

Oracle® Business Intelligence Applications

New Features Guide

Version 7.9.6.2

E16811-01

September 2010

E16811-01

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. 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, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

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

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	v
How to Use the New Features Guide.....	v
Related Documentation.....	v
Audience.....	v
Documentation Accessibility	v
1 Configuring Oracle Supply Chain and Order Management Analytics for Oracle's JD Edwards EnterpriseOne	
1.1 Before You Begin.....	1-1
1.2 Configuration Required Before a Full Load.....	1-1
1.2.1 About Configuring Domain Values and CSV Worksheet Files for JD Edwards EnterpriseOne Supply Chain and Order Management Analytics.....	1-2
1.2.2 How to Configure User Defined Code Categories	1-2
1.2.3 How to Configure Movement Type Domain Values	1-3
1.2.4 How to Configure Transaction Types Domain Values for Sales Invoice Lines.....	1-4
1.2.5 How to Configure Transaction Types Domain Values for Sales Order Lines.....	1-4
1.2.6 How to Configure Sales Order Statuses	1-5
1.2.7 How to Configure Channel Type Groups.....	1-7
1.2.8 How to Configure Inventory Consignments.....	1-8
1.2.9 How to Configure Inventory Return Locations	1-8
1.2.10 How to Process Bill of Material Explosion.....	1-9
1.2.11 How to Configure the Left Bound and Right Bound Calculation Option	1-10
2 Support for JD Edwards EnterpriseOne Ledger Types in Oracle Financial Analytics	
2.1 Ledger Type Mapping.....	2-1
2.2 Dashboard Page Prompts	2-2
2.3 Configuring Ledger Type	2-3
3 PeopleSoft Chartfields for Oracle BI Applications	
3.1 New PeopleSoft Chartfields	3-1
3.1.1 Configuring the New Chartfields.....	3-1

4	Enhanced Support for PeopleSoft Product/Purchasing Categories	
4.1	About Product/Purchasing Categories in PeopleSoft.....	4-1
4.2	Configuration Required to Enable Reporting on Product/Purchasing Categories.....	4-1
5	Support for Currency Balance Types in Oracle Financial Analytics for Oracle EBS and Oracle’s PeopleSoft	
5.1	Oracle EBS Currency Balance Types.....	5-1
5.2	PeopleSoft Currency Balance Types.....	5-3
6	New Features and Updates for Oracle Project Analytics	
6.1	New Project Commitments Subject Area for Oracle EBS.....	6-1
6.1.1	Overview.....	6-1
6.1.2	Before You Begin.....	6-2
6.1.3	Configuration Steps for the Project Commitments Subject Area	6-2
6.2	Updates to Universal Adapter for Oracle Project Analytics.....	6-3
6.3	Configuration Update for Project Customer for Oracle EBS.....	6-4
6.4	My Oracle Support Technical Note on How to Configure RPD if the Cost Aggregate Grain is Set to Period or Year	6-5
6.5	New Dimension for Project Budget and Forecast Fact.....	6-5
6.6	Linear Spread Metrics for Project Budget Fact.....	6-5
7	Support for Global 1 Amount Exchange Date Calculation	
7.1	Configuring the Exchange Date for JD Edwards EnterpriseOne Accounts Payable	7-1
7.2	Exchange Date for JD Edwards EnterpriseOne Account Receivables	7-2
8	Support for Back-Dated Changes to Position Hierarchy	
8.1	Overview	8-1
8.2	Supported Back-Dated Change Scenarios.....	8-1
8.2.1	Back-Dated Change Scenarios That Affect Position Hierarchy	8-2
8.2.1.1	Scenario 1	8-3
8.2.1.2	Scenario 1a.....	8-3
8.2.1.3	Scenario 2.....	8-3
8.2.1.4	Scenario 3.....	8-4
8.2.1.5	Scenario 3a.....	8-4
8.2.1.6	Scenario 3b.....	8-5
8.2.1.7	Scenario 3c	8-5
8.2.1.8	Scenario 3d.....	8-6
8.3	Objects Changed to Support Back-Dated Changes in Supervisor/Position Hierarchy ...	8-6
8.3.1	Data Model Changes.....	8-6
8.3.2	Informatica Code Changes.....	8-6
8.3.2.1	Mappings.....	8-6
8.3.2.2	Sessions/Workflows	8-7
8.3.2.3	Reusable Lookup	8-7
8.3.3	DAC Updates	8-7

Preface

This guide describes the new features in Release 7.9.6.2 of Oracle Business Intelligence Applications.

This preface contains the following sections:

- [How to Use the New Features Guide](#)
- [Related Documentation](#)
- [Audience](#)
- [Documentation Accessibility](#)

How to Use the New Features Guide

The New Features Guide describes the new features and configuration updates in Oracle BI Applications release 7.9.6.2. Use this guide as a companion guide to the Oracle BI Applications release 7.9.6.1 set of guides.

Related Documentation

In addition to the New Features Guide, the following guides have been published for release 7.9.6.2:

- *Oracle Business Intelligence Applications System Requirements and Supported Platforms*
- *Oracle Business Intelligence Applications Upgrade Guide for Informatica PowerCenter Users*
- *Oracle Business Intelligence Applications Release Notes*

Audience

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

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. 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.

Access to Oracle Support

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

Configuring Oracle Supply Chain and Order Management Analytics for Oracle's JD Edwards EnterpriseOne

This chapter contains configuration information for Oracle Supply Chain and Order Management Analytics for Oracle's JD Edwards EnterpriseOne.

This chapter includes the following topics:

- [Section 1.1, "Before You Begin"](#)
- [Section 1.2, "Configuration Required Before a Full Load"](#)

1.1 Before You Begin

Before you follow the configuration steps in this chapter, perform the source-independent configuration steps in the following chapters of the *Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users*:

- "Configuring Common Areas and Dimensions"
- "Configuring Oracle Supply Chain and Order Management Analytics"

1.2 Configuration Required Before a Full Load

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

- [Section 1.2.1, "About Configuring Domain Values and CSV Worksheet Files for JD Edwards EnterpriseOne Supply Chain and Order Management Analytics"](#)
- [Section 1.2.2, "How to Configure User Defined Code Categories"](#)
- [Section 1.2.3, "How to Configure Movement Type Domain Values"](#)
- [Section 1.2.4, "How to Configure Transaction Types Domain Values for Sales Invoice Lines"](#)
- [Section 1.2.5, "How to Configure Transaction Types Domain Values for Sales Order Lines"](#)
- [Section 1.2.6, "How to Configure Sales Order Statuses"](#)
- [Section 1.2.7, "How to Configure Channel Type Groups"](#)
- [Section 1.2.8, "How to Configure Inventory Consignments"](#)
- [Section 1.2.9, "How to Configure Inventory Return Locations"](#)

- [Section 1.2.10, "How to Process Bill of Material Explosion"](#)
- [Section 1.2.11, "How to Configure the Left Bound and Right Bound Calculation Option"](#)

1.2.1 About Configuring Domain Values and CSV Worksheet Files for JD Edwards EnterpriseOne Supply Chain and Order Management Analytics

Table 1–1 lists the CSV worksheet files and the domain values for Oracle Supply Chain and Order Management Analytics that are located in the \$PMServer\LkpFiles directory (for example, \PowerCenter8.6.x\server\infa_shared\LkpFiles).

Table 1–1 Domain Values and CSV Worksheet Files for Oracle Supply Chain and Order Management Analytics for JD Edwards EnterpriseOne

Worksheet File	Description	Session
file_udc_category_mapping_jde.csv	This file is used to provide JD Edwards EnterpriseOne user defined codes that are to be populated into the W_CODE_D table.	SDE_JDE_Code_Category_Map_Load
domainvalues_MovementTypes_jde.csv	This file provides override information of the Movement Type Description for a specific Movement Type.	SDE_JDE_MvmntTypeDimension
domainvalues_xact_type_codes_scom_jde_sales_ivclns.csv	This file provides the Sales Order Transaction Type Codes that are applicable to sales invoice lines.	SDE_JDE_XactTypeDimension_SalesInvoiceLine
domainvalues_xact_type_codes_scom_jde_sales_ordlns.csv	This file provides the Sales Order Transaction Type Codes that are applicable to Sales Order lines.	SDE_JDE_XactTypeDimension_SalesOrderLine
file_sales_order_status_jde.csv	This file provides Sales Order status information to populate the W_CODE_D (by category) and W_STATUS_D tables.	SDE_JDE_StatusDimension_SalesOrder
file_lkp_chnl_typ_jde.csv	This file serves to provide additional information by Channel Type Code to populate the Channel Type Dimension table.	SDE_JDE_ChannelTypeDimension
file_lkp_consign_inv_org_jde.csv	This file is used to identify the Branch/Plants (Organization ID) that are used for consignment within the inventory process.	SDE_JDE_Inventory_Daily_Bal_Fact
file_lkp_return_loc_jde.csv	This file is used to identify the return locations by Branch/Plant, Location, and Lot Number within the inventory process.	SDE_JDE_Inventory_Daily_Bal_Fact

1.2.2 How to Configure User Defined Code Categories

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

You populate this file with data before running the SDE_JDE_Category_Map_Load workflow. This workflow will run a mapping of the same name that uses the CSV file as input to create a temporary table called W_JDE_UDC_CATEGORY_MAP_TMP. This temporary table is then used as source input into the SDE_JDE_CODE_DIMENSION workflow to load the W_CODE_D table.

To configure this file:

1. Using a text editor, open the file `file_udc_category_mapping_jde.csv` file, located in the `$pmsserver\srcfiles` folder.
2. Compare the values in this file to the values in Table 3-18 in Section 3.5.1.2, "About Configuring the Code Dimension for Oracle's JD Edwards Enterprise One or JD Edwards World UDCs" in the *Oracle BI Applications Configuration Guide for Informatica PowerCenter Users*.
3. Add any new values to the CSV file from the Table 3-18 entries by System Code, User Defined Code, and Category. Ensure that each Category value is spelled correctly, as this Category matches what is defined in the mappings for Oracle Supply Chain and Order Management Analytics for JD Edwards EnterpriseOne.
The values that are added to this CSV file will be added as UDC entries to the `W_CODE_D` table by Category.
For more information about the `W_CODE_D` table and values, see *Oracle Business Analytics Warehouse Data Model Reference*.
4. Use commas to separate the entries under the System Code, UDC column, and Category. Ensure that you do not include any spaces.
5. Save and close the file.

1.2.3 How to Configure Movement Type Domain Values

This section explains how to configure the `domainvalues_Movement_Types_jde.csv`. You populate this flat file with data before running the `SDE_JDE_MvmntTypeDimension` workflow.

To configure this file:

1. Identify the Movement Types in your JD Edwards EnterpriseOne source system by using the following SQL:

```
SELECT RTRIM(DRSY) || '~' || DRRT || '~' ||
LTRIM(DRKY) FROM F0005 AS MVMNT_TYPE_CODE
WHERE DRSY = '00' AND DRRT = 'DT' AND DRKY <> ' '
```

2. Using a text editor, open the `domainvalues_Movement_Types_jde.csv` file, located in the `$PMServer\LkpFiles` directory (for example, `\PowerCenter8.6.x\server\infa_shared\LkpFiles`).
3. Copy the values from the `MVMNT_TYPE_CODE` column of the SQL to the `MVMNT_TYPE_CODE` column in the file.

The data must be copied starting from the second line. The first line is the column header.

4. In the file, for each `MVMNT_TYPE_CODE`, add the correct domain values for `W_MVMNT_TYPE_CODE` and `W_MVMNT_TYPE_DESC`. Valid domain values are `OTHERS`, `CUSTOMER RETURN`, or `VENDOR RETURN`.

All values for both `W_MVMNT_TYPE_CODE` and `W_MVMNT_TYPE_DESC` will be `OTHERS` except for those Document Types (Movement Types) that are actually used for Customer Returns or Vendor Returns. For those specific Document Types identified as such, use those values appropriately.

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

5. Save and close the file.

1.2.4 How to Configure Transaction Types Domain Values for Sales Invoice Lines

This section explains how to configure the domainvalues_xact_type_codes_scom_jde_sales_ivclns.csv. You populate this flat file with data before running the SDE_JDE_XactTypeDimension_SalesInvoiceLine workflow.

To configure this file:

1. Using a text editor, open the domainvalues_xact_type_codes_scom_jde_sales_ivclns.csv file, located in the \$pmsserver\lkpfiles folder.
2. Identify the Invoice Line transaction types in JD Edwards EnterpriseOne by using the following SQL:

```
SELECT F0005.DRKY FROM F0005
WHERE DRSY = '00 ' AND DRRT = 'DT'
```

3. Map the values returned with the flat file to correspond with the XACT_TYPE_CODE source column. Associate the values in this file to the Transaction Type domain values for Invoice Lines as specified in the *Oracle Business Analytics Warehouse Data Model Reference*.

An example configuration is shown here:

```
XACT_TYPE_CODE,XACT_CODE,W_XACT_TYPE_CODE,W_XACT_TYPE_DESC,W_XACT_TYPE_CODE1,W_
XACT_TYPE_DESC1
<DRKY>,SALES_IVCLNS,Standard Invoice,Standard Invoice,Order Management,Order
Management source code
```

Note: The values shown in the example are also the default values for the source code's domain values. If a Document Type <DRKY> is not configured in the flat file, the values presented in this example will be assumed by default for those documents.

For more information about the Transaction Type domain values in W_XACT_TYPE_D, see *Oracle Business Analytics Warehouse Data Model Reference*.

4. Use commas to separate the entries under the System Code, UDC column, and Category. Ensure that you do not include any spaces.
5. Save and close the file.

1.2.5 How to Configure Transaction Types Domain Values for Sales Order Lines

This section explains how to configure the domainvalues_xact_type_codes_scom_jde_sales_ordlns.csv. You populate this flat file with data before running the SDE_JDE_XactTypeDimension_SalesOrderLine workflow.

To configure this file:

1. Using a text editor, open the domainvalues_xact_type_codes_scom_jde_sales_ordlns.csv file, located in the \$pmsserver\lkpfiles folder.
2. Identify the Order Line transaction types in JD Edwards EnterpriseOne by using the following SQL:

```
SELECT F0005.DRKY FROM F0005
WHERE DRSY = '00 ' AND DRRT = 'DT'
```

3. Map the values returned with the flat file to correspond with the XACT_TYPE_CODE source column. Associate the values in this column to the Transaction Type

domain values for Order lines as specified in the *Oracle Business Analytics Warehouse Data Model Reference*.

An example configuration is shown here:

```
XACT_TYPE_CODE, XACT_CODE, W_XACT_TYPE_CODE, W_XACT_TYPE_DESC, W_XACT_TYPE_CODE1, W_XACT_TYPE_DESC1, W_XACT_TYPE_CODE2, W_XACT_TYPE_DESC2
<DRKY>, SALES_ORDLNS, Regular, Regular, EXTERNAL, External order, Self Ship, Internal inventory order
```

Note: The values shown in the example are also the default values for the source code's domain values. If a Document Type <DRKY> is not configured in the flat file, the values presented in this example will be assumed by default for those documents.

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

4. Use commas to separate the entries under the System Code, UDC column, and Category. Ensure that you do not include any spaces.
5. Save and close the file.

1.2.6 How to Configure Sales Order Statuses

This section explains how to configure the file_sales_order_status_jde.csv. You populate this flat file with data before running the SDE_JDE_StatusDimension_SalesOrder workflow. This flat file is the source for the table W_STATUS_DS and is available in the \$pmserver\srcfiles folder.

To configure this file:

1. Using a text editor, open the file_sales_order_status_jde.csv file, located in the \$pmserver\srcfiles folder.
2. Identify the Sales Order status combinations by analyzing your order activity rules setup, which will have data similar to this:

DocType	LineType	Last Status	Description	Next Status	Description
SO	S	520	Enter Sales Order	540	Print Pickslips
SO	S	520	Enter Sales Order	560	Shipment Confirmation
SO	S	520	Enter Sales Order	535	In Warehouse Management
SO	S	900	Backordered in SO Entry	540	Print Pickslips

3. Identify the distinct combination of DocType, LineType, Last Status, and Next Status from the F4211 table for all of your sales orders.
4. Render each of these combinations with a prefix of:

```
SALES_ORDLNS~<DocType>~<LineType>~<LastStatus>~NextStatus>
SALES_PCKLNS~<DocType>~<LineType>~<LastStatus>~NextStatus>
SALES_SCHLNS~<DocType>~<LineType>~<LastStatus>~NextStatus>
```

5. Copy the values rendered into the flat file to correspond with the STATUS_CODE, STATUS_NAME flat file column for each of the STATUS_CLASSES - SALES_ORDLNS, SALES_PCKLNS, and SALES_SCHLNS respectively.

An example configuration is shown here:

```
STATUS_CLASS,STATUS_CODE,STATUS_NAME,STATUS_CAT,STATUS_CAT_DESC
SALES_ORDER_PROCESS,SALES_ORDLNS~SO~S~620~998,SALES_ORDLNS~SO~S~620~998,Being
Processed,Being Processed
SALES_ORDER_PROCESS,SALES_ORDLNS~SO~S~620~999,SALES_
ORDLNS~SO~S~620~999,Closed,Closed
SALES_ORDER_PROCESS,SALES_ORDLNS~SO~S~985~999,SALES_
ORDLNS~SO~S~985~999,Closed,Closed
SALES_PICKING_PROCESS,SALES_PCKLNS~SO~S~520~540,SALES_PCKLNS~SO~S~520~540,Not
Yet Picked,Not Yet Picked
SALES_PICKING_PROCESS,SALES_PCKLNS~SO~S~520~560,SALES_PCKLNS~SO~S~520~560,Not
Yet Picked,Not Yet Picked
```

Note: ■ The code does not automatically identify statuses as ENTERED, BOOKED, BLOCKED, and so forth. These statuses are user-configured in OLTP. These statuses are derived from the setup in the Status Dimension flat file. For example, back-ordered records may be treated as Blocked. The code does not automatically handle this, therefore you should configure the status dimension flat file accordingly to treat back-ordered status combinations as BLOCKED.

- A set of hard-coded statuses are used to identify whether a Sales Order line is cancelled or not. These statuses are 980, 982, 983, 984, 985, 986, 987, 988, 990, 995, 996 and 997. No other Sales Order status is used to identify a cancelled Sales Order line. If you want a certain status combination to be treated as closed, then define it as closed in the Status Dimension flat file. When defined as closed, they will be overlooked by financial and operational backlog flags.
 - You do not have to configure certain statuses that are already hard-coded in the system like Closed, Cancelled, and so forth, as mentioned previously. The code automatically identifies them. However, a hard-coded status line should be present in the flat file to track these statuses, which are described in the following steps.
-

6. Append your configuration with these mandatory hard coded statuses:

```
FULFILLMENT_STATUS,ORDER BOOKED,ORDER BOOKED,Order Booked,Order Booked
FULFILLMENT_STATUS,ORDER ENTERED,ORDER ENTERED,Order Entered,Order Entered
FULFILLMENT_STATUS,ORDER FULLY CANCELLED,ORDER FULLY CANCELLED,Order Fully
Cancelled,Order Fully Cancelled
FULFILLMENT_STATUS,ORDER FULLY PICKED,ORDER FULLY PICKED,Order Fully
Picked,Order Fully Picked
FULFILLMENT_STATUS,ORDER FULLY SCHEDULED,ORDER FULLY SCHEDULED,Order Fully
Scheduled,Order Fully Scheduled
FULFILLMENT_STATUS,ORDER FULLY SHIPPED,ORDER FULLY SHIPPED,Order Fully
Shipped,Order Fully Shipped
FULFILLMENT_STATUS,ORDER PARTIALLY CANCELLED,ORDER PARTIALLY CANCELLED,Order
Partially Cancelled,Order Partially Cancelled
FULFILLMENT_STATUS,ORDER PARTIALLY SCHEDULED,ORDER PARTIALLY SCHEDULED,Order
Partially Scheduled,Order Partially Scheduled
FULFILLMENT_STATUS,ORDER PARTIALLY SHIPPED,ORDER PARTIALLY SHIPPED,Order
```

Partially Shipped,Order Partially Shipped
 FULFILLMENT_STATUS,ORDER NOT ELIGIBLE,ORDER NOT ELIGIBLE,Order Not
 Eligible,Order Not Eligible
 SALES_INVOICE_PROCESS,SALES_INVCLNS~CANCELLED,SALES_
 INVCLNS~CANCELLED,Cancelled,Cancelled
 SALES_INVOICE_PROCESS,SALES_INVCLNS~COMPLETED,SALES_
 INVCLNS~COMPLETED,Completed,Completed
 SALES_INVOICE_PROCESS,SALES_INVCLNS~OPEN,SALES_INVCLNS~OPEN,Open,Open
 SALES_INVOICE_PROCESS,SALES_INVCLNS~PENDING,SALES_
 INVCLNS~PENDING,Pending,Pending
 SALES_ORDER_PROCESS,SALES_ORDLNS~BEING PROCESSED,SALES_ORDLNS~BEING
 PROCESSED,Being Processed,Being Processed
 SALES_ORDER_PROCESS,SALES_ORDLNS~BLOCKED,SALES_ORDLNS~BLOCKED,Blocked,Blocked
 SALES_ORDER_PROCESS,SALES_ORDLNS~BOOKED,SALES_ORDLNS~BOOKED,Booked,Booked
 SALES_ORDER_PROCESS,SALES_ORDLNS~CANCELLED,SALES_
 ORDLNS~CANCELLED,Cancelled,Cancelled
 SALES_ORDER_PROCESS,SALES_ORDLNS~CLOSED,SALES_ORDLNS~CLOSED,Closed,Closed
 SALES_ORDER_PROCESS,SALES_ORDLNS~ENTERED,SALES_ORDLNS~ENTERED,Entered,Entered
 SALES_PICKING_PROCESS,SALES_PCKLNS~CANCELLED,SALES_
 PCKLNS~CANCELLED,Cancelled,Cancelled
 SALES_PICKING_PROCESS,SALES_PCKLNS~FULLY PICKED,SALES_PCKLNS~FULLY PICKED,Fully
 Picked,Fully Picked
 SALES_PICKING_PROCESS,SALES_PCKLNS~FULLY SHIPPED,SALES_PCKLNS~FULLY
 SHIPPED,Fully Shipped,Fully Shipped
 SALES_PICKING_PROCESS,SALES_PCKLNS~NOT RELEVANT,SALES_PCKLNS~NOT RELEVANT,Not
 Relevant,Not Relevant
 SALES_PICKING_PROCESS,SALES_PCKLNS~NOT YET PICKED,SALES_PCKLNS~NOT YET
 PICKED,Not Yet Picked,Not Yet Picked
 SALES_PICKING_PROCESS,SALES_PCKLNS~BACKORDERED,SALES_
 PCKLNS~BACKORDERED,Backordered,Backordered
 SALES_PICKING_PROCESS,SALES_PCKLNS~PURGED,SALES_PCKLNS~PURGED,Purged,Purged
 SALES_SCHEDULE_PROCESS,SALES_SCHLNS~BLOCKED,SALES_
 SCHLNS~BLOCKED,Blocked,Blocked
 SALES_SCHEDULE_PROCESS,SALES_SCHLNS~CANCELLED,SALES_
 SCHLNS~CANCELLED,Cancelled,Cancelled
 SALES_SCHEDULE_PROCESS,SALES_SCHLNS~CLOSED,SALES_SCHLNS~CLOSED,Closed,Closed
 SALES_SCHEDULE_PROCESS,SALES_SCHLNS~ENTERED,SALES_
 SCHLNS~ENTERED,Entered,Entered
 SALES_SCHEDULE_PROCESS,SALES_SCHLNS~NOT VALID,SALES_SCHLNS~NOT VALID,Not
 Valid,Not Valid

7. Save and close the file.

1.2.7 How to Configure Channel Type Groups

This section explains how to configure the file_lkp_chnl_typ_jde.csv. You populate this flat file with data before running the SDE_JDE_ChannelTypeDimension workflow.

To configure this file:

1. Identify the Channel Types in your JD Edwards EnterpriseOne source system by using the following SQL:


```
SELECT DRKY FROM F0005 WHERE DRSY = '90CB' AND DRRT = 'TC'
```
2. Using a text editor, open the file_lkp_chnl_typ_jde.csv file, located in the \$PMServer\LkpFiles directory (for example, \PowerCenter8.6.x\server\infa_shared\LkpFiles).
3. Copy the values from the DRKY column of the SQL to the CHNL_TYPE_CODE column in the file.

The data must be copied starting from the second line. The first line is the column header.

4. In the file, add the correct domain values for Channel Type Group Code (W_CHTY_GRP_CODE), Channel Type Subgroup Code (W_CHTY_SUBG_CODE), and W_INBOUND_TYPE_FLG for each Channel Type Code (CHNL_TYPE_CODE).

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

5. Save and Close the file.

1.2.8 How to Configure Inventory Consignments

This section explains how to configure the file_lkp_consign_inv_org_jde.csv. You populate this flat file with data before running the SDE_JDE_Inventory_Daily_Bal_Fact workflow.

To configure this file:

1. Identify the Inventory Consignment Branch/Plants (Warehouse and Inventory Organization IDs) in your JD Edwards EnterpriseOne source system.
2. Using a text editor, open the file_lkp_consign_inv_org_jde.csv file, located in the \$PMSEServer\LkpFiles directory (for example, \PowerCenter8.6.x\server\infa_shared\LkpFiles).
3. Manually enter the first value for a valid Inventory Consignment Branch/Plant on the second line in the file. The first line is the column header.

Note: The data for each Branch/Plant must be entered as 12 characters and right-justified with leading blanks. For example, Branch/Plant (INVENTORY_ORG_ID) ABC should be entered as -----ABC with nine leading blanks.

4. Enter each Inventory Consignment Branch/Plant on a separate line thereafter.
For more information on consignments, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

1.2.9 How to Configure Inventory Return Locations

This section explains how to configure the file_lkp_return_loc_jde.csv. You populate this flat file with data before running the SDE_JDE_Inventory_Daily_Bal_Fact workflow.

To configure this file:

1. Identify the Inventory Return Locations in each Branch/Plant (Warehouse/Inventory Organization ID) in your JD Edwards EnterpriseOne source system.
2. Using a text editor, open the file_lkp_return_loc_jde.csv file, located in the \$PMSEServer\LkpFiles directory (for example, \PowerCenter8.6.x\server\infa_shared\LkpFiles).
3. Manually enter the first value for a valid Inventory Return Location on the second line in the file. The first line is the column header.

Note: The data for each Return Location must be entered in the following format on each line:

STORAGE_LOC~Branch/Plant~Location~Lot Number

where:

- STORAGE_LOC~ is a constant
- Branch/Plant value is 12 characters and right-justified, blank filled
- Location is left-justified and 20 characters right-blank filled
- Lot Number is entered with no trailing blanks

For example, an Inventory Return Location with a Branch/Plant of ABC, a Location of 123 and a Lot Number of 789, would be entered in the following format:

STORAGE_LOC~ ABC~123 ~789

4. Enter each Inventory Return Location on a separate line thereafter.

For more information on return locations, see *Oracle Business Analytics Warehouse Data Model Reference*.

5. Save and close the file.

1.2.10 How to Process Bill of Material Explosion

This section explains how to process the Bill of Materials (BOM) for exploding to a multi-level structure to ultimately populate both the W_BOM_HEADER_D and W_BOM_ITEM_F tables.

JD Edwards EnterpriseOne maintains BOM information in a single level format, but Oracle BI Applications requires it in multi-level format. Therefore, before loading data into Oracle BI Application tables, the single level structure needs to be exploded into a multi-level structure.

Because all of the BOM information is stored in one single table in JD Edwards EnterpriseOne source and there are no defined levels for the BOM, the system has to loop through iteratively to get the BOM exploded. Also, Oracle BI Applications maintains all the revisions to the components as a new version of the BOM along with their effective dates. Considering these facts, it is not feasible to use ETL to convert the single level BOM to a multi-level BOM. Therefore the logic from an existing JD Edwards EnterpriseOne UBE (R30460) was used to create a new UBE for the explosion.

This new UBE (R30461) extracts the manufactured end products and converts the single-level BOM format into a multi-level BOM format. In addition, it also extracts some required information like Left bounds/Right bounds and level parents (1 - 10).

The UBE loads the multi-level BOM structure for manufactured end products with each revision into two work files respectively for BOM header and item (component). The ETL then extracts the data from the two work files and loads it into the Oracle BI Applications tables.

Note: If you plan to consume analytics on Bill of Materials, it is mandatory to run this UBE before starting the ETL. This UBE and the related JD Edwards EnterpriseOne objects are created solely for the benefit of analytics and therefore will not be available in the existing source system.

1.2.11 How to Configure the Left Bound and Right Bound Calculation Option

The SIL_BOMItemFact mapping contains the stored procedure called COMPUTE_BOUNDS which traverses the exploded BOM tree structure and calculates the left bound and right bound. By default, the COMPUTE_BOUNDS stored procedure is off.

In JD Edwards EnterpriseOne, the left and right bounds are calculated by the UBE (R30461).

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 3.1.5, "How to Deploy Stored Procedures" in the *Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users*.

Support for JD Edwards EnterpriseOne Ledger Types in Oracle Financial Analytics

This chapter contains the following topics:

- [Section 2.1, "Ledger Type Mapping"](#)
- [Section 2.2, "Dashboard Page Prompts"](#)
- [Section 2.3, "Configuring Ledger Type"](#)

2.1 Ledger Type Mapping

Oracle BI Applications release 7.9.6.2 extracts JD Edwards EnterpriseOne 'Ledger Type' and maps it into the Ledger Dimension (W_LEDGER_D). This mapping provides:

- The ability to filter or segregate data by Ledger Type for reporting and analysis.
- New support for consolidation ('AC') and restatement ledgers ('XA'); in addition, to already supported Actuals and Budget ledgers.

Release 7.9.6.1 Supported Ledgers	Release 7.9.6.2 Supported Ledgers	Ledgers Not Supported
<ul style="list-style-type: none"> ■ Actuals, such as 'AA' and 'A1' ■ Budget, such as 'BA' and 'B1' 	<ul style="list-style-type: none"> ■ Consolidation - 'AC' ■ Restatement, such as 'XA' and 'YA' 	<ul style="list-style-type: none"> ■ Transaction currency - 'CA' ■ Units - '*U' ■ Percent of Job complete - 'F%'

[Table 2–1](#) and [Table 2–2](#) provide an example of the difference in how records are populated into the Ledger Dimension prior to an after introducing support for Ledger Type. Note the changes for the following columns in release 7.9.6.2:

- LEDGER_NAME contains the concatenation of Company and Ledger Type. There will be as many entries for company as there are Ledger Types. For example, assuming there are two Ledger Types (AA and A1) and two Companies (00001 and 00050), LEDGER_NAME will contain the four rows shown in the table once the load process is complete.
- LEDGER_CATEGORY_CODE contains Ledger Type.
- LEDGER_CATEGORY_NAME contains Ledger Type Name.

Table 2–1 Example of Ledger Dimension Table Prior to Support for Ledger Type

LEDGER_NAME	LEDGER_DESC	CURRENCY_CODE	LEDGER_CATEGORY_CODE	LEDGER_CATEGORY_NAME
00001	Fixed Asset Company #2	USD	Null	Null
00050	SAR TESTING COMPANY	USD	Null	Null

Table 2–2 Example of Release 7.9.6.2 Ledger Dimension Table with Support for Ledger Type

LEDGER_NAME	LEDGER_DESC	CURRENCY_CODE	LEDGER_CATEGORY_CODE	LEDGER_CATEGORY_NAME
00001~AA	Fixed Asset Company #2	USD	AA	Actuals
00001~A1	Fixed Asset Company #2	USD	A1	User Defined Actuals
00050~AA	SAR TESTING COMPANY	USD	AA	Actuals
00050~A1	SAR TESTING COMPANY	USD	A1	User Defined Actuals

2.2 Dashboard Page Prompts

Dashboard prompts defined during the Financial Analytics configuration process will not require any additional changes because Ledger Name will display the new concatenated Ledger Name ("Company~Ledger Type") values.

Note: LEDGER_CATEGORY_NAME and LEDGER_CATEGORY_CODE can be exposed as Dashboard prompts with a few modifications to the original product configuration:

- LEDGER_CATEGORY_NAME is already available in the Presentation Subject Area as part of the Ledger Dimension. Therefore, all that is needed is to define LEDGER_CATEGORY_NAME as a prompt for the required Dashboard page or pages.
- LEDGER_CATEGORY_CODE is available in the RPD Business Model and Mapping (BMM) layer. Therefore, before defining as a Dashboard Page prompt, you will need to expose within the RPD Presentation Subject Area, as a member of the Ledger Dimension.

The following table summarizes the availability of each column across each area after the standard product configuration and associated action needed to define as a Dashboard page prompt.

Ledger Dimension Column	Dashboard Prompt	Presentation Subject Area	RPD BMM Layer	Action to Define as Dashboard Page Prompt
LEDGER_NAME	Yes	Yes	Yes	None. Already available as a Dashboard prompt.

Ledger Dimension Column	Dashboard Prompt	Presentation Subject Area	RPD BMM Layer	Action to Define as Dashboard Page Prompt
LEDGER_CATEGORY_NAME	No	Yes	Yes	Define as prompt in Dashboard Page.
LEDGER_CATEGORY_CODE	No	No	Yes	<ol style="list-style-type: none"> 1. Expose in Presentation Subject Area. 2. Define as prompt in Dashboard Page.

2.3 Configuring Ledger Type

The ETL map is run by passing the Ledger Type value through DAC as comma separated values. By default, the value for Ledger Type is AA.

To run the ETL map for the new supported Ledger types, consolidation ('AC') and restatement ledgers ('XA'), the DAC Source System Parameter \$\$LEDGER_TYPES should include the required Ledger Types as comma separated values: 'AA', 'XA', 'AC'.

For more detailed instructions on running the ETL maps, refer to the Oracle® Business Intelligence Applications documentation.

PeopleSoft Chartfields for Oracle BI Applications

This chapter describes the new PeopleSoft chartfields for Oracle BI Applications.

3.1 New PeopleSoft Chartfields

In PeopleSoft Enterprise applications, the fields that store your chart of accounts and provide your system with the basic structure to segregate and categorize transactional and budget data are called chartfields. With the 7.9.6.2 release, Oracle BI Applications is increasing the number of supported chartfields for PeopleSoft Enterprise.

The following table shows the chartfields available before release 7.9.6.2 and the chartfields added to release 7.9.6.2:

Pre- 7.9.6.2 Release Chartfields	New 7.9.6.2 Chartfields
<ul style="list-style-type: none"> ▪ Account ▪ Alternate Account ▪ Operating Unit ▪ Fund Code ▪ Department ▪ Program Code ▪ Class Field ▪ Budget Reference ▪ Product ▪ Project ▪ Affiliate ▪ Fund Affiliate ▪ Operating Unit Affiliate ▪ ChartField 1 ▪ ChartField 2 ▪ ChartField 3 ▪ Statistics Code 	<ul style="list-style-type: none"> ▪ PC Business Unit ▪ Activity ID ▪ Analysis Type ▪ Resource Type ▪ Resource Category ▪ Resource Sub Category

3.1.1 Configuring the New Chartfields

To use the six new Projects-related chartfields, the DAC parameter `$$INCLUDE_PROJ_CHARTFIELDS` must be set to 'Y' and a full ETL needs to be run.

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

This parameter is located in the Source System Parameters tab in the PeopleSoft container.

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

Enhanced Support for PeopleSoft Product/Purchasing Categories

This chapter is for customers using Oracle Procurement and Spend Analytics with a PeopleSoft source system. In Oracle BI Applications Release 7.9.6.2, Procurement and Spend Analytics supports reporting by Product/Purchasing Category hierarchy for organizations that use Product/Purchasing Category to create requisition and purchase orders.

This chapter contains the following topics:

- [Section 4.1, "About Product/Purchasing Categories in PeopleSoft"](#)
- [Section 4.2, "Configuration Required to Enable Reporting on Product/Purchasing Categories"](#)

4.1 About Product/Purchasing Categories in PeopleSoft

In PeopleSoft Enterprise applications, Product/Purchasing Category hierarchies are stored in trees. A PeopleSoft Enterprise application user chooses a tree and then assigns a category from that tree to an item using the Create Requisition option from the eProcurement menu.

4.2 Configuration Required to Enable Reporting on Product/Purchasing Categories

The default value of TREE_NAME in the PeopleSoft source system is ALL_PURCHASE_ITEMS. If your organization has not changed the value of TREE_NAME to another value, and you want to report by this TREE_NAME, then no additional configuration is required to report on Product/Purchasing Category hierarchies.

If your organization has changed the default value of TREE_NAME, then you need to perform the following tasks described in this section.

Use the following SQL query to extract the tree names from the PeopleSoft Enterprise application to confirm the value of TREE_NAME:

```
SELECT * FROM PSTREEDEFN WHERE TREE_STRCT_ID = 'ITEMS'
```

To enable business intelligence reporting on Product/Purchasing Categories, use DAC to configure the following tasks:

- SDE_PSFT_ProductDimension_ItemCategory
- SDE_PSFT_ProductDimension_ItemMaster

In both of these tasks, set the value of the `$$TREE_NAME1` variable to the tree name that you want to use for reporting, and then save your changes.

Support for Currency Balance Types in Oracle Financial Analytics for Oracle EBS and Oracle's PeopleSoft

This chapter contains the following topics:

- [Section 5.1, "Oracle EBS Currency Balance Types"](#)
- [Section 5.2, "PeopleSoft Currency Balance Types"](#)

5.1 Oracle EBS Currency Balance Types

The ETL process in 7.9.6.2 extracts four different currency balance types from Oracle EBS, namely base currency, translated currency, entered currency, and statistical balance. These records are loaded and stored in W_GL_BALANCE_F table in the Oracle Data Warehouse.

The following table shows sample records of the different balance types. All four records are from ledger 'US Ledger' with U.S. dollars (USD) as the base ledger currency. These five columns in the following table are columns in the physical data warehouse table:

- LOC_CURR_CODE
- ACCT_CURR_CODE
- TRANSLATED_FLAG
- BALANCE_LOC_AMT
- BALANCE_ACCT_AMT

Line	Ledger Name	LOC_CURR_CODE	ACCT_CURR_CODE	TRANSLATED_FLAG	BALANCE_LOC_AMT	BALANCE_ACCT_AMT	Balance Type Description
1	US Ledger	USD	NULL	N	200	0	A Base Currency Balance record with base ledger currency of USD
2	US Ledger	USD	GBP	Y	0	100	A Translated Currency Balance record with translation currency of GBP (British pounds). USD is the base ledger currency.

Line	Ledger Name	LOC_CURR_CODE	ACCT_CURR_CODE	TRANSLATED_FLAG	BALANCE_LOC_AMT	BALANCE_ACCT_AMT	Balance Type Description
3	US Ledger	USD	JPY	N	0	10000	An Entered Currency Balance record with entered currency in JPY (USD is the base ledger currency)
4	US Ledger	USD	STAT	N	0	150	A Statistical balance amount

Users can query these balances by using different value columns and filters. The following tables contain sample queries that you can use to view different currency balance types and their corresponding results from the sample warehouse data in the preceding table.

Note: Ledger 'US Ledger' corresponds to LEDGER_WID = 85228 in the following sample scenarios.

Sample	Desired Queries	Physical Table Value Column	Physical Query Filters	Result for Value Column
A	To query: balance amount for base currency of USD, in ledger 'US Ledger'	BALANCE_LOC_AMT	LEDGER_WID = 85228	200
B	To query: translated balance amount for translated currency of GBP, in ledger 'US Ledger'	BALANCE_ACCT_AMT	LEDGER_WID = 85228 AND TRANSLATED_FLAG = 'Y' AND ACCT_CURR_CODE = 'GBP'	100
C	To query: entered balance amount for entered currency of JPY, in ledger 'US Ledger'	BALANCE_ACCT_AMT	LEDGER_WID = 85228 AND ACCT_CURR_CODE = 'JPY' AND TRANSLATED_FLAG = 'N'	10000
D	To query: statistical balance amounts, in ledger 'US Ledger'	BALANCE_ACCT_AMT	LEDGER_WID = 85228 AND ACCT_CURR_CODE = 'STAT'	150

Oracle BI Answers users can create reports on currency balance types by selecting the desired facts from the Financials - GL Balance Sheet presentation table, as shown in the following table.

Sample Scenario	Desired Report	Presentation Table	Presentation Column	Result in Report
A	View the local currency balance amounts (Debit Local Amount), such as in sample query A in the preceding table	Ledger Facts – GL Balance	Ledger name Debit Local Amount	Apply filter on: "Ledger Name" = 'US Ledger' Ledger Name = US Ledger Debit Local Amount = 200
B	View the translated currency balance amounts (Debit Translated Amount), such as in sample query B in the preceding table	Ledger Facts – GL Balance GL Balance Details	Ledger Name Debit Translated Amount Translated Currency Code	Apply filters on: "Ledger Name" = 'US Ledger' "Translated Currency Code" = 'GBP' Ledger Name = US Leger Debit Translated Amount = 100 Translated Currency Code = GBP
C	View the entered currency balance amounts (Debit Entered Amount), such as in sample query C in the preceding table	Ledger Facts – GL Balance GL Balance Details	Ledger Name Debit Entered Amount Entered Currency Code	Apply filters on: "Ledger Name" = 'US Ledger' "Entered Currency Code" = 'JPY' Ledger Name = US Leger Debit Entered Amount = 10000 Entered Currency Code = JPY
D	View the statistical amount (Statistical Amount), such as in sample query D in the preceding table	Ledger Facts - Balance Sheet Statement	Ledger Name Statistical Amount	Ledger Name = US Leger Statistical Amount = 150

5.2 PeopleSoft Currency Balance Types

The ETL process in release 7.9.6.2 extracts four different currency balance types from PeopleSoft, namely base currency, translated currency, entered currency and statistical balance. These records are loaded and stored in the warehouse table W_GL_BALANCE_F. The following table shows some sample records of different balance types. The first four records are from Primary ledger 'Euro Ledger' with EUR as the base ledger currency. The next three records are from Translation ledger 'US Ledger' with USD as the base ledger currency. These five columns in the following table are columns in the physical data warehouse table:

- LOC_CURR_CODE
- ACCT_CURR_CODE
- TRANSLATED_FLAG
- BALANCE_LOC_AMT
- BALANCE_ACCT_AMT

Line	Ledger Name	LOC_CURR_CODE	ACCT_CURR_CODE	TRANSLATED_FLAG	BALANCE_LOC_AMT	BALANCE_ACCT_AMT	Balance Type Description
1	Euro Ledger (Primary ledger)	EUR	EUR	N	390	50	A Base Currency Balance record with base ledger currency of EUR
2	Euro Ledger (Primary ledger)	EUR	USD	N	0	200	An Entered Currency Balance record with entered currency in USD (EUR is the base ledger currency)
3	Euro Ledger (Primary ledger)	EUR	STAT	N	0	80	A Statistical balance amount
4	US Ledger (Translation ledger)	USD	USD	Y	340	340	A Translated Currency Balance record with translation currency of USD Note: US Ledger is a Translation ledger
5	US Ledger (Translation ledger)	USD	EUR	N	0	50	An Entered Currency Balance record with entered currency in EUR (USD is the base ledger currency) Note: US Ledger is a Translation ledger

Users can query these balances by using different value columns and filters. The following table contains sample queries to show how to view different currency balance types and their corresponding results from the sample warehouse data in the preceding table.

Note: Primary ledger 'Euro Ledger' corresponds to LEDGER_WID = 85228 and Translation ledger 'US Ledger' corresponds to LEDGER_WID = 87320 in the following sample scenarios.

Sample	Desired Queries	Physical Table Value Column	Physical Query Filters	Result for Value Column
A	To query: balance amount for base currency of EUR, in Primary ledger 'Euro Ledger'	BALANCE_LOC_AMT	LEDGER_WID = 85228	390
B	To query: entered balance amount for entered currency of EUR in Primary ledger 'Euro Ledger'	BALANCE_ACCT_AMT	LEDGER_WID = 85228 AND ACCT_CURR_CODE = 'EUR' AND TRANSLATED_FLAG = 'N'	50

Sample	Desired Queries	Physical Table Value Column	Physical Query Filters	Result for Value Column
C	To query: entered balance amount for entered currency of USD, in Primary ledger 'Euro Ledger'	BALANCE_ACCT_ AMT	LEDGER_WID = 85228 AND ACCT_CURR_CODE = 'USD' AND TRANSLATED_FLAG = 'N'	200
D	To query: statistical balance amount, in Primary ledger 'Euro Ledger'	BALANCE_ACCT_ AMT	LEDGER_WID = 85228 AND ACCT_CURR_CODE = 'STAT'	80
E	To query: translated balance account amount for translated currency of USD, in Translation ledger 'US Ledger'	BALANCE_ACCT_ AMT	LEDGER_WID = 87320 AND TRANSLATED_FLAG = 'Y'	340
F	To query: entered balance amount for entered currency of EUR, in Translation ledger 'US Ledger'	BALANCE_ACCT_ AMT	LEDGER_WID = 87320 AND ACCT_CURR_CODE = 'EUR' AND TRANSLATED_FLAG = 'N'	50

Oracle BI Answers users can create reports on currency balance types by selecting the desired facts from the Financials – GL Balance Sheet presentation table, as shown in the following table.

Sample Scenario	Desired Report	Presentation Table	Presentation Column	Result in Report
A	View the local currency balance amounts (Debit Local Amount), such as in sample query A in the preceding table	Ledger Facts – GL Balance	Ledger Name Debit Local Amount	Apply filter on: "Ledger Name" = 'Euro Ledger' Ledger Name = Euro Ledger Debit Local Amount = 390
B	View the local currency balance amounts (Debit Amount), such as in sample query B in the preceding table	Ledger Facts – GL Balance	Ledger Name Debit Amount	Apply filter on: "Ledger Name" = 'Euro Ledger' Ledger Name = Euro Ledger Debit Amount = 50

Sample Scenario	Desired Report	Presentation Table	Presentation Column	Result in Report
C	View the entered currency balance amounts (Debit Entered Amount), such as in sample query C in the preceding table	Ledger Facts – GL Balance GL Balance Details	Ledger Name Debit Entered Amount Entered Currency Code	Apply filters on: "Ledger Name" = 'Euro Ledger' Entered Currency Code" = 'USD' Ledger Name = Euro Ledger Debit Entered Amount = 200 Entered Currency Code = USD
D	View the statistical amount (Statistical Amount), such as in sample query D in the preceding table	Ledger Facts – Balance Sheet Statement	Ledger Name Statistical Amount	Ledger Name = Euro Ledger Statistical Amount = 80
E	View the translated currency balance amounts (Debit Translated Amount), such as in sample query E in the preceding table	Ledger Facts – GL Balance GL Balance Details	Ledger Name Debit Translated Amount Translated Currency Code	Apply filters on: "Ledger Name" = 'US Ledger' "Translated Currency Code" = 'USD' Ledger Name = US Ledger Debit Translated Amount = 340 Translated Currency Code = USD
F	If we want to view the Entered currency balance amounts (Debit Entered Amount), such as in sample query F in the preceding table	Ledger Facts – GL Balance GL Balance Details	Ledger Name Debit Entered Amount Entered Currency Code	Apply filters on: "Ledger Name" = 'US Ledger' "Entered Currency Code" = 'EUR' Ledger Name = US Ledger Debit Entered Amount = 50 Entered Currency Code = EUR

New Features and Updates for Oracle Project Analytics

This chapter contains the following topics:

- [Section 6.1, "New Project Commitments Subject Area for Oracle EBS"](#)
- [Section 6.2, "Updates to Universal Adapter for Oracle Project Analytics"](#)
- [Section 6.3, "Configuration Update for Project Customer for Oracle EBS"](#)
- [Section 6.4, "My Oracle Support Technical Note on How to Configure RPD if the Cost Aggregate Grain is Set to Period or Year"](#)
- [Section 6.5, "New Dimension for Project Budget and Forecast Fact"](#)
- [Section 6.6, "Linear Spread Metrics for Project Budget Fact"](#)

6.1 New Project Commitments Subject Area for Oracle EBS

This section contains the following topics:

- [Section 6.1.1, "Overview"](#)
- [Section 6.1.2, "Before You Begin"](#)
- [Section 6.1.3, "Configuration Steps for the Project Commitments Subject Area"](#)

6.1.1 Overview

The following new features have been added to Oracle Project Analytics for Oracle EBS:

- Project Commitments subject area
- Commitments metrics to the Project Performance subject area

The Project Commitments subject area provides the ability to report on project commitments, which includes total raw and burdened amounts for requisitions, purchase orders, and supplier invoices for organizations, projects, tasks, resources, suppliers, and associated hierarchies. It provides the ability to track commitments at the commitment document level.

To support the new subject area, there is a new star schema in the Oracle Business Analytics Warehouse. This star contains metrics to report on total commitments and its components, which includes quantity and amount (raw and burdened) for requisitions, purchase orders, and supplier invoices.

The W_PROJ_COMMITMENT_F fact table at the center of the star schema stores the latest commitment data, sourced from the transactional source PA_COMMITMENT_TXNS.

Existing commitment reports have been redesigned to use the new metrics and dimensions.

6.1.2 Before You Begin

Before you follow the configuration steps in this chapter, perform the source-independent configuration steps in the following chapters of the *Oracle Business Intelligence Analytics Configuration Guide for Informatica PowerCenter Users*:

- "Common Areas and Dimensions"
- "Configuring Oracle Project Analytics"

6.1.3 Configuration Steps for the Project Commitments Subject Area

To use the new project commitments subject area, you need to configure the domainValues_Project_CommitmentType_ora<ver>.csv file with the commitment types that your organization uses in the Oracle EBS implementation.

Table 6–1 describes the CSV worksheet file and domain values for the Project Commitments subject area.

Table 6–1 CSV Worksheet File and Domain Values for the Project Commitments Subject Area

Domain Values	Domain Value Table Column	Description	Session
domainValues_Project_CommitmentType_ora<ver>.csv	W_PROJ_COMMITMENT_F.W_COMMITMENT_TYPE	Lists the Commitment Type codes and their corresponding domain values of Commitment Type for the Oracle 11i/12 Application	SDE_ORA_ProjectCommitmentFact

To configure domainValues_Project_CommitmentType_ora<ver>.csv:

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

```
SELECT LOOKUP_CODE FROM FND_LOOKUP_VALUES WHERE LOOKUP_TYPE = 'COMMITMENT LINE TYPE'
```

2. Using a text editor, open the domainValues_Project_CommitmentType_ora<ver>.csv file located in the \$pmsserver\lkpfiles folder.
3. Copy the LOOKUP_CODE to the COMMITMENT_TYPE column in the file. The data must be copied starting from the 6th line. Use commas to separate the entries.
4. Map each Lookup Code (COMMITMENT_TYPE) to one domain value and a corresponding meaning. For more information about Project Commitment Type domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

6.2 Updates to Universal Adapter for Oracle Project Analytics

Refer to the following guides before performing the configuration described in this section:

- *Oracle BI Applications Configuration Guide for Informatica PowerCenter Users*, Section 15.2.4 "Configuration Steps for Project Analytics for Universal"
- *Oracle Business Analytics Warehouse Data Model Reference*

In addition to the CSV files mentioned in the configuration documentation, the following CSV files also need to be configured so that the code dimension is loaded properly and the code – name pairs are properly resolved in the dimension tables:

- file_codes_project_bill_hold.csv
- file_codes_project_budget_changereason.csv
- file_codes_project_budget_class.csv
- file_codes_project_budget_entrylevel.csv
- file_codes_project_budget_status.csv
- file_codes_project_budget_timephasedtype.csv
- file_codes_project_budget_versiontype.csv
- file_codes_project_budgettype.csv
- file_codes_project_contract_type.csv
- file_codes_project_cost_code_type.csv
- file_codes_project_event_class.csv
- file_codes_project_event_source_code.csv
- file_codes_project_expenditure_category.csv
- file_codes_project_expenditure_class.csv
- file_codes_project_finplantype.csv
- file_codes_project_funding_category.csv
- file_codes_project_fundingapproval.csv
- file_codes_project_invoice_class.csv
- file_codes_project_invoice_line_type.csv
- file_codes_project_invoice_status.csv
- file_codes_project_priority_code.csv
- file_codes_project_projecttype.csv
- file_codes_project_resource_class.csv
- file_codes_project_resource_type.csv
- file_codes_project_resource_type_code.csv
- file_codes_project_revenue_category.csv
- file_codes_project_revenue_distributed.csv
- file_codes_project_revenue_status.csv
- file_codes_project_role.csv

- file_codes_project_securitylevel.csv
- file_codes_project_service_type.csv
- file_codes_project_status.csv
- file_codes_project_task_priority_code.csv
- file_codes_project_task_status.csv
- file_codes_project_task_type.csv
- file_codes_project_transfer_status.csv
- file_codes_project_type_class.csv
- file_codes_project_uom.csv

As a general rule, use the default value 0 for numeric columns and 'N/A' or 'Unspecified' for string columns so that you do not encounter 'Not Null' errors when running the ETLs. Date fields can be null as well.

The dimension ID fields in the fact staging tables have to be populated with the integration_id of the various dimensions. This is important, otherwise the dimension wid fields in the fact tables will default to 0.

Use the following SQL to retrieve the foreign key information for project facts from the DAC metadata. In addition, refer to the *Oracle Business Analytics Warehouse Date Model Reference* for the star schema diagrams and other foreign key information:

```
select a.name tbl, b.name col, aa.name fk_tbl, bb.name fk_col
from W_ETL_TABLE a, W_ETL_TABLE_COL b,W_ETL_TABLE aa, W_ETL_TABLE_COL bb
where a.row_wid = b.table_wid
and b.fk_table_wid = aa.row_wid
and b.fk_col_wid = bb.row_wid
and a.name like 'W_PROJ%F'
and b.name like '%WID'
AND B.NAME NOT IN ('ETL_PROC_WID')
order by 1,2;
```

Similarly, the common dimensions that Projects uses such as W_INT_ORG_D, W_MCAL_DAY_D, W_MCAL_CONTEXT_G, W_EMPLOYE_D, W_JOB_D, W_INVENTORY_PRODUCT_D, and so forth also need to be populated correctly from the source files.

W_MCAL_CONTEXT_G has a class field that holds two values—GL or PROJECTS. To resolve the project accounting dates in the fact tables, there must be data present in this table for class 'PROJECTS.'

6.3 Configuration Update for Project Customer for Oracle EBS

By default, Oracle EBS only has 'PRIMARY' relationship code in the PA_PROJECT_CUSTOMERS table. Therefore, the value has been added in the Informatica lookup used in the source extract mapping for the Project dimension to get the customer for a project. Customers can define an additional value such as 'OVERRIDE CUSTOMER' as the relationship value. In this case, the lookup needs to be edited to include any additional values.

To edit the lookup:

1. Open the PowerCenter Designer and connect to your Informatica repository where you want to effect these changes.

2. Open the SDE_ORA_11510_Adaptor or SDE_ORA_R12_Adaptor folder of the Informatica repository.
3. Open the mapping SDE_ORA_ProjectDimension and open the mplt_SA_ORA_ProjectDimension maplet in it.
4. Check out the mplt_SA_ORA_ProjectDimension maplet.
5. Open the lookup LKP_PROJ_CUST, select the Properties tab, and then open the SQL query.
6. Remove the existing SQL and add the following sample SQL where it is assumed the values are 'PRIMARY' and 'OVERRIDE CUSTOMER'. Modify it according to your configuration. If you want it to be independent of any relationships, then just remove the filters on PROJECT_RELATIONSHIP_CODE.

```
SELECT MAX(PA_PROJECT_CUSTOMERS.CUSTOMER_ID) AS CUSTOMER_ID,
PA_PROJECT_CUSTOMERS.PROJECT_ID           AS PROJECT_ID
FROM PA_PROJECT_CUSTOMERS
WHERE UPPER(PA_PROJECT_CUSTOMERS.PROJECT_RELATIONSHIP_CODE) in
('PRIMARY', 'OVERRIDE CUSTOMER')
AND PA_PROJECT_CUSTOMERS.CUSTOMER_BILL_SPLIT =
(SELECT MAX(CUSTOMER_BILL_SPLIT)
FROM PA_PROJECT_CUSTOMERS A
WHERE A.PROJECT_ID = PA_PROJECT_CUSTOMERS.PROJECT_ID
and UPPER(a.PROJECT_RELATIONSHIP_CODE) in ('PRIMARY', 'OVERRIDE CUSTOMER')
)
GROUP BY PA_PROJECT_CUSTOMERS.PROJECT_ID
```

7. Validate the lookup SQL and press OK twice, and then save the repository and check in the maplet. Review the mapping to ensure it is valid.
8. Open PowerCenter Workflow Manager and connect to one of these repository folders: SDE_ORA_11510_Adaptor or SDE_ORA_R12_Adaptor.
 - a. Check out the SDE_ORA_Project session in the Task Developer tab.
 - b. Validate, save, and check in the session.
 - c. Save your repository.

6.4 My Oracle Support Technical Note on How to Configure RPD if the Cost Aggregate Grain is Set to Period or Year

A technical note has been posted to My Oracle Support to explain how to configure the grain of Cost aggregate (W_PROJ_COST_A) to Fiscal Period or Fiscal Quarter or Fiscal Year.

To locate the note in My Oracle Support, enter 1088171.1 in the search field.

6.5 New Dimension for Project Budget and Forecast Fact

The Expenditure Category dimension was added to Oracle Project Analytics. This new dimension is an addition to the dimensions for Project Budget and Forecast Fact listed in the *Oracle Business Analytics Warehouse Data Model Reference*.

6.6 Linear Spread Metrics for Project Budget Fact

In the Oracle EBS Project application, the Budget Line amount is split equally between all the days (including weekend days) in the budget line duration when calculating the Cost Amount, Cost ITD Amount, or Cost PTD Amount for a certain Period. By default,

Oracle Project Analytics allocates the full Budget Line Amount to the Period containing the Budget Line Start Date. All the necessary metadata is provided in the RPD to support the calculation method adopted by the Oracle Project application that uses Linear Spreading to calculate PTD and ITD metrics.

A new logical table, Fact – Project Budget Linear Spread, has been created that contains Budget metrics built using a Linear Spreading logic in the business model and mapping layer. However, the new metrics are not exposed in the presentation layer. You can either replace the existing Budget metrics in presentation layer with the new ones (the same metric names are used in the old and new logical Budget fact tables), or create a new subject area to expose the new metrics. The first option enables you to reuse the existing reports with Budget metrics without any modification.

Support for Global 1 Amount Exchange Date Calculation

This chapter describes how to configure the Global 1 Amount Exchange Date calculation in Oracle Financial Analytics for JD Edwards EnterpriseOne. It contains the following topics:

- [Section 7.1, "Configuring the Exchange Date for JD Edwards EnterpriseOne Accounts Payable"](#)
- [Section 7.2, "Exchange Date for JD Edwards EnterpriseOne Account Receivables"](#)

7.1 Configuring the Exchange Date for JD Edwards EnterpriseOne Accounts Payable

JD Edwards EnterpriseOne customers using Oracle Financial Analytics can now choose either the invoice date or the general ledger date to populate the field EXCHANGE_DT in Accounts Payable (AP).

This release includes a new Oracle Business Intelligence Data Warehouse Administration Console (DAC) parameter, \$\$EXCHANGE_DATE, that you can use to specify the invoice date or the general ledger date. Based on your selection, the EXCHANGE_DT field in the W_AP_XACT_FS table is populated with the appropriate date in the Oracle Business Analytics Warehouse.

To configure the \$\$EXCHANGE_DT parameter:

1. Make sure to configure the invoice date or general ledger date the same as it is in JD Edwards EnterpriseOne so that your results are the same as the JD Edwards EnterpriseOne source system values.
2. In the DAC client, select the Design view, and then select your custom container from the drop-down list.
3. Click the Tasks tab, and perform a query for the AP transaction fact-related tasks.
4. In the Detail area of the selected tasks, click the Parameters tab.
5. Modify the parameter value as follows:
 - Specify INV to use the invoice date.
 - Specify GL to use the general ledger date.
6. Save your changes.

7.2 Exchange Date for JD Edwards EnterpriseOne Account Receivables

The EXCHANGE_DT field in W_AR_XACT_FS will be populated from the F03B11-RPERDJ table. The JD Edwards EnterpriseOne application stores the exchange date in the F03B11 table, which contains the value selected by the user through the processing option.

Support for Back-Dated Changes to Position Hierarchy

This chapter contains the following topics:

- [Overview](#)
- [Supported Back-Dated Change Scenarios](#)
- [Objects Changed to Support Back-Dated Changes in Supervisor/Position Hierarchy](#)

8.1 Overview

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

In previous releases, Position Hierarchy was not designed to support back-dated changes to an employee's supervisor, organization, or other information to support deletes. If there is a correction or delete in the OLTP primary driving table or auxiliary tables that are sourced to build the supervisor hierarchy structure in the warehouse, changes that are effective prior to the last ETL run date will be loaded into the warehouse as new records but not as corrections, thereby introducing duplicates during the index creation process and causing the ETL to abort. In this release, back-dated changes and deletes in the OLTP that affect the supervisor and position hierarchy will be processed accordingly by the ETL to reflect the retroactive changes in the warehouse.

8.2 Supported Back-Dated Change Scenarios

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

In an OLTP source system such as Oracle EBS, corrections can be made to historical rows or the current rows. Corrections can be made to OLTP tables that provide the

primary data source for the data warehouse or to tables that serve as auxiliary lookups in the ETL process. Corrections can affect the following attributes:

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

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

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

- Type A: The primary driving table that contains all historical assignment data, for example PER_ALL_ASSIGNMENTS_F. This table is the primary driver for the employee assignment history in the data warehouse.
- Type B: Auxiliary source table which keeps no history (Type 1 SCD) in OLTP but the warehouse tracks attribute changes as Type 2 SCD, for example HR_ORGANIZATION_UNITS. This table is the source for attributes that are history tracked in the warehouse.
- Type C: Auxiliary source table which keeps no history in OLTP and the warehouse also keeps no historical changes for the attribute (Type 1 SCD), such as PER_JOBS. This table is the source for attributes that are not history tracked in the warehouse.

The functionality in Oracle BI Applications Release 7.9.6 assumes that:

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

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

Table 8–1 Example of Employee P1's Assignment History Before the Back-Dated Changes

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

8.2.1 Back-Dated Change Scenarios That Affect Position Hierarchy

This section describes the types of OLTP back-dated changes that affect the Position Hierarchy and the solutions provided in the warehouse to address the back-dated change scenarios.

8.2.1.1 Scenario 1

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

Option 1

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

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

Option 2

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

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

8.2.1.2 Scenario 1a

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

Option

Update the name change in the relevant history records.

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

8.2.1.3 Scenario 2

Auxiliary table change: Job information is changed in the source OLTP system that is referred to in the historical data in the warehouse, for example Job name is changed to

lower case. This is a Type C change where neither the OLTP nor the data warehouse track historical changes.

Option

New Job is propagated to all historical rows.

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

8.2.1.4 Scenario 3

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

Option

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

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

8.2.1.5 Scenario 3a

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

Option 1

Update the historical data in the warehouse. This doesn't require fact table updates. This option is enabled if DAC parameter UPDATE_CORRECTIONS_FLG is set to 'Y'.

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

Option 2

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

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

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

8.2.1.6 Scenario 3b

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

Option

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

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

8.2.1.7 Scenario 3c

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

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

Option

Update the historical and current records in the warehouse:

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

8.2.1.8 Scenario 3d

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

Option

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

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

8.3 Objects Changed to Support Back-Dated Changes in Supervisor/Position Hierarchy

This section describes the objects have been changed to support back-dated changes in Position Hierarchy.

8.3.1 Data Model Changes

The following changes were made to the data model:

- Added SCD1_WID to W_POSITION_D and W_POSITION_DH.
- Adding DELETE_FLG to W_POSITION_DH
- Adding new tables W_POSITION_DH_BAK, W_POSITION_DH_PRE_CHG_TMP, W_POSITION_DH_POST_CHG_TMP
- Adding new columns SCD_TYPE1B_FLG and SCD_TYPE2B_FLG to W_POSITION_DH_BASE_TMP

8.3.2 Informatica Code Changes

This section describes the code changes in Informatica.

8.3.2.1 Mappings

The following new mappings have been added:

- SIL_PositionDimensionHierarchy_SoftDelete
- SIL_PositionDimensionHierarchy_IdentifyBaseModified_TypeB
- SIL_PositionDimensionHierarchy_PreChangeTemp

- SIL_PositionDimensionHierarchy_PostChangeTemp
- SIL_PositionDimensionHierarchy_CopyDelete
- SIL_PositionDimensionHierarchy_Delete

The following mappings have been updated:

- SIL_PositionDimensionHierarchy
- SIL_PositionDimensionHierarchy_Full
- SIL_PositionDimensionHierarchy_IdentifyBaseModified

8.3.2.2 Sessions/Workflows

The following sessions/workflows have been added to Informatica:

- SIL_PositionDimensionHierarchy_SoftDelete
- SIL_PositionDimensionHierarchy_IdentifyBaseModified_TypeB
- SIL_PositionDimensionHierarchy_PreChangeTemp
- SIL_PositionDimensionHierarchy_PostChangeTemp
- SIL_PositionDimensionHierarchy_CopyDelete
- SIL_PositionDimensionHierarchy_Delete

8.3.2.3 Reusable Lookup

To support back-dated changes to position hierarchy, the Lkp_W_Position_DH_WID_Type1 lookup was added.

8.3.3 DAC Updates

The following tasks have been added to DAC:

- SIL_PositionDimensionHierarchy_SoftDelete
- SIL_PositionDimensionHierarchy_IdentifyBaseModified_TypeB
- SIL_PositionDimensionHierarchy_PreChangeTemp
- SIL_PositionDimensionHierarchy_PostChangeTemp
- SIL_PositionDimensionHierarchy_CopyDelete
- SIL_PositionDimensionHierarchy_Delete

