the number of days you want the preceding date variables to be moved back.
YEAR_AGO_DAY	Returns the value of year ago date in the mm/dd/yyyy format.

10.8 About User Authentication

You need to configure the user authentication of the Oracle Business Analytics Warehouse, depending on your requirements. The Oracle Business Analytics Warehouse supports various authentication modes, for example, repository authentication, database authentication, and LDAP.

For more information about configuring user authentication, see *Oracle Business Intelligence Server Administration Guide*.

10.9 About the Security or Visibility Configuration

The Oracle Business Analytics Warehouse is preconfigured with a set of user groups. These groups control the visibility of catalogs in the Presentation layer.

For more information about adding a user to a repository user group, see *Oracle Business Intelligence Server Administration Guide*.

[Table 10–3](#) lists the repository user groups in the Oracle Business Analytics Warehouse.

Table 10–3 Repository User Groups

Repository User Group	Description
Administrators	The Administrators user group has all rights and privileges. It cannot be removed.
Agent Scorecard User	This user group is able to view Agent Scorecard application content.
AP Analyst	This user group is able to view application content for Oracle Payables Analytics.
AP Manager	This user group is able to view high-level application content for Oracle Payables Analytics.

Table 10–3 (Cont.) Repository User Groups

Repository User Group	Description
AR Analyst	This user group is able to view application content for Oracle Receivables Analytics.
AR Manager	This user group is able to view high-level application content for Oracle Receivables Analytics.
CFO	This user group is able to view most of the Oracle Financial Analytics application content.
Contact Center and Agent Performance Analyst	This user group is able to view Contact Center Telephony Analytics and Agent Performance application content.
Contact Center and Agent Performance User	This user group is able to view a subset of Contact Center Telephony Analytics and Agent Performance application content.
Contact Center Sales Analyst	This user group is able to view Contact Center Telephony Analytics and Order Management Analytics application content.
Contact Center Sales User	This user group is able to view a subset of Contact Center Telephony Analytics and Order Management Analytics application content.
Controller	This user group is able to view application content for Oracle General Ledger and Profitability Analytics and Siebel Profitability Analytics.
Customer Service Analyst	This user group is able to view Customer Service for Oracle Contact Center Telephony Analytics application content.
Customer Service User	This user group is able to view a subset of Customer Service for Oracle BI Contact Center Telephony Analytics application content.
Contact Center Telephony Analytics User	This user group is able to view Oracle BI Contact Center Telephony Analytics application content.
Financial Analyst	This user group is able to view Oracle Financial Analytics application content.
Human Resources Analyst	This user group is able to view Oracle HR Analytics application content.
Human Resources Vice President	This user group is able to view high-level application content for Oracle HR Analytics application.
Inventory Analyst	This user group is able to view application content for Oracle's Procurement and Spend Analytics Family of Products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics).
Inventory Manager	This user group is able to view high-level application content for Oracle's Procurement and Spend Analytics Family of Products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics).
Primary Owner-Based Security	Used for securing owner-based data elements that come from the transactional system.
Primary Position-Based Security	Used for securing position-based data elements that come from the transactional system.
Purchasing Buyer	This user group is able to view Oracle's Procurement and Spend Analytics Family of Products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) content pertaining to purchasing.

Table 10–3 (Cont.) Repository User Groups

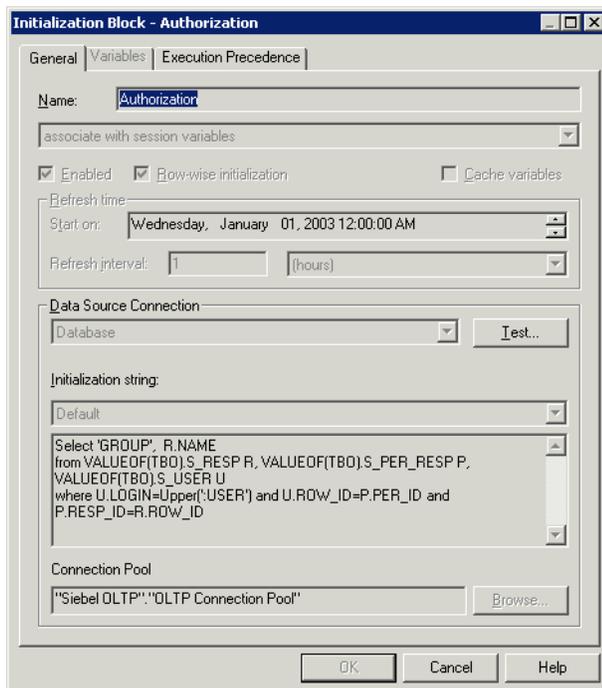
Repository User Group	Description
Sales Executive Analytics	This user group is able to view high-level application content for the Order Management Analytics application.
Sales Manager	This user group is able to view most of the high-level application content for Oracle BI Contact Center Telephony Analytics application.
Sales Manager Analytics	This user group is able to view most of the high-level application content for Oracle BI Contact Center Telephony Analytics application.
Sales Operations Analytics	This user group is able to view operational application content for Oracle BI Contact Center Telephony Analytics application.
Sales Representative Analytics	This user group is able to view low-level application content for Oracle BI Contact Center Telephony Analytics application.
Sales Rev and Fulfill Analyst	This user group is able to view the content for Oracle BI Contact Center Telephony Analytics Revenue and Fulfillment application.
Sales Rev and Fulfill Exec	This user group is able to view the high-level application content for Oracle BI Contact Center Telephony Analytics Revenue and Fulfillment application.
Sales Rev and Fulfill Mgr	This user group is able to view most of the high-level application content for Oracle BI Contact Center Telephony Analytics Revenue and Fulfillment application.
Sales Rev and Fulfill Rep	This user group is able to view low-level application content for Order Management Analytics Revenue and Fulfillment application.
Sales Revenue Analyst	This user group is able to view the content for Oracle BI Contact Center Telephony Analytics Revenue application.
Sales Revenue Exec	This user group is able to view the high-level application content for Oracle BI Contact Center Telephony Analytics Revenue application.
Sales Revenue Mgr	This user group is able to view most of the high-level application content for Oracle BI Contact Center Telephony Analytics Revenue application.
Sales Revenue Rep	This user group is able to view low-level application content for Oracle BI Contact Center Telephony Analytics Revenue application.
Service Delivery and Costs Analyst	This user group is able to view Service Delivery and Costs for Oracle BI Contact Center Telephony Analytics application content.
Service Delivery and Costs User	This user group is able to view a subset of Service Delivery and Costs for Oracle BI Contact Center Telephony Analytics application content.
Supplier Performance Analyst	This user group is able to view Oracle's Procurement and Spend Analytics Family of Products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) content pertaining to supplier performance.
Supplier Performance Manager	This user group is able to view high-level content for Oracle's Procurement and Spend Analytics Family of Products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) content pertaining to supplier performance.
Supply Chain Executive	This user group is able to view Oracle's Procurement and Spend Analytics Family of Products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) content.

10.10 About the Group Variable

The Group variable determines the membership of a user in the various security groups. You need to associate users to the appropriate groups defined in the OracleBIAnalyticsApps.rpd for the security filters to take effect.

If you are using the Oracle 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 about configuring the Group variable, see *Oracle Business Intelligence Presentation Services Administration Guide*. [Figure 10–1](#) shows an example of an initialization block that associates a user to a Group membership.

Figure 10–1 Screenshot of the Initialization Block - Authorization screen



10.11 About Configuring Usage Tracking for Oracle Business Analytics Warehouse

Oracle Business Analytics Warehouse supports the accumulation of usage tracking statistics. The Oracle BI Repository is preconfigured with a connection pool to enable the population of the Usage Tracking log table.

You need to configure this connection pool to connect to the S_NQ_ACCT table. For more information on the Usage Tracking application, see the *Oracle Business Intelligence Server Administration Guide*.

10.12 About the Incremental Deployment of the Oracle BI Applications Repository

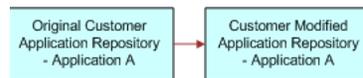
Oracle BI Applications consists of various application families, for example, Supplier Performance Analytics, Contact Center Telephony Analytics, General Ledger and

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.

The figure below 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.

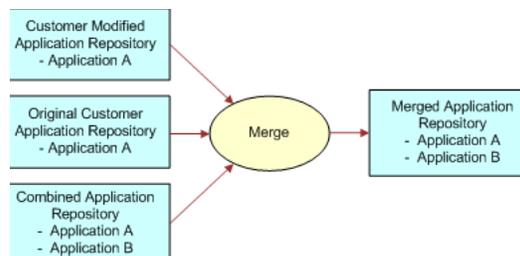
Figure 10–2 Oracle Business Analytics Warehouse Environment



When you purchase another Oracle BI Applications application, you need to extract new metadata for all the modules that you have licensed. Use the Administration Tool merge utility to perform a three-way merge of the original repository, the modified repository, and the combined repository. For more information on merging repositories, see *Oracle Business Intelligence Server Administration Guide*.

The merged repository preserves your modifications from the original Oracle BI Repository and appends the information with the new Oracle BI Repository, as shown in the figure below.

Figure 10–3 Merging with an Oracle BI Repository



You can repeat this merging procedure to add more Oracle BI applications to the Oracle BI Repository

Part IV

Customizing the Oracle Business Analytics Warehouse

Part IV contains instructions for customizing the Oracle Business Analytics Warehouse. The tasks in this section are optional.

Part IV includes the following section: [Chapter 11, "Customizing the Oracle Business Analytics Warehouse."](#)

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:

- [Section 11.1, "Oracle Business Analytics Warehouse Customization Scenarios"](#)
- [Section 11.2, "Category 1 Customizations: Adding Columns to Existing Fact or Dimension Tables"](#)
- [Section 11.3, "Other Types of Customizations Requiring Special Handling"](#)
- [Section 11.4, "Category 2 Customizations: Adding Additional Tables"](#)
- [Section 11.5, "Category 3 Customizations: Adding New Data as a Whole Row into a Standard Dimension Table"](#)
- [Section 11.6, "Configuring Extracts"](#)
- [Section 11.7, "Configuring Loads"](#)
- [Section 11.8, "Configuring Slowly Changing Dimensions"](#)
- [Section 11.9, "About Stored Lookups"](#)
- [Section 11.10, "Codes Lookup"](#)
- [Section 11.11, "About Resolving Dimension Keys"](#)
- [Section 11.12, "About Domain Values"](#)
- [Section 11.13, "Configuring the Domain Value Set with CSV Worksheet Files"](#)
- [Section 11.14, "Configuring the Domain Value Set Using Informatica PowerCenter Designer"](#)
- [Section 11.15, "Configuring Conformed Dimensions"](#)

11.1 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.

The figure below shows the categories of supported customization scenarios, based on the data source.

Figure 11–1 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*.

11.1.1 Types of Customizations

Figure 11–1 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.

11.1.2 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.

11.2 Category 1 Customizations: Adding Columns to Existing Fact or Dimension Tables

This section includes the following topics:

- [Section 11.2.1, "About Extending Mappings"](#)

- [Section 11.2.2, "Impact of Customization on Upgrade"](#)
- [Section 11.2.3, "Typical Steps to Extend Mappings in the Oracle Business Analytics Warehouse"](#)
- [Section 11.2.4, "Scenario for Extending the Oracle Business Analytics Warehouse"](#)

11.2.1 About Extending Mappings

Category 1 customizations involve extracting additional columns from source systems for which pre-packaged adapters are included (for example, Siebel 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 11.5, "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.

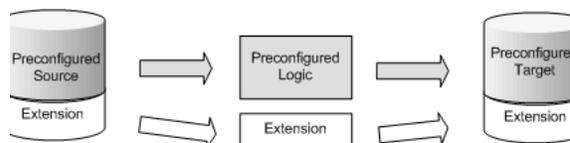
Oracle BI Applications recognizes two types of customization: extension and modification. The supported extension logic allows you to add to existing objects. For example, you can extract additional columns from a source, pass them through existing mappings, and populate new columns added to an existing table. Generally, Oracle 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 11–2 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 needs to 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.

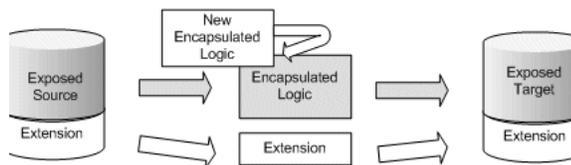
Note: Targets should not be added to a mapping.

11.2.2 Impact of Customization on Upgrade

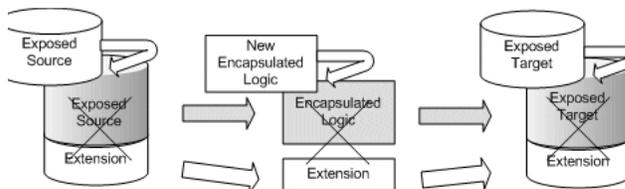
When upgrading, you will deploy customized mappings on an individual basis. Only the actual mappings that have changed will be applied in your existing environment. This means any mappings that have not changed will not be affected, so any customizations made to these mappings remain. Only the mappings 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.

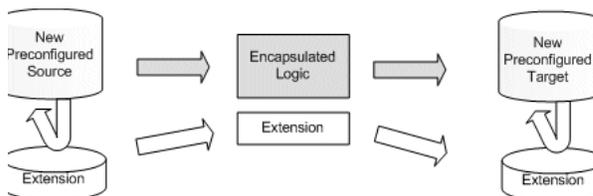
Figure 11–3 Encapsulating logic



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 11-4 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 11-5 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.

11.2.2.1 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.

11.2.3 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.

11.2.4 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 11.2.4.1, "Tips for Modifying the SQL Override in a Source Qualifier"](#)
- [Section 11.2.4.2, "Example of Extracting Data from a Siebel Source"](#)
- [Section 11.2.4.3, "Including a Source Table for the Change Capture Process"](#)
- [Section 11.2.4.4, "Example of Extracting Data from a Non-Siebel Packaged Source"](#)
- [Section 11.2.4.5, "Example of Extracting Data from a Universal Source"](#)
- [Section 11.2.4.6, "Example of Loading Data into an Existing Target Table"](#)
- [Section 11.2.4.7, "Updating DAC"](#)

11.2.4.1 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 11.2.4.2, "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.

11.2.4.2 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.

11.2.4.3 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.

11.2.4.4 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.

Because Oracle Applications always runs 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.

11.2.4.5 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)

Column Name	Data Type
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.

11.2.4.6 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.

11.2.4.7 Updating DAC

After making these changes to the mappings, you need to register the changes in DAC. You need to 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 the *Oracle Business Intelligence Data Warehouse Administration Console Guide*.

11.3 Other Types of Customizations Requiring Special Handling

This section includes the following topics:

- [Section 11.3.1, "Modifying Category 2 SCD Triggers"](#)
- [Section 11.3.2, "Adding Dimensions to Existing Facts"](#)
- [Section 11.3.3, "Adding Date Dimensions to Existing Facts"](#)
- [Section 11.3.4, "Adding Currencies to an Existing Table"](#)

11.3.1 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, needs to 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_BOMHeaderDimension mapping compares the following columns:

- BOM_HEADER
- BOM_VERSION
- BASE_QTY
- ACTIVE_FLG

If you wanted to include BOM_VERSION as part of Category 2 logic, you would change the logic for 'TYPE2_COLS_DIFF' from the following:

```
IIF(BOM_NUMBER != LKP_BOM_NUMBER, 'Y',
IIF(BOM_VERSION != LKP_BOM_VERSION, 'Y',
IIF(BASE_QTY != LKP_BASE_QTY, 'Y',
IIF(ACTIVE_FLG != LKP_ACTIVE_FLG, 'Y',
'N'))))
```

To this:

```
IIF(BOM_NUMBER != LKP_BOM_NUMBER, 'Y',
IIF(BOM_VERSION != LKP_BOM_VERSION, 'Y',
IIF(BASE_QTY != LKP_BASE_QTY, 'Y',
IIF(ACTIVE_FLG != LKP_ACTIVE_FLG, 'Y',
IIF(BOM_VERSION != LKP_BOM_VERSION, 'Y',
'N')))))
```

11.3.2 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 11-1](#) 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 11-1 Formats to Derive INTEGRATION_ID

Dimension	Foreign Key	When Source is Oracle Application	When Source is Siebel Application
W_AP_TERMS_D		TO_CHAR(TERM_ID)	Not applicable
W_BUSN_LOCATION_D	ASSET_LOC_WID	ASSET_LOC~' LOCATION_ID	Not applicable
W_BUSN_LOCATION_D	EMP_LOC_WID	EMP_LOC~' LOCATION_ID	Not applicable
W_BUSN_LOCATION_D	INVENTORY_LOC_WID	STORAGE_LOC' '~' ORGANIZATION_ID '~' SUBINVENTORY_CODE '~' INVENTORY_LOCATION_ID	Not applicable
W_BUSN_LOCATION_D	PLANT_LOC_WID	'PLANT' '~' TO_CHAR(ORGANIZATION_ID)	Not applicable
W_BUSN_LOCATION_D	RECEIVING_LOC_WID	'RECIPIENT_LOC' '~' TO_CHAR(LOCATION_ID)	Not applicable
W_BUSN_LOCATION_D	STORAGE_LOC_WID	'STORAGE_LOC' '~' ORGANIZATION_ID '~' SECONDARY_INVENTORY_NAME '~'	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
W_CUSTOMER_LOC_D		To get the customer location key, look up W_CUSTOMER_LOC_USE_D as below	Not applicable
W_CUSTOMER_LOC_USE_D		TO_CHAR(SITE_USE_ID) - Get Site Use Id from HZ_CUST_ACCOUNT_ROLES	Not applicable
W_FREIGHT_TERMS_D		LOOKUP_CODE	Not applicable
W_GL_ACCOUNT_D		to_char(ccid)	Not applicable
W_INT_ORG_D	COMPANY_ORG_KEY	COMPANY' '~' TO_CHAR(SET_OF_BOOKS_ID)	S_ORG_EXT.ROW_ID
W_INT_ORG_D	*_ORG_KEY	Remove any prefixes and use TO_CHAR()	S_ORG_EXT.ROW_ID

Table 11–1 (Cont.) Formats to Derive INTEGRATION_ID

Dimension	Foreign Key	When Source is Oracle Application	When Source is Siebel Application
W_ORG_D	CUSTOMER_WID	TO_CHAR(CUSTOMER_ID) - CUSTOMER_ID is CUST_ACCOUNT_ID from HZ_CUST_ACCOUNTS	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)
W_PAYMENT_METHOD_D		LOOKUP_CODE	Not applicable
W_PAYMENT_METHOD_D		TO_CHAR(TERM_ID)	Not applicable
W_PERSON_D	CUST_CONTACT_WID	TO_CHAR(PARTY_ID) - PARTY_ID from HZ_PARTY_RELATIONS	S_CONTACT.ROW_ID
W_PRODUCT_D	PRODUCT_WID	TO_CHAR(INVENTORY_ITEM_ID)	S_PROD_INT.ROW_ID
W_SALES_PRODUCT_D		TO_CHAR(INVENTORY_ITEM_ID) '~' TO_CHAR(ORGANIZATION_ID)	Not applicable

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.

11.3.3 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.

11.3.4 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 maplet 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.

11.4 Category 2 Customizations: Adding Additional Tables

This section includes the following topics:

- [Section 11.4.1, "About Creating New Dimension or Fact Tables"](#)
- [Section 11.4.2, "Using Custom Folders in the Oracle Business Analytics Warehouse"](#)
- [Section 11.4.3, "Creating Custom Informatica Workflows"](#)
- [Section 11.4.4, "Important Notes About Customizing the Oracle Business Analytics Warehouse"](#)
- [Section 11.4.5, "Including a Source Table for Change Capture Process"](#)
- [Section 11.4.6, "Adding a New Dimension in the Oracle Business Analytics Warehouse"](#)
- [Section 11.4.7, "Adding a New Fact Table in the Oracle Business Analytics Warehouse"](#)
- [Section 11.4.8, "Adding a New Dimension Table for a New Fact Table in the Oracle Business Analytics Warehouse"](#)

11.4.1 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.

11.4.1.1 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.

11.4.1.2 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. Make sure 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.

11.4.2 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.

11.4.3 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.

11.4.4 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.

11.4.4.1 Additional Customization Instructions

- **Table definitions in Informatica.** Make sure 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 needs to 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 allows 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 maplet 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.

11.4.5 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.

The DAC Client does not validate this information when entering data.
4. Create Image tables in the Siebel transactional database.
 - a. Right-click on the table record in the DAC Client, 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.

11.4.6 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.

11.4.7 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 11.4.5, "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.

11.4.8 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.

11.5 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 needs to 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, make sure 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.

11.6 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 11.6.1, "Extracting Additional Data"](#)
- [Section 11.6.2, "Setting Up the Delimiter for Source Files"](#)

11.6.1 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.

11.6.1.1 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.

11.6.1.2 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.

11.6.2 Setting Up the Delimiter for Source Files

You must make sure 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.

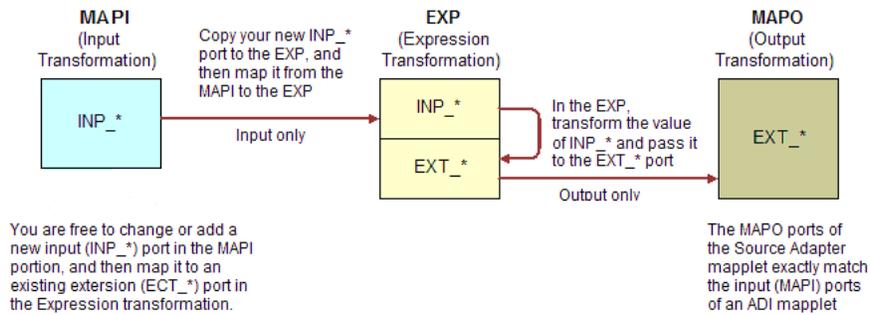
To set up the delimiter for source files

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.

11.6.3 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.

The figure below 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 11–6 Components of the Source adapter mapplet

In the figure above, 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.

11.7 Configuring Loads

The Oracle Business Analytics Warehouse prepackages load mappings for every data warehouse table.

11.7.1 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.

11.7.2 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.

The figure below provides an example of the beginning of the extract process. It shows the sequence of events over a two day period during which the information in the source table has changed. On day one, the data is extracted from a source table and loaded into the Oracle Business Analytics Warehouse table. On day two, Sales Order number three is deleted and a new sales order is received, creating a disparity between the Sales Order information in the two tables.

Figure 11–7 Extract and load mappings

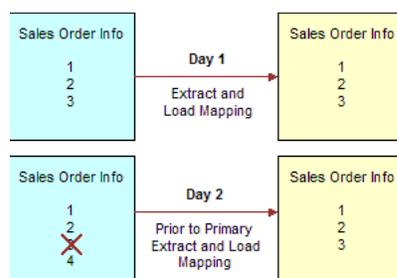


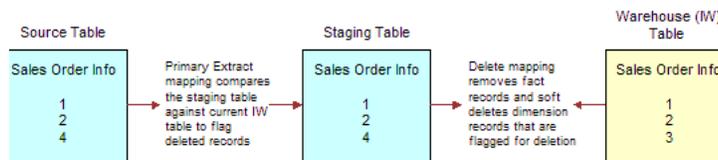
Figure 11–8 shows the primary extract and delete process that occurs when day two's information is extracted and loaded into the Oracle Business Analytics Warehouse from the source. The initial extract brings record four into the Oracle Business Analytics Warehouse. Then, using a primary extract mapping, 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 11–8](#). When the extract and load mappings run, the new sales order is added to the warehouse.

Figure 11–8 Primary Extract and Delete Mappings



11.7.3 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.

11.7.3.1 Deleting the Configuration for Source-Archived Records

Some sources archive records in separate databases and retain only the current information in the main database. If you have enabled the delete mappings, you must reconfigure the delete mappings in the Oracle Business Analytics Warehouse to retain the archived data.

To retain source-archived records in the Oracle Business Analytics Warehouse, make sure the `$$LAST_ARCHIVE_DATE` parameter value is set properly to reflect your archive date. The delete mappings will not mark the archived records as 'deleted'. For more information about extract and delete mappings, see [Section 11.7.3, "About Working with Primary Extract and Delete Mappings"](#).

11.7.3.2 Enabling Delete and Primary Extract Sessions

If you want to mark your source-deleted records as deleted in the Oracle Business Analytics Warehouse, you need to enable the delete and primary extract tasks for your application.

To enable primary extract and delete sessions

1. In DAC, go to the Design view, and select the appropriate custom container from the drop-down list.
2. Click the Tasks tab.
3. Query for all tasks containing the string 'Delete' or 'Primary'.
4. Deselect the Inactive check box for those tasks.

11.8 Configuring Slowly Changing Dimensions

The Oracle Business Analytics Warehouse provides Category 2 slowly changing dimension (SCD) functionality, which allows you to track the history of updates to dimension records. When a record in the Oracle Business Analytics Warehouse has an update, the updated information is posted into a new row and the old information is kept for historical reporting purposes.

The Oracle Business Analytics Warehouse identifies and applies the slowly changing dimension logic 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.

By default, all dimensions are using Category 1 updates. If you need change a dimension to Category 2 SCD update, follow the procedure below.

The table below shows the dimensions that are required using the SCD Category 2 update.

Table 11–2 Dimensions that are required using the SCD Category 2 update

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 and Siebel

Note: If you want to turn off the TYPE2_FLG for any of the dimensions listed above, you will need to customize the source dependent extract mapping. The following changes should be made:

1. Override the SQL statement in the Source Qualifier to filter out the historical records.
The change will vary depending on the OLTP data model. For example, you can select only records effective from the Informatica parameter \$\$\$SessStartTime.
 2. Disconnect the ports that bring the source effective from and to dates.
-
-

To turn a dimension to Category 2 SCD update enabled

1. In DAC, go to the Design view.
2. Select the appropriate custom container.
3. Click the Tasks tab.
4. Query for the SIL task that populating the dimension.
5. Click the Parameters subtab, and set the value for \$\$TYPE2_FLG to Y.

11.8.1 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 allows you to determine the category of SCD you require. However, before you can select the appropriate type of SCD, you must understand their differences.

11.8.1.1 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.

11.8.2 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.

11.8.2.1 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 the figure below, the State Name column for the supplier KMT is changed in the source table Suppliers, because it was incorrectly entered as California. When the data is loaded into the 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 11–9 An example Type 1 Slowly Changing Dimension

Supplier (Source)			Supplier (Analytic Data Whs.)		
Supplier Name	State Name	Contact Name	Supplier Name	State Name	Contact Name
Acme	NY	Chris	Acme	NY	Chris
KMT	CA MI	Suzanne	KMT	MI	Suzanne

11.8.2.2 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 allows you to track historically significant attributes. The old records point to all history prior to the latest change, and the new record maintains the most current information.

Slowly changing dimensions work in different parts of a star schema (the fact table and the dimension table). The figure below shows how an extract table (SOURCE_CUSTOMERS) becomes a 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 11–10 An example Type 2 Slowly Changing Dimension

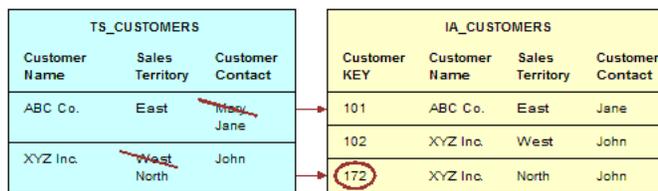
SOURCE_CUSTOMERS			W_ORG_D			
Customer Name	Sales Territory	Customer Contact	Customer KEY	Customer Name	Sales Territory	Customer Contact
ABC Co.	East	Mary	101	ABC Co.	East	Mary
XYZ Inc.	West	John	102	XYZ Inc.	West	John

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. The figure below shows that records for the two customers, ABC Co., and XYZ inc. have changed when compared with the figure below. Notice that ABC's Customer Contact has changed from Mary to Jane, and XYZ's Sales Territory has changed from West to North.

As discussed earlier in this example, the Customer Contact column is historically insignificant; therefore a 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 the figure below, instead of overwriting the Sales Territory column in the XYZ's record, a new record is added, assigning a new ROW_WID, 172, to XYZ in W_ORG_D. XYZ's original record, 102, remains and is linked to all the sales that occurred when XYZ was located in the West sales territory. However, new sales records coming in are now attributed to ROW_WID 172 in the North sales territory.

Figure 11–11 An example Type 2 Slowly Changing Dimension



11.8.2.3 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 you can set up the Oracle Business Analytics Warehouse to populate the missing effective dates using a wrapper mapping. This situation is discussed in this section. By default, these wrapper sessions are disabled and need to be enabled in order to be executed.

For example, in the W_ORG_D table previously discussed, XYZ moved to a new sales territory.

If your source system supplied historical data on the location changes, your table 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 11–3](#).

Table 11–3 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 you schedule a wrapper session to execute, 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 11–4](#).

Table 11–4 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. Finally, if your source system does not supply any Category 2 information, you may disable the wrapper session completely; in this case all Category 2 work is handled by the Analytics Data Interface maplet.

11.9 About Stored Lookups

A lookup transformation allows 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 11.10, "Codes Lookup"](#).
- Dimension Key Lookups. For more information, see [Section 11.11, "About Resolving Dimension Keys"](#).

11.10 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:

11.10.1 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.

11.10.2 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.

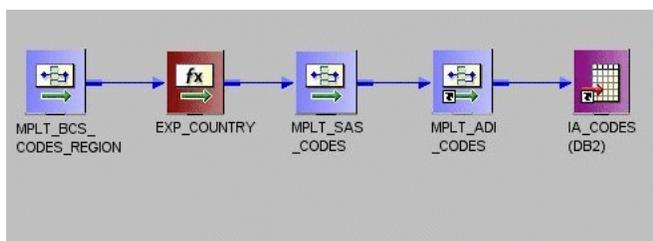
To understand how codes mappings function, it is helpful to first understand the columns within `W_CODES`. [Table 11-5](#) describes these columns.

Table 11-5 Columns in Code Maplet

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]`. The figure below shows an example of a code mapping in Informatica PowerCenter Designer.

Figure 11-12 Example of a code mapping in Informatica PowerCenter Designer



11.10.3 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.

11.10.4 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.

11.11 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.

11.11.1 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 11-6](#).

Table 11-6 Columns Used in the load mapping Dimension Key Lookup

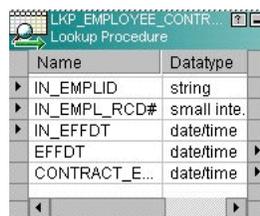
Port	Description
INTEGRATION ID	Uniquely identifies the dimension entity within its source system. Formed from the transaction in the source adapter of the fact table.
DATASOURCE_NUM_ID	Unique identifier of the source system instance.
Lookup Date	The primary date of the transaction; for example, receipt date, sales date, and so on.

In the figure below, 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 the figure below, you can see the effective dates used to provide the effective period of employee contracts.

Figure 11-13 Lookup for Employee Contract Data



11.12 About Domain Values

The Oracle Business Analytics Warehouse foundation comprises a data model that accommodates data from disparate source systems. Data is sourced from operational systems and systematically molded into a source-independent format. After the data is made source independent, it can then be used to create key metrics for analytic reporting, so that metric calculations are not source dependent. This clear separation allows you to swap source systems or integrate additional source systems without having to reconfigure the metric calculations to accommodate each source system's requirements.

One method for transforming source data into a source-independent format is to convert the source-supplied values to domain values. Domain values are a set of distinct values used to calculate prepackaged metrics. These values are provided by the Oracle Business Analytics Warehouse to allow you to create metric calculations independent of source system values.

11.12.1 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.

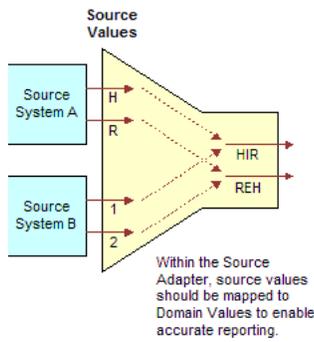
11.12.1.1 Preparing to Define the Rules

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 11.14, "Configuring the Domain Value Set Using Informatica PowerCenter Designer"](#).

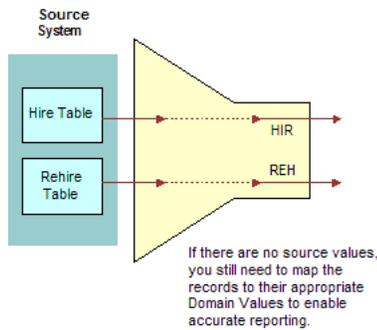
[Figure 11-14](#) illustrates how the source values are converted to the domain values—HIR and REH.

Figure 11–14 Source Values Translated to Domain Values



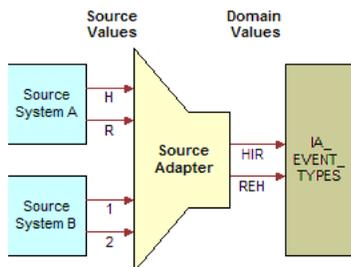
The figure below illustrates a different situation where the records may not contain a source value that flags the record as Hire or Rehire. In this case, the source system stores hires in one table and rehires in another table. To make this work, one possible solution is to modify the extract mappings 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 11–15 Source Values in Different Tables Translated to Domain Values



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 the figure below.

Figure 11–16 HIR and REH Values Populating the W_EVENT_TYPES table



11.12.2 About the Importance of Domain Values

Values in the `W_EVENT_TYPES` table are used to create metrics in the front end. Some metrics are defined using domain values. For example, seven metrics use the HIR and

REH event group code in their calculation. The following are the seven metrics, along with their descriptions and calculations:

11.12.2.1 Hire Count

This metric counts all hires for a specified period. The calculation is:

```
SUM(CASE WHEN (CMMNEVTP.W_EVENT_GRP_CODE IN ('HIR', 'REH')) THEN EVNT.EVENT_CNT
ELSE 0 END)
```

11.12.2.2 Re-hires Ratio

This metric determines the ratio of rehires to all employees hired during a specified period. The calculation is:

```
CASE WHEN SUM(CASE WHEN CMMNEVTP.W_EVENT_GRP_CODE IN ('REH', 'HIR') THEN
EVNT.EVENT_CNT ELSE 0 END) = 0 THEN 0 ELSE SUM(CASE WHEN CMMNEVTP.W_EVENT_GRP_CODE
IN ('REH') THEN EVNT.EVENT_CNT ELSE 0 END)/SUM(CASE WHEN CMMNEVTP.W_EVENT_GRP_CODE
IN ('REH', 'HIR') THEN EVNT.EVENT_CNT ELSE 0 END) END
```

11.12.2.3 New Hire Count

This metric counts the head count hired for regular full-time positions. The calculation is:

```
SUM(CASE WHEN CMMNEMPT.FULL_TIME_FLAG = 'Y' AND CMMNEMPT.EMP_CAT_CODE = 'R' AND
(CMMNEVTP.W_EVENT_GRP_CODE = 'HIR' OR CMMNEVTP.W_EVENT_GRP_CODE = 'REH') AND
EVNT.EVENT_DK >= (CMMNDATE.DATE_KEY - 365) AND EVNT.EVENT_DK <= CMMNDATE.DATE_KEY
THEN EVNT.EVENT_CNT ELSE 0 END)
```

11.12.2.4 Newly Separated Veterans - New Hires

This metric counts the regular full-time and part-time employees who belong to this category of veterans and were hired during the previous 12 months. The calculation is:

```
SUM(CASE WHEN CMMNEMPD.VETERAN_STAT_CODE = '4' AND CMMNEMPT.EMP_CAT_CODE = 'R' AND
(CMMNEVTP.W_EVENT_GRP_CODE = 'HIR' OR CMMNEVTP.W_EVENT_GRP_CODE = 'REH') AND
EVNT.EVENT_DK >= (CMMNDATE.DATE_KEY - 365) AND EVNT.EVENT_DK <= CMMNDATE.DATE_KEY
THEN EVNT.EVENT_CNT ELSE 0 END)
```

11.12.2.5 Other Protected Veterans - New Hires

This metric counts regular full-time and part-time employees who belong to this category of veterans. The calculation is:

```
SUM(CASE WHEN CMMNEMPD.VETERAN_STAT_CODE = '3' AND CMMNEMPT.EMP_CAT_CODE = 'R' AND
(CMMNEVTP.W_EVENT_GRP_CODE = 'HIR' OR CMMNEVTP.W_EVENT_GRP_CODE = 'REH') AND
EVNT.EVENT_DK >= (CMMNDATE.DATE_KEY - 365) AND EVNT.EVENT_DK <= CMMNDATE.DATE_KEY
THEN EVNT.EVENT_CNT ELSE 0 END)
```

11.12.2.6 Special Disabled Veteran Head count - New Hires

This metric counts regular full-time and part-time employees who belong to this category of veterans and were hired during the previous 12 months. The calculation is:

```
SUM(CASE WHEN CMMNEMPD.VETERAN_STAT_CODE = '1' AND CMMNEMPT.EMP_CAT_CODE = 'R' AND
(CMMNEVTP.W_EVENT_GRP_CODE = 'HIR' OR CMMNEVTP.W_EVENT_GRP_CODE = 'REH') AND
EVNT.EVENT_DK >= (CMMNDATE.DATE_KEY - 365) AND EVNT.EVENT_DK <= CMMNDATE.DATE_KEY
THEN EVNT.EVENT_CNT ELSE 0 END)
```

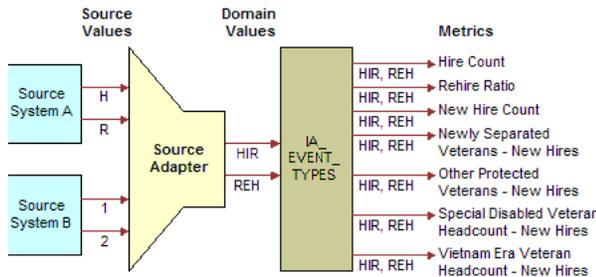
11.12.2.7 Vietnam Era Veteran Head count - New Hires

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 the figure below.

Figure 11-17 Metric Values From HIR and REH Domain Values



11.12.3 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.

11.13 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 11.14, "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\server\infa_shared\LkpFiles` folder (for example, `D:\Informatica\PowerCenter8.1.1\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.

11.14 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 11.13, "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.

11.15 Configuring Conformed Dimensions

This section provides procedures on configuring objects that apply to more than one application and contains the following topics:

- [Section 11.15.1, "Configuring Conformed Dimensions for Universal Adapter"](#)

11.15.1 Configuring Conformed Dimensions for Universal Adapter

This section provides configuration procedures for modifying dimensions that are loaded using Universal Adapter.

11.15.1.1 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.

Symbols

\$\$PERIOD

- about, 6-18
- configuring, 6-19

\$\$WH_DATASOURCE_NUM_ID

- about, 6-18
- configuring, 6-19

A

Applications

- Oracle Contact Center Telephony Analytics, 4-1, 8-1, 9-1
 - Oracle Financial Analytics, 5-1
 - Oracle HR, 7-1
 - Oracle Order Management and Fulfillment Analytics, 6-1
 - Oracle Sales Analytics, 8-1
 - Oracle Supply Chain Analytics, 4-1, 8-1
- ### AR Aging calculations
- including UnApplied Payments, 5-12
- ### attributes
- historically significant attributes, identifying, 11-32

C

calculations

- including UnApplied Payments in AR Aging calculations, 5-12

code lookup

- about, 11-35
- code mappings, 11-36
- codes mapplets, 11-37
- sessions for lookups, configuring, 11-37
- W_CODES table, 11-36

configuration steps

- all source systems, 3-1
- controlling data set
 - Oracle Contact Center Telephony Analytics, 9-1
 - Oracle Financial Analytics, 5-1, 5-12
 - Oracle Human Resources Analytics, 7-1
 - Oracle Procurement and Spend Analytics, 4-1
 - Oracle Sales Analytics, 8-1

- source independent, 3-1
 - Supply Chain and Order Management Analytics, 6-1
- ### Oracle BI Repository, 10-1
- ### Oracle EBS-specific, 3-16
- ### PeopleSoft-specific, 3-26
- ### required before full load
- Oracle Contact Center Telephony Analytics, 9-1
 - Oracle Financial Analytics, 5-1
 - Oracle Human Resources Analytics, 7-1
 - Oracle Procurement and Spend Analytics, 4-1
 - Oracle Sales Analytics, 8-1
 - source independent, 3-1
 - Supply Chain and Order Management, 6-1
- ### conformed dimensions
- universal source, configuring for, 11-44
- ### cross-module configuration, performing
- codes lookup, 11-35
 - conformed dimensions, configuring, 11-44
 - dimension keys, resolving, 11-37
 - domain values, working with, 11-39
 - loads, configuring, 11-28
 - records, filtering and deleting, 11-28
 - slowly changing dimensions, changing, 11-31
 - stored lookups, configuring, 11-35
- ### Customer Status History Fact table
- about, 6-18
 - configuring, 6-19

D

Data Source NUM ID

- configuring \$\$WH_DATASOURCE_NUM_ID, 3-15

DATASOURCE_NUM_ID

- about, 3-15
- configuring, 3-15

dimensions

- key resolution, using a lookup, 11-38
- keys, resolving, 11-37

domain values

- domain value conversion process, understanding, 11-39
- domain value set, configuring, 11-43
- domain value set, extending, 11-42

importance, understanding, 11-40
working with, 11-39

E

ETL
troubleshooting, 8-3
Event Types
configuring for PSFT and HR, 7-31
extracts, configuring
extracting additional data, about, 11-25
extracting data from new source table, 11-26
extracting new data from existing source
table, 11-26

H

historically significant attributes
identifying, 11-32
HR Analytics
configuring Event Types for PSFT, 7-31

I

Informatica
username and password, 10-1
installing
Oracle BI repository, setting up, 10-2

L

loads, configuring
about, 11-28
Source Adapter mapplet. creating, 11-27

O

Oracle BI Supply Chain Analytics
domain values and CSV worksheet files, 4-4
universal source, configuring expenses, 4-13
Oracle Contact Center Telephony Analytics
Benchmarks and Targets table, setting up, 9-8
Contact Representative Snapshot table, setting
up, 9-7
Contact Status Type columns, about configuring
for, 9-6
dates and times, configuring, 9-11
dimension key resolution, 9-2
dimension table, loading, 9-2
Event Type columns, about configuring for, 9-3
fact table, loading, 9-3
overview, 9-1
Representative Activities table, setting up, 9-6
Oracle Financial Analytics
configuration steps before full load, 5-2
overview, 5-1
Oracle HR Analytics, 7-37
domain values and CSV worksheet files, 7-3
overview, 7-1
Oracle Order Management and Fulfillment Analytics
options, 6-1

overview, 6-1
Oracle Procurement and Spend Analytics
configuration steps before full load, 4-3
overview, 4-1
OracleBIAnalyticsApps.rpd
Informatica username and password, 10-1
Order Management and Fulfillment Analytics
multiple products, about tracking, 6-14

P

password
OracleBIAnalyticsApps.rpd, 10-1
Payroll table, about aggregating, 7-37
PeopleSoft Trees
configuring department trees, 7-32
primary extract and delete mapping
delete and primary extract sessions,
enabling, 11-31
source-archived records, deletion
configuration, 11-30
working with, 11-30

R

records
source-archived records, deletion
configuration, 11-30
records, filtering and deleting
about, 11-28
primary extract and delete mappings,
understanding, 11-29
primary extract and delete mappings, working
with, 11-30
repository
Oracle BI repository, setting up, 10-2
repository, setting up, 10-2
responsibilities
in Usage Accelerator, 8-3

S

slowly changing dimensions, configuring
about, 11-31
effective dates, 11-34
historically significant attributes,
identifying, 11-32
Type I changing dimension, 11-33
Type II slowly changing dimension, 11-33
Source Adapter mapplet
creating, 11-27
source-archived records
deletion configuration, 11-30
stored lookups
configuring, 11-35

T

troubleshooting
ETL failure, 8-3

U

UnApplied Payments

Using in AR Aging calculations, 5-12

Universal Adapter

conformed dimensions, configuring, 11-44

Oracle BI Supply Chain Analytics, configuring
expenses, 4-13

Usage Accelerator

ETL failure, 8-3

username

OracleBIAnalyticsApps.rpd, 10-1

