

Oracle Business Intelligence Applications for SAP

Install Config Guide

Release 7.9.7.2

E18927-03

Copyright © 2010, 2012, 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 is software or related documentation that 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 America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware 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 that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware 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.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Table of Contents

Oracle Business Intelligence Applications for SAP	1
Preface	9
Audience	10
Documentation Accessibility.....	11
Accessibility of Code Examples in Documentation	12
Accessibility of Links to External Web Sites in Documentation	13
TTY Access to Oracle Support Services	14
Related Documents.....	15
Conventions	16
Chapter 1: What's New in This Release	17
What's New In Oracle Business Intelligence Applications Version 7.9.7.2?	17
Chapter 2: Overview of Oracle Business Intelligence Applications	18
What is Oracle Business Intelligence Applications?.....	18
Oracle Business Analytics Warehouse Overview	19
Oracle Business Analytics Warehouse Architecture	19
Oracle Business Analytics Warehouse Architecture Components	19
Roadmap to Installing, Configuring, and Customizing Oracle Business Intelligence Applications with ODI	20
Using Oracle BI-EE Repository Documentation	21
Chapter 3: Pre-installation and Pre-deployment Requirements for Oracle Business Intelligence Applications.....	22
Oracle-Specific Database Guidelines for Oracle Business Analytics Warehouse.....	22
Database Client Connectivity Software Requirements	22
Chapter 4: Installing and Setting Up Oracle Business Intelligence Applications	23
About Oracle Business Intelligence Applications Topologies	23
Installation and Set up Process Task List	26
Mandatory Requirements and Pre-installation Tasks	27
Oracle Business Intelligence Infrastructure Requirements	27
Code Page and Data Movement Requirements	27
Pre-installation Tasks	28
Installing Oracle Business Intelligence Applications	34
How to set up the Oracle BI Applications files for ODI	38
Setting Up the Oracle Business Intelligence Applications and ODI Components	40
(Optional) How to Copy Over Installation Files.....	41

How to configure the ODI parameter file	41
How to configure the ODI Repository	41
How to create the blank ODI Work Repository	45
How to set up the ODI Topology	48
Setting up the Data Servers	48
How to set up the Data Source Number	64
How to set up the ODI Designer connection to the ODI Master Repository	65
How to import the Oracle BI Applications Work Repository	67
How to start the ODI Agents	68
How to set up the Agents	70
How to set up the INTERFACE Agent	70
How to set up the WORKFLOW Agent	71
Add the Open Tool	73
How to install Base programs in SAP using RKM	73
Validating the LKM Option for file transfer	76
Indexes Recommendation for Performance improvement on SAP Server	76
How to Upload SAP ABAP programs (PART 1 of 2)	77
How to generate the required ODI Scenarios	79
How to install the Schema Objects for the Oracle Business Analytics Warehouse	81
Setting Up Oracle BI Applications Configuration Manager	85
Using Oracle BI Applications Configuration Manager	97
About Resetting the Default Passwords, Schema Names, and ODI Connection Details	100
How to configure and activate Automated Database Statistics Collection	105
How to configure and activate Automated Delete Handling	106
About Data Warehouse Loads	107
How to reset the Oracle Business Analytics Warehouse for Full Load	107
How to add partitions to Data Warehouse tables	108
About deploying ODI across multiple environments	115
How to run an ODI Agent as a Unix background process	116
How to uninstall ODI Agent Windows Services	116
How to Resolve Conflicts in ODI Agent Port Numbers	117
How to activate or deactivate Flow Control in ODI	117
List of Log Files	117
Loading Source Data Using an Execution Plan	119
How to set up Master Packages to run an Execution Plan	120

How to perform E-LT Using Oracle BI Applications Configuration Manager	123
How to Upload SAP ABAP programs (PART 2 of 2)	127
How to perform E-LT Using ODI Designer	135
Example of Running a Full Load E-LT in Oracle BI Applications Configuration Manager.....	138
Managing Execution Plans	141
Editing and Deleting Execution Plans.....	141
Monitoring E-LT Processes.....	141
How to Monitor E-LT Processes	142
How to Monitor E-LT Processes	142
Miscellaneous and Supporting Tasks	144
How to verify installation and Set up.....	144
Using ODI Designer to Manage Your Environment.....	145
Chapter 5: Configuring Common Areas and Dimensions	149
Steps Required Before a Full Load	149
How to Configure Initial Extract Date	149
Configuring Global Currencies	150
How to Configure Fiscal Calendar	152
About Configuring Calendars	153
About Configuring Enterprise Calendars	153
Using SAP ERP Application to configure the Enterprise calendar.....	154
About Configuring Warehouse Generated Fiscal Calendars.....	155
Notes on Configuring Calendars	155
Examples of Configuring the Universal Adapter	157
Configuring the General Ledger Account Hierarchies	160
Configuring Product Hierarchy, Master Inventory Organization, and Group Accounts	171
Configuration Steps for Controlling Your Data Set	176
Chapter 6: Configuring Oracle Financial Analytics for SAP	179
Overview of Oracle Financial Analytics	179
Configuration Required Before a Full Load for Financial Analytics for SAP	180
About Configuring Domain Values and CSV Worksheet Files for Oracle FinancialAnalytics	180
How to configure file_xact_category_AP_SAP.csv and file_xact_category_AR_SAP.csv for Oracle General Ledger and Profitability Analytics.....	183
How to Specify the Ledger or Set of Books for which GL Data is Extracted	189
Configuration Steps for Controlling Your Data Set	189
How to Configure AP Balance ID for Oracle Payables Analytics	190

How to Configure AR Balance ID for Oracle Receivables Analytics and Oracle General Ledger and Profitability Analytics	191
Chapter 7: Configuring Oracle Procurement and Spend Analytics for SAP	194
Overview of Oracle Procurement and Spend Analytics for SAP	194
Oracle Procurement and Spend Analytics Module	194
Supplier Performance Analytics Module	195
Configuration Required Before A Full Load for Oracle Procurement and Spend Analytics.....	195
How to Configure the Parameter for Purchase Cycle Line	196
Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics.....	196
Configuration Steps for Controlling Your Data Set	226
Chapter 8:Configuring Oracle Supply Chain and Order Management Analytics	229
Overview of Oracle Supply Chain and Order Management Analytics for SAP	230
Configuration Required Before A Full Load for Oracle Supply Chain and Order Management Analytics for SAP	230
About Configuring Domain Values and CSV Worksheet Files for Oracle Supply Chain and Order Management Analytics	231
How to Configure Invoice Type Domain Values.....	233
How to Configure Pick Types Domain Values	234
How to Configure Order Types Domain Values	235
How to Configure Pick Status Domain Values	236
How to Configure Invoice Status Domain Values	237
How to Configure Order Overall Status Domain Values	238
How to Configure Order Schedule Status Domain Values.....	239
Chapter 9: Configuring the Oracle Business Intelligence Applications Repository	240
Configuring the Oracle BI Repository Connections.....	241
About The Predefined Connection Pools In The Oracle Business Analytics Warehouse.....	241
How to Configure Oracle Business Intelligence Applications Repository Connections	242
How to Configure Oracle Business Intelligence Applications Repository Variables	242
Deploying the Metadata Repository and Presentation Catalog	243
Applying the Oracle BI Applications Security Policy to the BI Domain	245
Localizing the Policy Store.....	245
Importing the Oracle BI Applications Version 7.9.7.2 Identity Store (LDIF) File into the Embedded LDAP Server.....	246
Setting up Date Specific Metrics	246
Setting Up Additional Time Series Metrics for Oracle Business Analytics Warehouse	247
Setting Up Additional Dimension Tables for Oracle Business Analytics Warehouse.....	247

About the Period Ago Keys for Oracle Business Analytics Warehouse	248
About Oracle BI Time Repository Variables	248
About User Authentication	250
About the Security or Visibility Configuration	250
About the Group Variable	252
About Configuring Usage Tracking for Oracle Business Analytics Warehouse.....	252
About the Incremental Deployment of the Oracle BI Applications Repository.....	253
Chapter 10: Customizing the Oracle Business Analytics Warehouse	254
Overview of Customization in Oracle Business Intelligence Applications	254
What is Customization in Oracle Business Intelligence Applications?.....	254
About the Customization Process	255
About the Impact of Patch Installation on Customizations	256
Category 1 Customizations: Adding Columns to Existing Fact or Dimension Tables	258
Including a Source Table for the Change Capture Process	258
ETL Process.....	259
About Extending Mappings.....	259
Typical Steps to Extend Mappings in the Oracle Business Analytics Warehouse.....	260
Example of Extending the Oracle Business Analytics Warehouse	261
Other Types of Customizations Requiring Special Handling.....	269
Category 2 Customizations: Adding Additional Tables	277
About Creating New Dimension or Fact Tables	277
About the DATASOURCE_NUM_ID Column.....	278
Creating Custom ODI Master Packages	278
Adding a New Dimension to the Oracle Business Analytics Warehouse	287
Adding a New Fact Table to the Oracle Business Analytics Warehouse.....	288
Adding a New Dimension Table for a New Fact Table in the Oracle Business Analytics Warehouse	289
Category 3 Customizations	289
How to Add New Data as a Whole Row Into a Standard Dimension Table	289
Configuring Extracts	290
Configuring Loads.....	292
Customizing Stored Lookups, Domain Values, and Adding Indexes	299
About Stored Lookups.....	299
Codes Lookup	299
About Domain Values	302
How to add an index to an existing fact or dimension table	305

Appendix 309

 Localizing Oracle Business Intelligence Deployments 309

 1. Process of MaintainingTranslationTables for Oracle BI 309

Preface

Oracle Business Intelligence Applications for SAP are comprehensive prebuilt solutions that deliver pervasive intelligence across an organization, empowering users at all level, 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 SAP, and corporate data warehouses, into actionable insight that enables more effective actions, decisions, and processes.

Oracle Business Intelligence Applications are built on Oracle Business Intelligence Suite Enterprise Edition, a comprehensive next-generation BI and analytics platform. Oracle Business Intelligence Applications include the following application families:

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

This guide explains how to install, set up, configure, and customize Oracle Business Intelligence Applications Version 7.9.7.2. For a high level road map of the steps required, see "Roadmap to Installing, Configuring, and Customizing Oracle Business Intelligence Applications with ODI".

- Oracle recommends reading Oracle Business Intelligence Applications Release Notes before installing or using Oracle Business Intelligence Applications. 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 Business Intelligence Applications.

Documentation Accessibility

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

TTY Access to Oracle Support Services

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

Related Documents

For more information, see the following documents in the Oracle Business Intelligence Applications Release 7.9.7.2 documentation set

- *Oracle Business Intelligence Applications Release Notes for Oracle Data Integrator Users*
- *Oracle Business Intelligence Applications Security Guide*

(available at <http://www.oracle.com/technetwork/middleware/bi-foundation/documentation/bi-apps-098545.html>):

- *Certification Matrix for Oracle Business Intelligence Applications for Oracle Data Integrator Users*
(available at <http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html>)

Conventions

The following text conventions are used in this document:

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

Chapter 1: What's New in This Release

This topic summarizes the adapters and content that is supported in this release of Oracle Business Intelligence Applications.

What's New In Oracle Business Intelligence Applications Version 7.9.7.2?

The main changes in Oracle Business Intelligence Applications Version 7.9.7.2 are:

- Oracle Data Integrator support.

Oracle Business Intelligence Applications Version 7.9.7.2 uses Oracle BI Applications Configuration Manager in conjunction with Oracle Data Integrator (ODI) 10.1.3.6.9 to perform Extract-Load Transform (E-LT). 7.9.7.1 was using Oracle Data Integrator (ODI) 10.1.3.6.1 to perform Extract-Load Transform (E-LT).

- Applications support (No Change from 7.9.7.1).

Oracle Business Intelligence Applications Version 7.9.7.2 supports the following application product families:

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

Chapter 2: Overview of Oracle Business Intelligence Applications

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

- What is Oracle Business Intelligence Applications?
- Oracle Business Analytics Warehouse Overview
- Oracle Business Analytics Warehouse Architecture
- Oracle Business Analytics Warehouse Architecture Components
- Roadmap to Installing, Configuring, and Customizing Oracle Business Intelligence Applications with ODI
- Using Oracle BI-EE Repository Documentation

What is Oracle Business Intelligence Applications?

Oracle Business Intelligence Applications is a prebuilt business intelligence solution. Oracle Business Intelligence Applications Version 7.9.7.2 supports SAP R/3 release 4.6c and ECC 6.0 source systems, and uses Oracle Data Integrator (ODI) as its E-LT environment.

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

Table 1. *Oracle Business Intelligence Applications Components (with ODI)*

Component	Description
Oracle Data Integrator	This is the Oracle data integration tool (E-LT tool) that performs the extract, load, transform operations for the data warehouse. Oracle Data Integrator works in conjunction with Oracle BI Applications Configuration Manager.
Oracle BI Applications Configuration Manager	This is a Web tool that enables you to set E-LT parameters, create and execute Execution Plans, and monitor ELT executions. Oracle BI Applications Configuration Manager works in conjunction with Oracle Data Integrator.
Prebuilt ODI content	This content includes Extract-Load-Transform (E-LT) repository objects (that is, scenarios, packages, and interfaces), which are contained in the ODI repository.
Prebuilt metadata content	This metadata content is contained in the Oracle Business Intelligence Applications repository file (EnterpriseBusinessAnalytics.rpd).
Prebuilt reports and dashboard content	This content is contained in the Oracle BI Presentation Services Catalog.
Oracle Business Analytics Warehouse	The prebuilt data warehouse that holds data extracted, loaded, and transformed from the transactional database (for more information, see "Oracle Business Analytics Warehouse Overview").

Oracle Business Analytics Warehouse Overview

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

The Oracle Business Analytics Warehouse includes the following:

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

For more information about the data warehouse data model, please see the Oracle Business Analytics Fusion Edition Data Model Reference.
- An open architecture to allow organizations to use third-party analytical tools in conjunction with the Oracle Business Analytics Warehouse using the Oracle Business Intelligence Server.
- Prebuilt data extractors to incorporate data from external applications into the Oracle Business Analytics Warehouse.
- A set of E-LT (extract, load and transform) processes that takes data from an SAP R/3 release 4.6c and ECC 6.0 source system to create the Oracle Business Analytics Warehouse tables.
- A set of easy-to-use tools for the set up, configuration, administration, loading, and monitoring of the Oracle Business Analytics Warehouse. For example, ODI Designer, and Oracle BI Applications Configuration Manager.

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

Oracle Business Analytics Warehouse Architecture

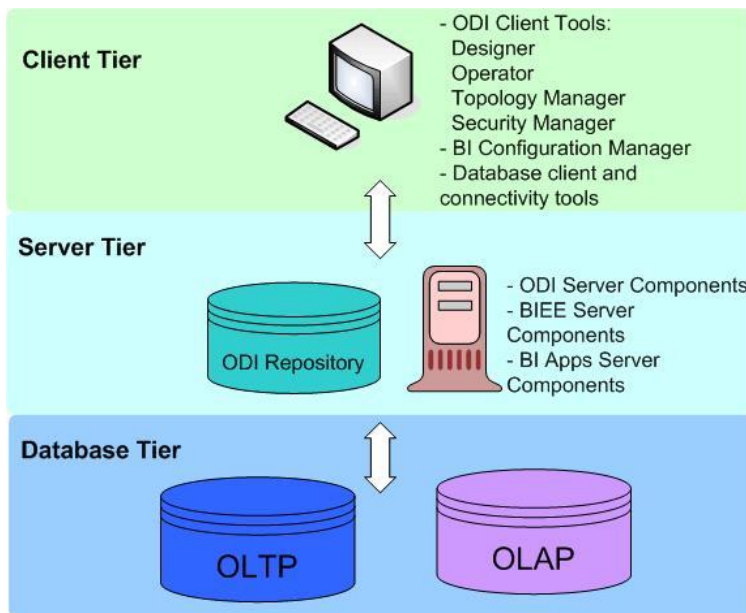
High-level analytical queries, like those commonly used in Oracle Business Intelligence, scan and analyze large volumes of data using complex formulas. This Roadmap to Installing, Configuring, and Customizing Oracle Business Intelligence Applications with ODI process can take a long time when querying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

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

Oracle Business Analytics Warehouse Architecture Components

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

Figure 1. An example Oracle Business Analytics Warehouse architecture with ODI



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

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

Roadmap to Installing, Configuring, and Customizing Oracle Business Intelligence Applications with ODI

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

- Optimize your Oracle database, as described in the chapter titled "Pre-installation and Pre-deployment Requirements for Oracle Business Intelligence Applications".
- Install and set up the Oracle Business Intelligence Applications components and Oracle Data Integrator components as described in the chapter titled "Installing and Setting up Oracle Business Intelligence Applications".

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

- (Optional) Perform any required configuration steps for the applications that you deploy, from the appropriate chapters below:
 - All applications - see the chapter titled "Configuring Common Areas and Dimensions".
 - Financial applications - see the chapter titled "Configuring Oracle Financial Analytics for SAP".
- (Optional) If you want to modify the out-of-the-box Oracle Business Intelligence Repository (RPD file), see the chapter titled "Configuring the Oracle Business Intelligence Applications Repository".
- (Optional) If you want to customize the out-of-the-box Oracle Business Intelligence Applications functionality, follow the steps described in the chapter titled "Customizing the Oracle Business Analytics Warehouse".

- (Optional) If you want to modify the out-of-the-box Oracle Business Intelligence Applications security, follow the steps described in Oracle Business Intelligence Applications Security Guide.

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

For a detailed example of how to run an E-LT process for Oracle Financials with an SAP ERP Source, see "Example of Running a Full Load E-LT in Oracle BI Applications Configuration Manager".

Using Oracle BI-EE Repository Documentation

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

- Oracle BI-EE Repository Documentation

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

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

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

- Oracle BI-EE Presentation Services Catalog

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

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

Chapter 3: Pre-installation and Pre-deployment Requirements for Oracle Business Intelligence Applications

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

- Oracle-Specific Database Guidelines for Oracle Business Analytics Warehouse
- Database Client Connectivity Software Requirements

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

Oracle-Specific Database Guidelines for Oracle Business Analytics Warehouse

To configure the Business Analytics Data Warehouse on Oracle databases more easily, refer to the parameter template files `init10gR2.ora` and `init11g.ora`. These files are located in `biapps_odi.zip` file on the Oracle Business Intelligence Applications installation machine.

The parameter template files provide parameter guidelines based on the cost-based optimizer for Oracle 10g and 11g. Use these guidelines as a starting point. You will need to make changes based on your specific database sizes, data shape, server size (CPU and memory), and type of storage. The database administrator should make changes to the settings based on performance monitoring and tuning.

Copy the appropriate template file into your `$ORACLE_HOME/dbs` directory. Then, review the recommendations in the template file, and make the changes based on your specific database configuration. The database administrator should make changes to the settings based on performance monitoring and tuning considerations.

Note: The `NLS_LENGTH_SEMANTICS` parameter enables you to define byte- or character-length semantics. Oracle BI Applications supports `BYTE` and `CHAR` values for this parameter. You can add this parameter to the `init10gR2.ora` and `init11g.ora` files if you are using `MLS` characters.

Note: Oracle provides Globalization Support that enables users to interact with a database in their own language, as defined by the `NLS_LANG` parameter. Please refer to Oracle Database Globalization Support for setting up `NLS_LANG` parameters for the DB Instance of the Warehouse.

Database Client Connectivity Software Requirements

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

Chapter 4: Installing and Setting Up Oracle Business Intelligence Applications

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

- About Oracle Business Intelligence Applications Topologies
- Installation and Set up Process Task List
- Mandatory Requirements and Pre-installation Tasks
- Installing Oracle Business Intelligence Applications and Oracle Data Integrator Software
- Setting Up the Oracle Business Intelligence Applications and ODI Components
- Miscellaneous and Supporting Tasks
- Loading Source Data Using an Execution Plan

For information about supported platform versions, see Certification Matrix for Oracle Business Intelligence Applications for Oracle Business Intelligence Applications.

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

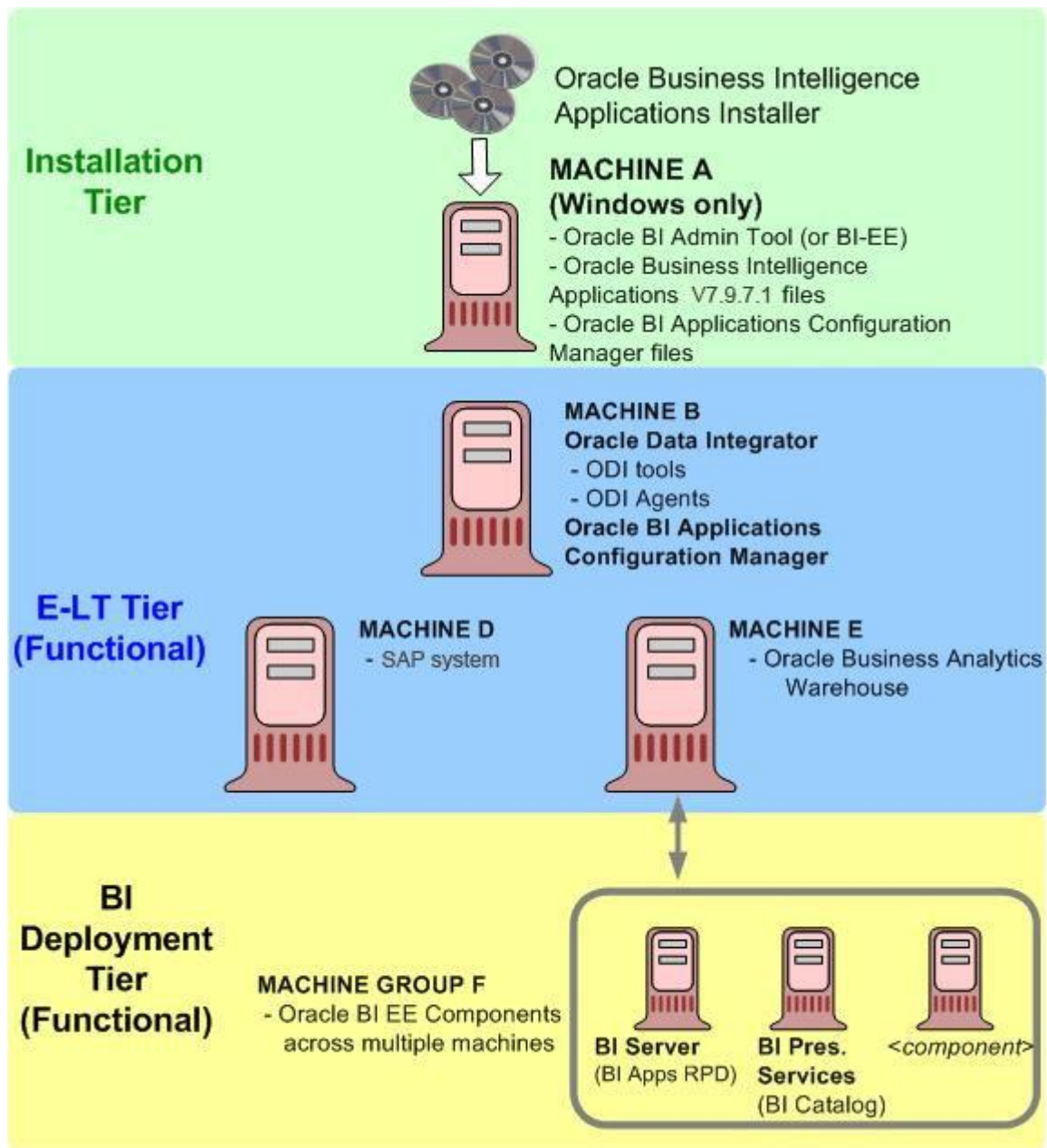
About Oracle Business Intelligence Applications Topologies

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

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

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

Figure 2. A typical topology for an Oracle Business Intelligence Applications deployment with ODI



In the figure above, note the following:

- Installation Tier
 - MACHINE A (Windows-only)

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

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

 - You manually copy the <OBIEE 11G Instance HOME>\biapps7972\repository EnterpriseBusinessAnalytics.rpd file from MACHINE A to the machine that runs the BI Server

in MACHINE GROUP E in the path <OBIEE 11G Instance HOME>\OracleBIServerComponent\coreapplication_obis1\repository\EnterpriseBusinessAnalytics.rpd

- You manually copy the <OBIEE 11G Instance HOME>\biapps7972\catalog files from MACHINE A to the machine that runs the BI Presentation Services Catalog in MACHINE GROUP E in the following path <OBIEE 11G Instance HOME>\OracleBIPresentationServicesComponent\coreapplication_obips1\catalog\EnterpriseAnalytics*.*

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

- You manually copy the <OBIEE 11G Home>\biapps7972\odi\biapps_odi\biapps_odi.zip file from MACHINE A to the Oracle Data Integrator machine (that is, MACHINE B) and unzip the file into the oracledi\ directory.
- You manually copy the Oracle BI Applications Configuration Manager files in the <OBIEE 11G Home>\biapps7972\odi\biapps_configmgr directory from MACHINE A to the Oracle Data Integrator machine (that is, MACHINE B).

- E-LT Tier (Functional)

- MACHINE B (Windows, Unix, Linux)

MACHINE B is a machine on which ODI is installed, and which runs the ODI production environment (for example, ODI tools, ODI Agents). You manually copy the biapps_odi.zip to this machine, and unzip the file into the \oracledi\ directory. In addition, you manually copy the Oracle BI Applications Configuration Manager files in the the <OBIEE 11G Home>\biapps7972\odi\biapps_configmgr directory from MACHINE A to this machine.

- MACHINE C (Windows, Unix, Linux)

MACHINE C is a machine that hosts the transactional (OLTP) database.

- MACHINE D (Windows, Unix, Linux)

MACHINE D is a machine that hosts the Oracle Business Analytics Warehouse database.

- BI Deployment Tier (Functional)

The BI Deployment Tier (Functional) tier is used to deploy the business intelligence dashboards, can have either of the following:

- MACHINE GROUP E (Windows, Unix, Linux)

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

Notes

- In an ODI environment, you typically maintain different environments (known as Contexts) for QA, Development, and Production. For information about Contexts, and about moving from a test to a production environment in ODI, please refer to the section titled "About deploying ODI across multiple environments".
- To maintain a back-up of the ODI project for recovery purposes, use the Import and Export option in ODI Designer. For example, you can select File, then Export to export the Work Repository to a

directory or ZIP file. You can select File, then Import to import the Work Repository from a directory or ZIP file. For information about how to import a Work Repository, see "How to import the Oracle BI Applications Work Repository".

Installation and Set up Process Task List

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

1. Before you install Oracle Business Intelligence Applications and Oracle Data Integrator, make sure that you have satisfied the following requirements:
 - a. Make sure that you satisfy the Oracle Business Intelligence Infrastructure requirements that are specified in "Oracle Business Intelligence Infrastructure Requirements".
 - b. Make sure that you satisfy the Oracle Data Integrator requirements that are specified in the Oracle Data Integrator Installation Guide. In addition, before you install ODI, Oracle recommends that you read the ODI documentation in detail to enable you to plan your ODI topology. ODI documentation is available on the BI media pack.
 - c. Make sure that you satisfy the Code Page requirements that are specified in "Code Page and Data Movement Requirements").
 - d. Make sure that you perform those mandatory pre-installation tasks that are specified in "Pre-installation Tasks").
2. Install the Oracle Business Intelligence Applications Version 7.9.7.2 software (for more information, see "Installing Oracle Business Intelligence Applications and Oracle Data Integrator Software"), as follows:
 - a. Run the Oracle installer to install the Oracle Business Intelligence Applications software. For more information, see "How to Run the Oracle Business Intelligence Applications Installer (Windows)".
 - b. Run the Oracle Data Integrator installer to install the Oracle Data Integrator software. For more information, see "Installing Oracle Data Integrator Version 10.1.3.6.9".
3. (Optional) Copy the BI server components to appropriate UNIX or Windows machines, see "(Optional) How to Copy over Installation Files".
4. Set up the Oracle Business Intelligence Applications and Oracle Data Integrator components (for more information, see "Setting Up the Oracle Business Intelligence Applications and ODI Components"), as follows:
 - a. Edit the ODI parameter file to set your deployment-specific settings (for more information, see "How to configure the ODI parameter file").
 - b. Set up the ODI Repository (for more information, see "How to configure the ODI Repository").
 - c. Set up the ODI Topology (for more information, see "How to set up the ODI Topology").
 - d. Create and start the ODI Agents (for more information, see "How to start the ODI Agents").
 - e. Create the ODI Scenarios (for more information, see "How to generate the required ODI Scenarios").
 - f. Set up the Data Warehouse Target Schema Objects (for more information, described in the chapter titled "How to install the Schema Objects for the Oracle Business Analytics Warehouse").
 - g. Setup Oracle BI Applications Configuration Manager. For more information, see "Setting up Oracle BI Applications Configuration Manager".

- h. Set up the Master Packages in ODI to run Execution Plans (for more information, see "How to set up Master Packages to run an Execution Plan").
5. Review and perform any post-installation tasks that are applicable to your deployment. For more information, see "Miscellaneous and Supporting Tasks".
6. Perform a test full load E-LT (for more information, see "Loading Source Data Using an Execution Plan").

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

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

Mandatory Requirements and Pre-installation Tasks

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

- Oracle Business Intelligence Infrastructure Requirements
- Code Page and Data Movement Requirements
- Pre-installation Tasks

Oracle Business Intelligence Infrastructure Requirements

Oracle Business Intelligence Applications has the following infrastructure requirements:

The Oracle BI Applications installer runs on Windows and requires an Oracle Business Intelligence infrastructure (or ORACLE_HOME) to be installed in an Oracle Fusion Middleware home directory (or MW_HOME).

Note: This instance of Oracle Business Intelligence Enterprise Edition does not need to be the functional version that you will use to deploy reports and dashboards in your live system. This instance is only required to enable the Oracle BI Applications installer to install the Oracle BI Applications files on a machine. The functional version of OBIEE can be on any supported OS for OBIEE (that is, it is not limited to Windows).

To determine the minimum version of Oracle Business Intelligence Enterprise Edition that is supported for this release of Oracle BI Applications, see the System Requirements and Supported Platforms for Oracle Business Intelligence Applications.

Code Page and Data Movement Requirements

The Oracle Business Analytics Warehouse can be deployed in various code page environments and supports global deployments. Data movements in the following source database and data warehouse configuration modes are supported:

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

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

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

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

Pre-installation Tasks

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

- Opening Firewall Ports
- Creating the Required Databases and Tablespaces
- Configuring the Language Environment
- Installing Oracle Data Integrator Version 10.1.3.6.9
- How to install and configure JCo
- Installation of OBIA

Opening Firewall Ports

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

Creating the Required Databases and Tablespaces

This topic provides information on creating database accounts that are required by Oracle Business Intelligence Applications components. You must create the database accounts listed in this topic in a single Oracle database instance. For example SQL commands that you can use to create database accounts, see section "Example SQL Commands for Creating Database Users and Tablespaces".

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

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

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

Table 2.Create the Following Database Users and Tablespaces

Database User ID	Description	Default Password	Tablespace	Privileges
ODI_REP_7972	ODI Master and Work Repositories database schema.	ODI_REP_7972	BIAPPS_REP	CONNECT, RESOURCE
TEMP_BIAPPS	ODI staging database schema. TEMP_BIAPPS is used by ODI to create and drop segments during ELT processes. This schema must have a dedicated and locally managed tablespace with a uniform extent size.	TEMP_BIAPPS	BIAPPS_TEMP	CONNECT RESOURCE
DATA_BIAPPS	Data Warehouse database schema.	DATA_BIAPPS	BIAPPS_DATA	CONNECT RESOURCE CREATE DATABASE LINK CREATE ANY DIRECTORY CREATE ANY INDEX CREATE ANY PROCEDURE CREATE ANY SEQUENCE CREATE ANY SYNONYM CREATE ANY TABLE CREATE ANY TRIGGER CREATE ANY VIEW DELETE ANY TABLE DROP ANY DIRECTORY DROP ANY INDEX DROP ANY PROCEDURE DROP ANY SEQUENCE DROP ANY SYNONYM DROP ANY TABLE

				DROP ANY TRIGGER DROP ANY VIEW INSERT ANY TABLE SELECT ANY SEQUENCE SELECT ANY TABLE UPDATE ANY TABLE ANALYZE ANY
Index Tablespace	Index Tablespace. This is a mandatory Tablespace that is used as the INDEX_TABLESPACE FlexField value in ODI (for more information, see "Verifying the INDEX_TABLESPACE Setting").	Not applicable.	BIAPPS_INDEX	Not applicable.
DATA_BIAPPSTX	Oracle BI Applications Configuration Manager database schema. Additional privileges are created as part of the setup (for more information, see "How to Set Up Oracle BI Applications Configuration Manager on Windows").	DATA_BIAPPSTX	BIAPPS_DATA	CONNECT RESOURCE

Configuring the Language Environment

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

Setting the NLS_LANG Environment Variable for Oracle Databases

Follow this procedure to set the NLS_LANG environment variable for Oracle databases.

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

To set the NLS_LANG environment variable for Oracle databases:

1. Determine the NLS_LANG value.
 - a. In the data warehouse database, run the command

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

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

For example: American_America.UTF8

2. For Windows:

- a. Navigate to Control Panel > System and click the Advanced tab. Click Environment Variables.
- b. In System variables section, click New.
- c. In the Variable Name field, enter NLS_LANG.
- d. In the Variable Value field, enter the NLS_LANG value that was returned in Step 1.

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

For example: American_America.UTF8.

Note: The NLS_LANG character set should reflect the setting of the operating system character set of the client. For example, if the database character set is AL32UTF8 and the client is running on a Windows operating system, then you should not set AL32UTF8 as the client character set in the NLS_LANG parameter because there are no UTF-8 WIN32 clients. Instead, the NLS_LANG setting should reflect the code page of the client. For example, on an English Windows client, the code page is 1252. An appropriate setting for NLS_LANG is AMERICAN_AMERICA.WE8MSWIN1252. Setting NLS_LANG correctly allows proper conversion from the client operating system character set to the database character set. When these settings are the same, Oracle assumes that the data being sent or received is encoded in the same character set as the database character set, so character set validation or conversion may not be performed.

This can lead to corrupt data if the client code page and the database character set are different and conversions are necessary

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

```
setenv NLS_LANG <NLS_LANG>
```

```
echo $NLS_LANG
```

For example: setenv NLS_LANG American_America.UTF8. Use the echo \$NLS_LANG command to verify that you have set the environment correctly. If your data is 7-bit or 8-bit ASCII and ODI is running on UNIX or Linux, then set NLS_LANG <NLS_LANGUAGE>_<NLS_TERRITORY>.WE8ISO8859P1

Note: The Language variables configurations should be done as per Oracle-Specific Database Guidelines for Oracle Business Analytics Warehouse.

Installing Oracle Data Integrator Version 10.1.3.6.9

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

To install Oracle Data Integrator Version 10.1.3.6.9 for Oracle Business Intelligence Applications, do the following:

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

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

- Agent
- Designer
- Operator
- Repository Management
- Security Manager
- Topology Manager
- You can add other ODI components later if required (for example, Oracle Data Profiling, Metadata Navigator) by running the ODI installer again on a machine. For more information, refer to the Oracle Data Integrator Documentation Library.
- For more information, on the patch level for or Oracle Data Integrator 10gR3, refer to the System Requirements And Supported Platforms Document.(Specific to Oracle Data Integrator SAP ABAP Adaptor)
- A Java Virtual Machine version compatible with both Oracle Data Integrator and JCo must be used. more information, refer to System Requirements And Supported Platforms Document.
- A Java Virtual Machine (JVM) enables a set of computer software programs and data structures to use a virtual machine model for the execution of other computer programs and scripts.

How to Run the Oracle Data Integrator Installer

ODI 10.1.3.6.9 is the ODI version required. Download the following patches from My Oracle Support (<https://support.oracle.com>):

1. Patch 14262715: Oracle Data Integrator, Oracle Data Profiling, and Oracle Data Quality for Data Integrator 10g (10.1.3.5.0)
2. Patch 13428746: ORACLE DATA INTEGRATOR 10.1.3.6.9 CUMULATIVE PATCH
3. Patch 14096928: ORACLE DATA INTEGRATOR 10.1.3.6.9_02 ONE OFF PATCH
4. Patch 14312673: ORACLE APPLICATION ADAPTERS FOR DATA INTEGRATION 10.1.3.6.9_02 ONE OFF PATCH

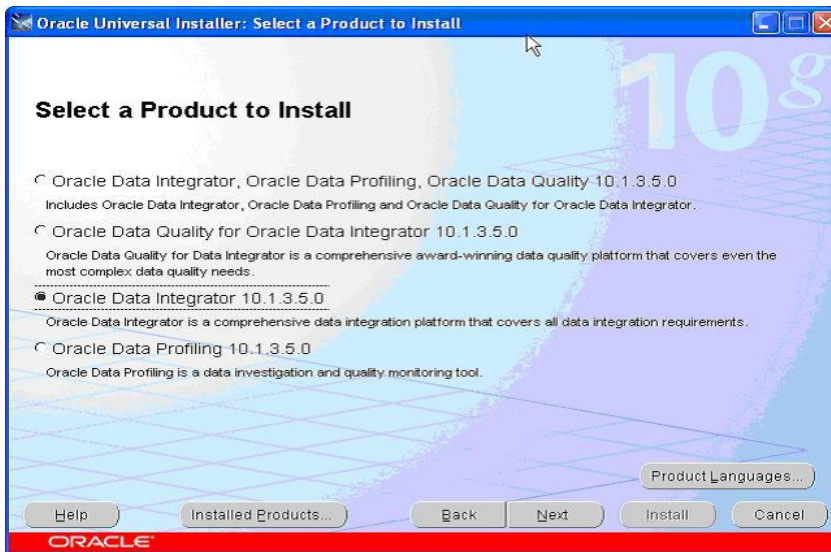
To start the Oracle Data Integrator installer to install ODI 10.1.3.5 (Patch 14262715):

1. Extract the ODI Installer zip file(s) to a suitable location, then:
 - on Windows, double-click setup.bat to start the installer.
 - on UNIX, enter the ./runInstaller command to start the installer (You may have to modify the permissions of the scripts to allow them to execute).

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

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

Figure 3.



3. Click Next and follow the remaining on-screen instructions to start the ODI installation.
4. The Oracle Data Integrator installer creates an Oracle Home directory that contains a \oracledi\ directory. For example, you might have an ODI installation directory called D:\OraHome_1\. The ODI installation directory is referred to as <ODI_HOME>.
5. After successfully installing Oracle Data Integrator 10.1.3.5.0, apply the patches in the following order :
 - Patch 13428746: Oracle Data Integrator 10.1.3.6.9 Cumulative Patch
 - Patch 14096928: Oracle Data Integrator 10.1.3.6.9_02 One Off Patch
 - Patch 14312673: Oracle Application Adapters For Data Integration 10.1.3.6.9_02 One Off Patch

Note:

On UNIX platforms, add the following environment variable for the user who has installed Oracle Data Integrator: ODI_JAVA_HOME=<Path where Java is installed>/JAVA/<JAVA_VERSION_FOLDER>. Refer to Oracle Data Integrator Installation Guide for a full list of environment variables required for other components.

For UNIX/Linux platform, you need to add some environment variables, for the user who installed Oracle Data Integrator. Variables should be defined as shown in the examples below:

```
LD_LIBRARY_PATH=<ODI_Home>/oracledi/drivers
CLASSPATH=$LD_LIBRARY_PATH/sapjco3.jar
JAVA_HOME=/usr/java/jdk1.6.0_34
ODI_JAVA_HOME=$JAVA_HOME
ORACLE_HOME=/home/oracle/app/oracle/product/11.2.0/client_2
PATH=$PATH:$LD_LIBRARY_PATH:$JAVA_HOME/bin:$ORACLE_HOME/bin
```

For Windows the ODI_JAVA_HOME has to be added to the system environment variables. Please ensure that there are no spaces in the install path.

How to install and configure JCo

The SAP adapter uses JCo to connect to the SAP system. SAP Java Connector (SAP JCo) is a middleware component that enables the development of SAP-compatible components and applications in Java. JCo must be configured before proceeding with the project.

To install and configure JCo:

1. Download a supported JCo version for your configuration from <http://service.sap.com/>. Check the supported JCo version in the Compatibility Matrix available at Oracle Technology Network for Oracle Data Integrator. Note that a minimum version of JCo 3.0.2 is required. Please take support of SAP BASIS team to download the compatible version of JCO.
2. Unzip the appropriate distribution package into an arbitrary directory {sapjco-install-path}.
3. Follow the installation instructions in {sapjco-install-path}/javadoc/installation.html for the respective platform.
4. Copy sapjco3.jar and sapjco3.dll (or respective binary) into the oracledi/drivers directory.

Installing Oracle Business Intelligence Applications

This topic explains how to install Oracle Business Intelligence Applications software and contains the following topics:

- How to Run the Oracle Business Intelligence Applications Installer

Note: After installation, you must follow the set up instructions specified in "Setting Up the Oracle Business Intelligence Applications and ODI Components". The Oracle Business Intelligence Applications installer runs on Windows, and requires an Oracle Business Intelligence infrastructure to be installed. For more information on Oracle Business Intelligence infrastructure requirements, including versions supported for this release of Oracle BI Applications, see "Oracle Business Intelligence Infrastructure Requirements".

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

This section explains how to install the Oracle BI Applications files using the Oracle BI Applications Installer.

When you run Oracle BI Applications installer, the Oracle BI Applications files are installed into the BI ORACLE_HOME in a directory named biapps7972.

Note: To launch the Oracle BI Applications installer in non-English platforms, use this command:

```
%JAVA_HOME%\bin\java.exe -cp <PATH_TO_7.9.7.2_BI_APPS_INSTALLER>\setup.jar -
Dtemp.dir="%TEMP%" -Dis.jvm.home="%JAVA_HOME%" -Dis.jvm.temp="1" -
Dis.external.home="<PATH_TO_7.9.7.2_BI_APPS_INSTALLER>" -Xms64m -Xmx128m run
```

To install Oracle BI Applications on Windows

Run the program setup.exe to display the Welcome page.

Figure 4:



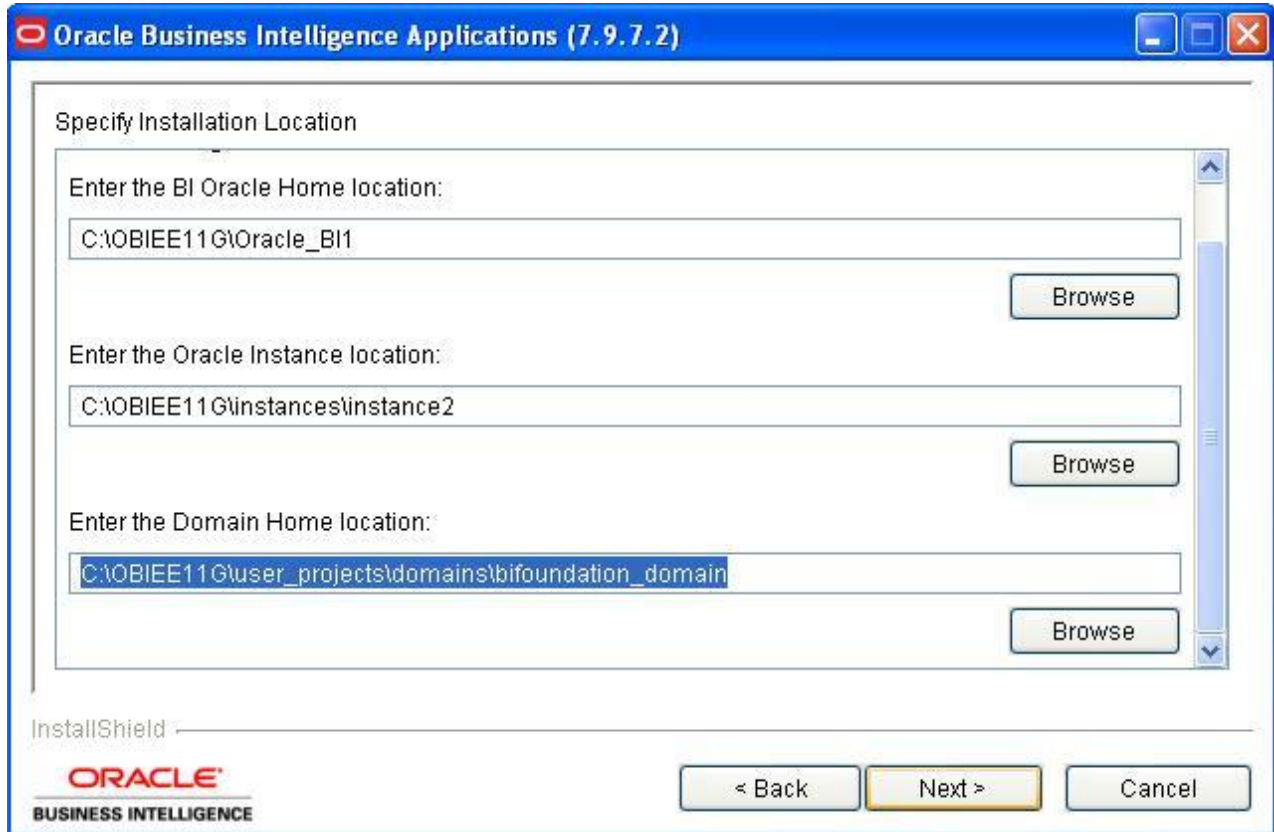
Note: To run the installer in console (or text) mode, run the command `setup.exe -console`. You do not see the following screens in console installation mode. Instead, you enter input as plain text in the terminal window when prompted.

Click Next to display the Specify Installation Location screen.

3. Use this screen to specify the details of the Oracle BI EE installation on the local drive.

Note: Oracle BI EE must be installed on a local drive; it cannot be a Windows mapped drive that maps to a separate machine.

Figure 5:



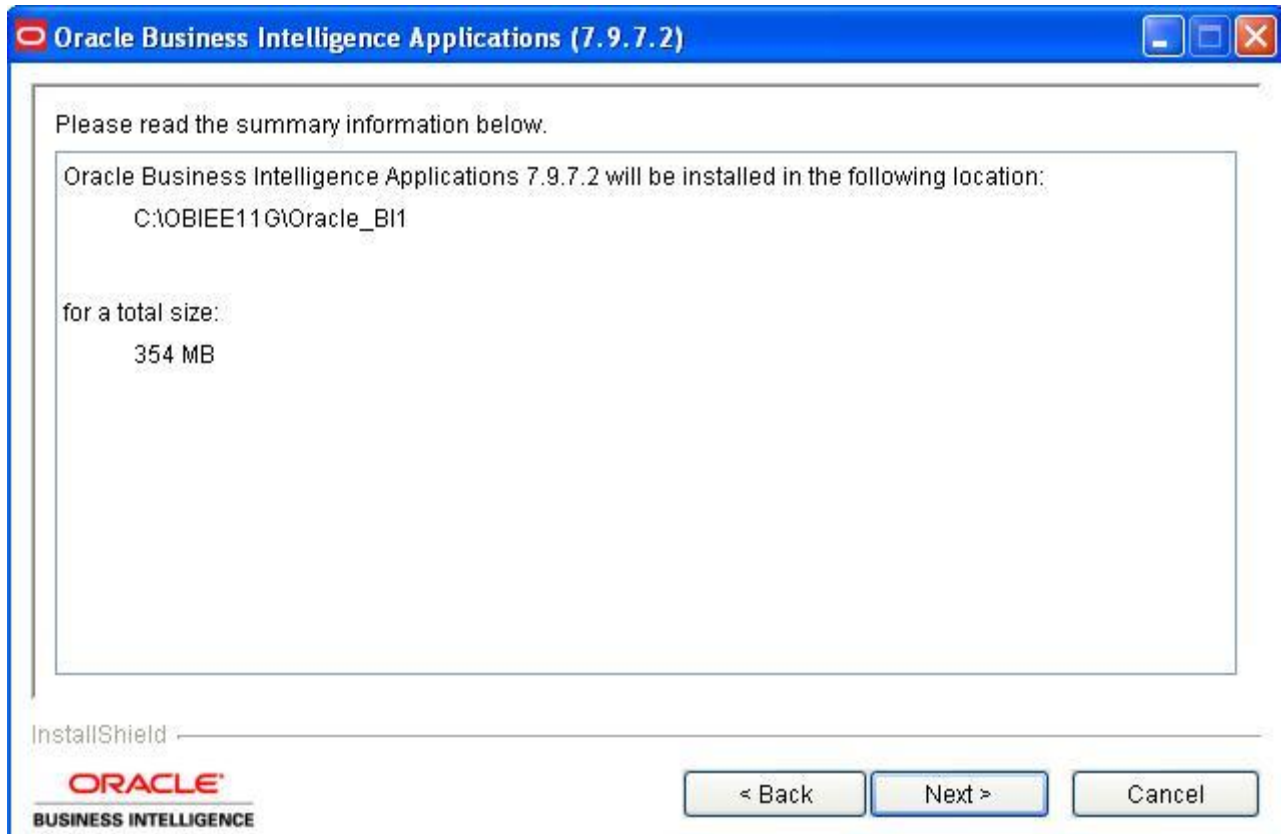
At the Enter the BI Oracle Home Location field, specify the Oracle BI EE Infrastructure home location (or ORACLE_HOME) on the local drive. For example, D:\OBIEE_NEW\Oracle_BI1\.

At the Enter the BI Instance Location field, specify the Oracle BI EE instance home location (or ORACLE_INSTANCE) on the local drive. For example, D:\OBIEE_NEW\instances\instance1.

At the Enter the Domain Home Location field, specify the BI domain home location on the local drive. For example, D:\OBIEE_NEW\user_projects\domains\bifoundation_domain.

4. Click Next to display the summary screen.
5. At the summary screen, review the summary information, and click Next to start the installation.

Figure 6:



The installer installs the Oracle BI Applications directories and files in the Oracle Business Intelligence infrastructure installation.

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

6. Click Finish.

Tip: Look in the < OBIEE11G_HOME >\biapps7972\version_apps.txt file in the to check that you have installed the correct version of Oracle BI Applications.

When the installation is complete, you will see the following directories and files:

File Location	File Name
<OBIEE11G_HOME>\biapps7972\repository	EnterprisebusinessAnalytics.rpd You must copy this file to the \OracleBIServerComponent\directory on a Oracle BI EE machine. For more information, see Section, "Deploying the Metadata Repository and Presentation Catalog."
<OBIEE11G_HOME> \biapps7972\catalog	The Presentation Catalog is provided as a zipped file named EnterpriseBusinessAnalytics.zip, which you must unzip before it can be used. For more information about unzipping the Presentation Catalog, see Section , "Deploying the Metadata Repository and Presentation Catalog."
<OBIEE11G_HOME> \biapps7972\odi	Sub-directories. Config manager: \Oracle_BI1\biapps7972\odi\biapps_configmgr Data Warehouse repository files and scripts: \Oracle_BI1\biapps7972\odi\biapps_odi

Notes

- Refer to the Oracle Business Intelligence Enterprise Edition documentation for more information on working with the Oracle BI Repository and Presentation Catalog.

How to set up the Oracle BI Applications files for ODI

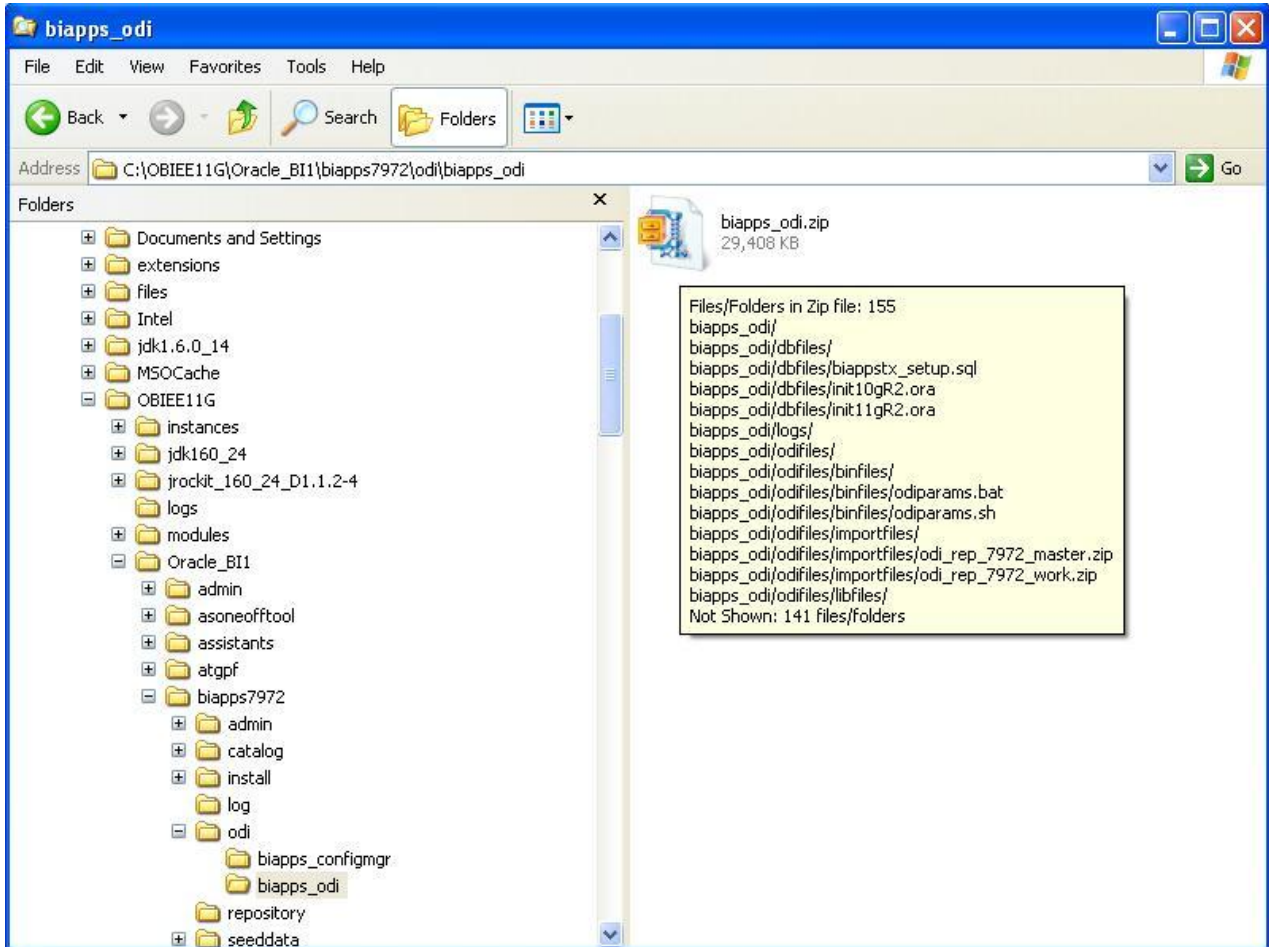
After you have installed Oracle Data Integrator Version 10.1.3.6.9, you need to manually add some Oracle Business Intelligence Applications files to the ODI installation machine, as follows:

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

1. On the Oracle Business Intelligence Applications installation machine, locate the biapps_odi.zip file in the <OBIEE11G_HOME> \biapps7972\odi\biapps_odi\ directory.

For example, D:\OBIEE_NEW\Oracle_BI1\biapps7972\odi\biapps_odi\biapps_odi.zip.

Figure 7.



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

For example, D:\OraHome_1\oracledi\.

3. On the Oracle Data Integrator machine, copy the <OBIEE11G_HOME> \biapps7972\odi\biapps_odi\ biapps_odi.zip file from the Oracle Business Intelligence Applications installation machine to the \oracledi\ directory on the Oracle Data Integrator installation machine.
4. On the Oracle Data Integrator installation machine, unzip the <ODI_HOME>\oracledi\biapps_odi.zip file into the <ODI_HOME>\oracledi\ directory.

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

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

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

5. On the Oracle Data Integrator machine, copy the files in the <ODI_HOME>\oracledi\biapps_odi\odifiles\binfiles\ directory to the <ODI_HOME>\oracledi\bin\ directory.
6. On the Oracle Data Integrator machine, copy the files in the <ODI_HOME>\oracledi\biapps_odi\odifiles\libfiles\ directory to the <ODI_HOME>\oracledi\lib\ directory.
7. On the Oracle Data Integrator machine, copy the init<DB Version>.ora file in the <ODI_HOME>\oracledi\biapps_odi\dbfiles\ directory to the machine that hosts the data warehouse (for example, in the \$ORACLE_HOME/dbs directory).

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

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

Setting Up the Oracle Business Intelligence Applications and ODI Components

These topics explain how to set up the Oracle Business Intelligence Applications and Oracle Data Integrator components to create an operational E-LT environment. Perform the tasks in this topic in the sequence specified:

1. (Optional) How to Copy Over Installation Files
2. How to configure the ODI parameter file
3. How to start the ODI Agents
4. How to configure the ODI Master Repository
5. How to set up the ODI Topology
6. How to Import ODI work repository
7. How to Install Abap programs
8. How to generate the required ODI Scenarios
9. How to install the Schema Objects for the Oracle Business Analytics Warehouse
10. Setting Up Oracle BI Applications Configuration Manager

Notes

- Before following the steps in this topic, you must have installed the Oracle Business Intelligence Applications software and the Oracle Data Integrator software as specified in "Installing Oracle Business Intelligence Applications and Oracle Data Integrator Software".
- The installation and set up steps in this chapter assume the following recommended ODI configuration:
 - The ODI Master and Work Repositories are installed in a single database schema.
 - The Data Warehouse database schema is hosted on the same database instance as the ODI Repository.

- The ODI Work (temporary) database schema is hosted on the same database instance as ODI Repository.
- There is one SAP Application server for each environment (for example, the Production context, for more information about Contexts, refer to the ODI documentation).

(Optional) How to Copy Over Installation Files

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

To copy over installation files:

1. Copy the <OBIEE_HOME>\biapps7972\ repository\ EnterpriseBusinessAnalytics.rpd file from the Oracle Business Intelligence Applications installation machine to the machine that runs the BI Server.
2. Copy the <OBIEE_HOME>\biapps7972\ Catalog\EnterpriseBusinessAnalytics*. * files from the Oracle Business Intelligence Applications installation machine to the machine that runs the BI Presentation Services Catalog.

The Presentation Services Catalog is installed as a ZIP file named EnterpriseBusinessAnalytics.zip. Make sure that you un-zip the EnterpriseBusinessAnalytics.zip file into the <OBIEE_HOME>\biapps7972\ Catalog\EnterpriseBusinessAnalytics*. * directory.

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

How to configure the ODI parameter file

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

To configure the odiparams.bat file:

1. On the ODI machine, open the odiparams.bat (for Windows) or odiparams.sh (for UNIX) file in a text editor.

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

2. Set the value of the following parameter:

set ODI_SECU_URL=jdbc:oracle:thin:@<host>:<port>:<sid> Replace <host>, <port>, and <sid> with your database specific parameters.

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

How to configure the ODI Repository

To configure the ODI Repository, do the following:

- Import the ODI Master Repository (for more information, see "How to import the ODI Master Repository").
- Configure the Topology Manager connection to the ODI Master Repository (for more information, see "How to configure the Topology Manager connection to the ODI Master Repository").

- Create the ODI Work Repository (for more information, see "How to create the blank ODI Work Repository").

How to import the ODI Master Repository

To import the ODI Master Repository:

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

Figure 8.

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

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

Table 4 *Master Repository Import Wizard dialog fields*

Field	Description
Login	Do not change the default option (OracleDI Local Repository).
Driver	Specify 'oracle.jdbc.driver.OracleDriver'.
URL	Specify the JDBC URL to the Oracle Business Analytics Warehouse in the format <code>jdbc:oracle:thin:@<host>:<port>:<sid></code> . Replace <host>, <port> and <sid> with your database installation specific values. For example,

	'jdbc:oracle:thin:@US12345.us.company.com:1521:US12345'.
User	Specify ODI_REP_7972. This is the ODI Repository database user.
Password	Specify ODI_REP_7972. This is the default password for the ODI Repository database user.
Id	Specify a non-zero integer (the recommended value is between 50 and 60).
Use a zip file	Select this check box.
Zip	Specify <ODI_HOME>\oracledi\biapps_odi\odifiles\importfiles\ODI_REP_7972_master.zip
Technology	Select Oracle from the list.

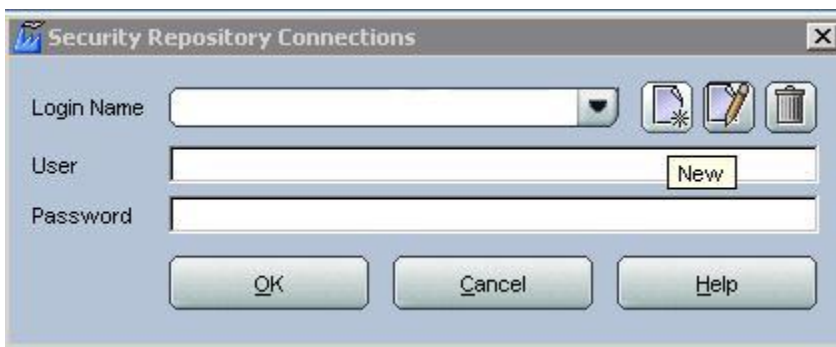
3. Click Test Connection to verify the details, then click OK to close the Information dialog.
4. Clicking OK will save the details and start the import.

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

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

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

Figure 9.



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

2. Click New to display the Repository Connections dialog

Figure 10.

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

Table 5. *Repository Connections dialog fields*

Field	Description
Login Name	Specify an identifier for the repository connection.
User	Specify SUPERVISOR. This is the ODI Administrator database user.
Password	Specify SUPERVISOR. This is the default password for the ODI Administrator database user.
User	Specify ODI_REP_7972. This is the ODI Repository database user name.
Password	Specify ODI_REP_7972. This is the default password for the ODI Repository database user name.
Driver List	Select 'Oracle JDBC Driver'.
Driver Name	Specify 'oracle.jdbc.driver.OracleDriver'.
URL	Specify the JDBC URL to the Oracle Business Analytics Warehouse in the format jdbc:oracle:thin:@<host>:<port>:<sid>. Replace <host>, <port> and <sid> with your database installation specific

	values. For example, 'jdbc:oracle:thin:@US12345.us.company.com:1521:US12345'.
Default Connection	Select the check box.

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

How to create the blank ODI Work Repository

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

To create the blank ODI Work Repository:

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

Figure 11.

The screenshot shows the 'Data Server: New' dialog box with the 'Definition' tab active. The fields are as follows:

- Name:** Empty text field.
- Technology:** Dropdown menu showing '<Undefined>'.
- Instance / dblink (Data Server):** Empty text field.
- Connection:** A group box containing:
 - User:** Empty text field.
 - Password:** Empty text field.
 - JNDI Connection:** A checkbox that is currently unchecked.
- Array Fetch Size:** Text field with the value '30'.
- Batch Update Size:** Text field with the value '30'.

At the bottom of the dialog are five buttons: OK, Cancel, Apply, Help, and Test.

Table 6. Data Server: New dialog, Definition tab fields

Field	Description
Name	Specify ORACLE_WORK_REPOSITORY. This is the identifier for the repository connection.
Technology	Select Oracle from the drop down list
Instance/dblink (Data Server)	Leave this field blank.
User	Specify 'ODI_REP_7972'. This is the repository database user name.
Password	Specify 'ODI_REP_7972'. This is default password for the repository database user name.
JNDI Connection	Clear this check box.
Array Fetch Size	Specify a value suitable to your environment (default value is 30).
Batch Update Size	Specify a value suitable to your environment (default value is 30).

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

Figure 12.

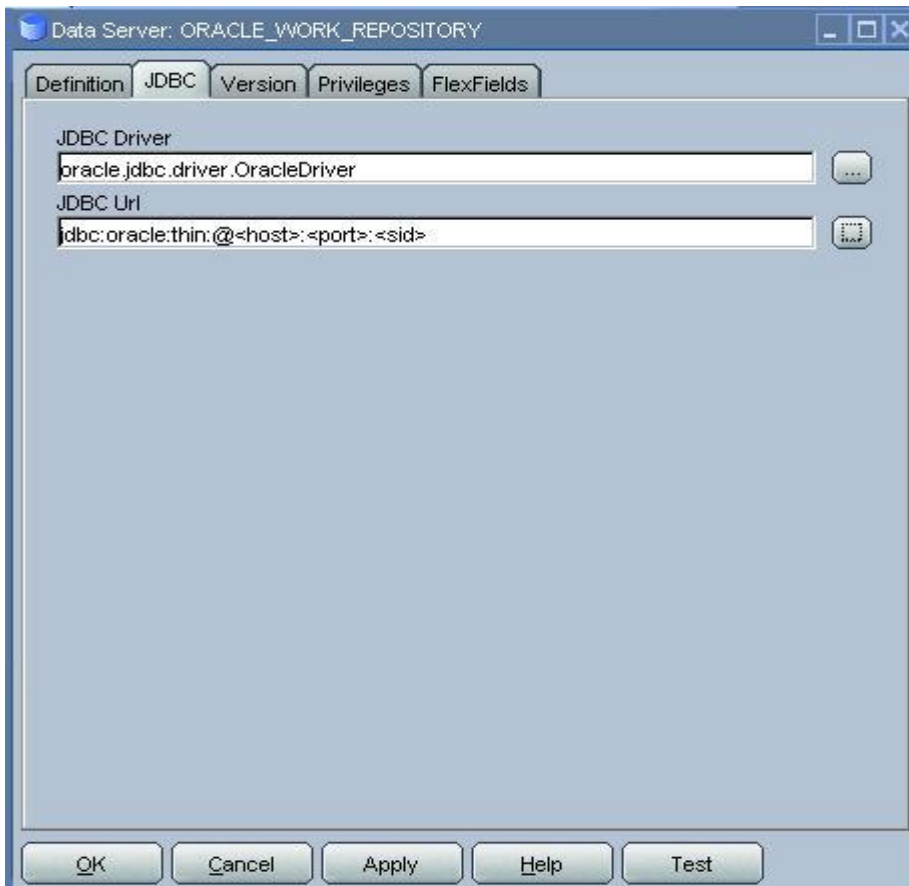


Table 7. Data Server: New dialog, JDBC tab fields

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

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

Figure 13.

The screenshot shows the 'Work Repository: New' dialog box. It has a title bar with standard window controls. Below the title bar are three tabs: 'Definition', 'Version', and 'Privileges'. The 'Definition' tab is selected. Inside the dialog, there are several input fields and buttons. The 'ID' field is empty. The 'Type' field is a dropdown menu currently showing 'Development'. The 'Name' field is empty. The 'Password' field is empty, and there is a 'Change password' button next to it. The 'External ID' field is empty, and there is a 'Connection' button next to it. At the bottom of the dialog are four buttons: 'OK', 'Cancel', 'Apply', and 'Help'.

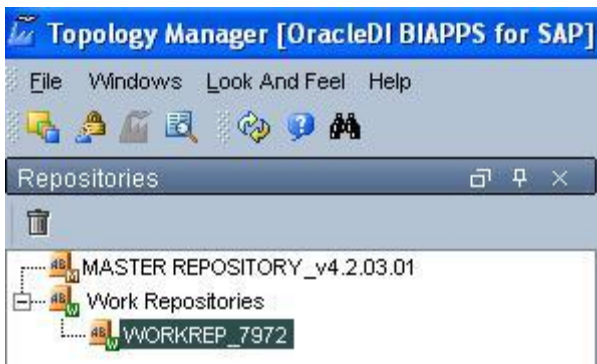
Table 8. *Work Repository: New dialog fields*

Field	Description
ID	Specify a unique ID (Between 500 and 600).
Type	Select Development from the drop down list.
Name	Specify WORKREP_7972. This is the name of the work repository.

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

Topology Manager displays the WORKREP_7972 repository in the Work Repositories list in the Repositories pane.

Figure 14.



How to set up the ODI Topology

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

- Set up the Oracle Data Servers (for more information, see "Setting up the Oracle Data Servers").
- Set up the Data Source Number (for more information, see "How to set up the Data Source Number").
- Set up the ODI Designer connection to the ODI Master Repository (for more information, see "How to set up the ODI Designer connection to the ODI Master Repository").
- Import the ODI Work Repository (for more information, see "How to import the Oracle BI Applications Work Repository").
- Set up the Agents (for more information, see "How to set up the Agents").

Setting up the Data Servers

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

- Locate the Physical Architecture tab in ODI Topology Manager (for more information, see "How to display the Physical Architecture tab in Topology Manager").

- Set up the ORACLE_BI_APPLICATIONS Data Server (for more information, see "How to set up the ORACLE_BI_APPLICATIONS Data Server").
- Set up the Physical Schema for the ORACLE_BI_APPLICATIONS Data Server (for more information, see "How to set up the Physical Schema for the ORACLE_BI_APPLICATIONS Data Server").
- Set up the FSMOUNT and SAP ABAP (for more information, see " How to set up the FSMOUNT ").
- Set up the Physical Schema for the SAP ERP Application Server (for more information, see "How to set up the Physical Schema for the FSMOUNT and SAP ABAP Data Server for SAP ABAP").
- Set up the ORACLE_WORK_REP Data Server (for more information, see "How to set up the ORACLE_WORK_REP Data Server").
- Set up the Physical Schema for the ORACLE_WORK_REP Data Server (for more information, see "How to set up the Physical Schema for the ORACLE_WORK_REP Data Server").
- Set up the Logical Data Servers (for more information, see "How to set up the Logical Data Servers").

How to display the Physical Architecture tab in Topology Manager

To display the Physical Architecture tab in Topology Manager:

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

How to set up the ORACLE_BI_APPLICATIONS Data Server

To set up the ORACLE_BI_APPLICATIONS Data Server:

1. Start Topology Manager and display the Oracle node in the Physical Architecture tab (for more information, see "How to display the Physical Architecture tab in Topology Manager").
2. Double click the ORACLE_BI_APPLICATIONS node to display the Data Server: <Name> dialog.

Figure 15.

The screenshot shows a Windows-style dialog box titled "Data Server: ORACLE_BI_APPLICATIONS". It has a tabbed interface with tabs for "Definition", "JDBC", "Properties", "Version", "Privileges", and "FlexFields". The "Definition" tab is selected. Inside the dialog, there are several input fields and a checkbox. The "Name" field is empty. The "Technology" dropdown menu is set to "Oracle". The "Instance / dblink (Data Server)" field is empty. The "User" and "Password" fields are empty. There is a checkbox for "UNDI Connection" which is unchecked. Below these, there are two numeric input fields: "Array Fetch Size" and "Batch Update Size", both set to "5000". At the bottom of the dialog, there are five buttons: "OK", "Cancel", "Apply", "Help", and "Test".

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

Table 9. *Data Server: ORACLE_BI_APPLICATIONS dialog, Definition tab fields*

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

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

Figure 16.

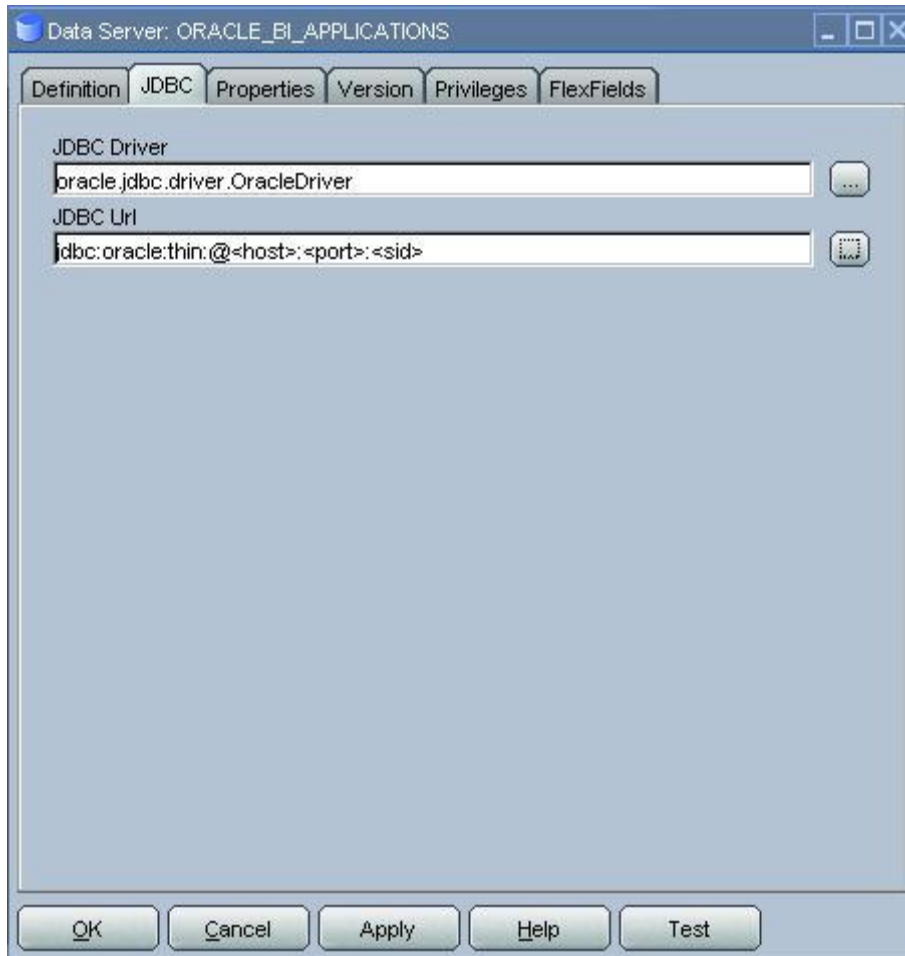


Table 10. *Data Server: New dialog, JDBC tab fields*

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

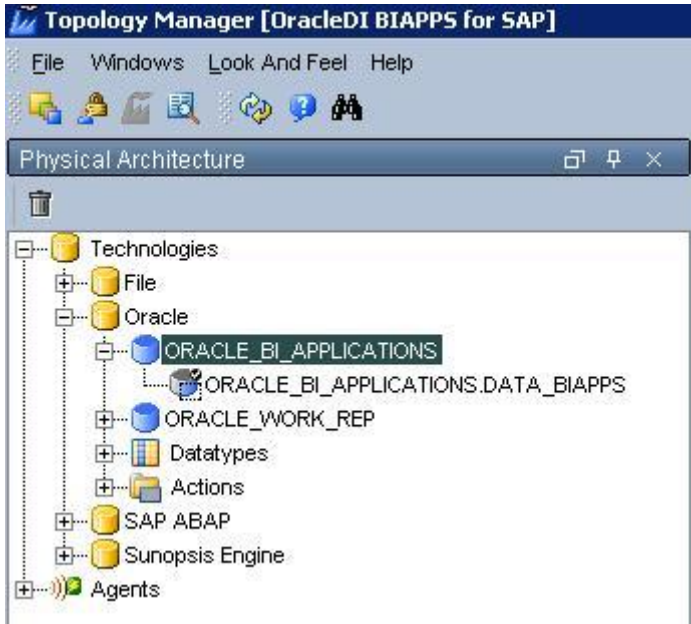
4. Click Test to display the Test Connection for: <Connection> dialog.
5. From the Agent drop down list, select Local (No Agent).
6. Click Test to verify the details.
7. Click OK on the Information dialog.
8. Click Apply, then click OK.

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

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

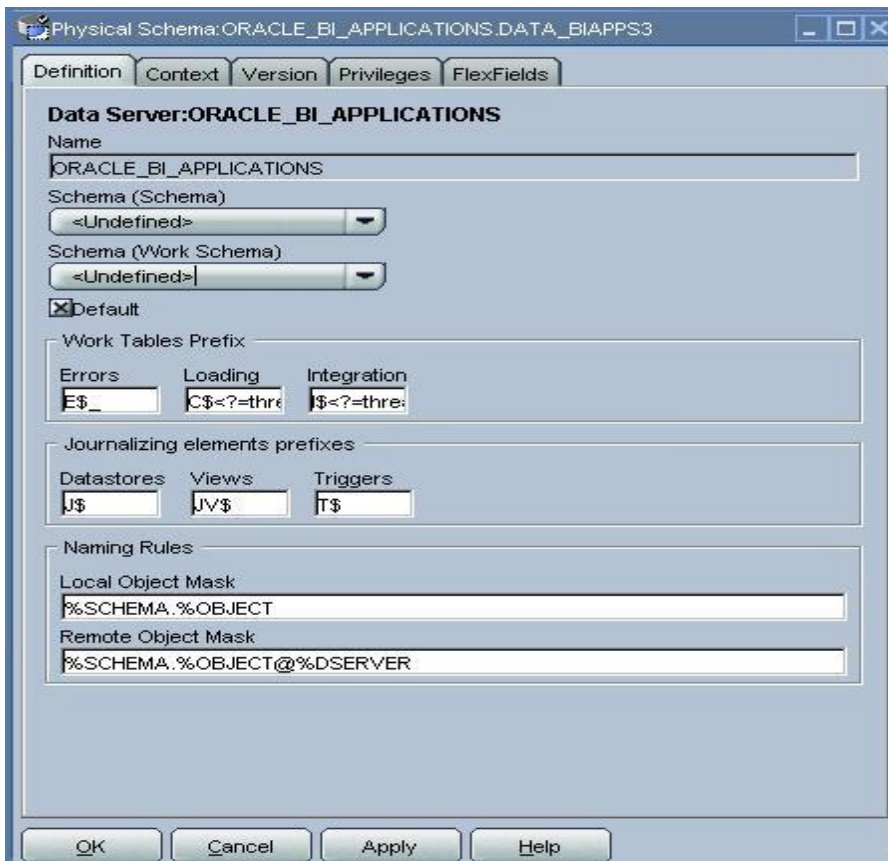
1. Start Topology Manager and display the Oracle node in the Physical Architecture tab (for more information, see "How to display the Physical Architecture tab in Topology Manager").
2. Expand the ORACLE_BI_APPLICATIONS node.

Figure 17.



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

Figure 18.



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

Table 11. ***Physical Schema: <Name> dialog, Definition tab fields***

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

Note: Do not change the other field values.

5. Click OK to save the details.

How to set up the FSMOUNT

FSMOUNT is 'Mount File system'. In FSMOUNT we can mount any server in the Network and write the data there. Data transfer is through a Shared Directory (recommended transfer method). BIApps utilizes FSMOUNT_DIRECT FTP transfer method of the LKM SAP ERP to Oracle (SQLLDR)

The LKM SAP ERP to Oracle (SQLLDR) requires a folder that is shared between the SAP system and the ODI agent. The SAP application server transfers the data by writing it out into a folder that is accessible from the SAP system and the ODI agent machine. This is typically done by sharing a folder of the ODI agent machine with the SAP system. Note that the shared folder does not necessarily have to be located on the ODI agent machine. A shared folder on a third machine is also possible, as long as the shared folder is accessible to both the ODI agent machine and the SAP system.

Note: For security reasons, the SAP folders should not be shared with the ODI agent. Share only folders of the ODI agent machine with the SAP system.

The shared folder must be accessible to SAP system and not just to the underlying operating system. This means that the folder needs to be declared in SAP transaction AL11 and the folder opens successfully in AL11. Please refer, to section "Setting up FS MOUNT" for the steps.

Validating SAP Privileges

This section describes how to test some of the key SAP privileges. Proceed with the subsequent steps in this guide only after successful validation of these tests. This validation is typically performed by your SAP Basis team.

Please find the steps below to validate whether a SAP user has appropriate dev rights and owns a dev license key:

1. Start SAPGUI.
2. Use the ODI SAP user and password to connect to the SAP system and client.
3. Go to transaction SE38. (SE38 is the ABAP editor which edit and executes SAP programs)
4. Enter any sample program name like ZSAP_TEST in the program name field.
5. Click Create.
6. Perform similar tests for the transaction SE37 and SE11.(SE37 is SAP is used to display the function module and SE11 to display/change/create database tables)

If a transaction allows the creation of a program without asking for any key or other authorization message, then the SAP user has validated that it has the appropriate dev rights and license key. Otherwise your SAP basis team needs to register the SAP user in service.sap.com to get the license key and a Basis person can help him with Dev rights.

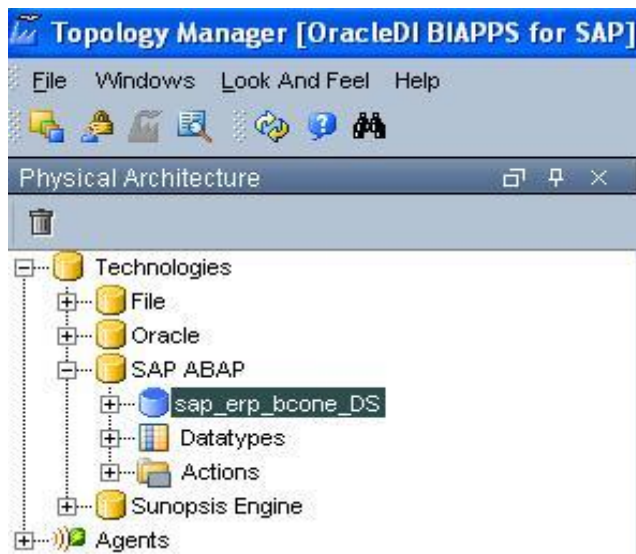
Setting up FS MOUNT

1. Create a folder in the machine where ODI is installed. share it and give folder permissions to specific users.
2. Add user -SIDADM & SAPSERVICESID to the systems administrator group.
3. Login into SAP system and run the transaction code 'AL11' to configure a new external directory in SAP system
4. Click on 'Configure' tab.
5. Fill in the required fields :
 - Directory Name : External drive path which needs to be configured (ex : \\<ODI_MACHINE_NAME>\oditest\)
 - Enter directory : Any name (ex: DIR_ODITEST)
 - Enter Server Name : all (if everyone has rights to access the directory. **Note** : 'all' keyword should be maintained in lower caps)
6. Click on 'SAVE' button. (**Note** : Do not Click 'ADD' or 'RETRIEVE' button)
7. Run the transaction code 'AL11' again.
8. Double Click on directory saved in step6 on 'AL11' transaction screen on sapgui to check the whether this directory is accessible.

How to set up the Physical Schema for the FSMOUNT and SAP ERP Data Server for SAP ABAP

1. In the Physical Architecture tree view, expand the Technologies node, select the SAP ABAP technology and then select the Data server "sap_erp_bcone_DS".
2. Set the parameters for this data server as follows:
 - Name: sap_erp_bcone_DS.
Note: Please do not change the default settings or names.
 - Host (Data Server): SAP ERP System IP Address or Hostname given to the host on which SAP is running.
 - User: SAP User Name given to a user for logging onto SAP system, as provided by the SAP Administrator.
 - Password: This user's SAP Password. This password is case-sensitive.

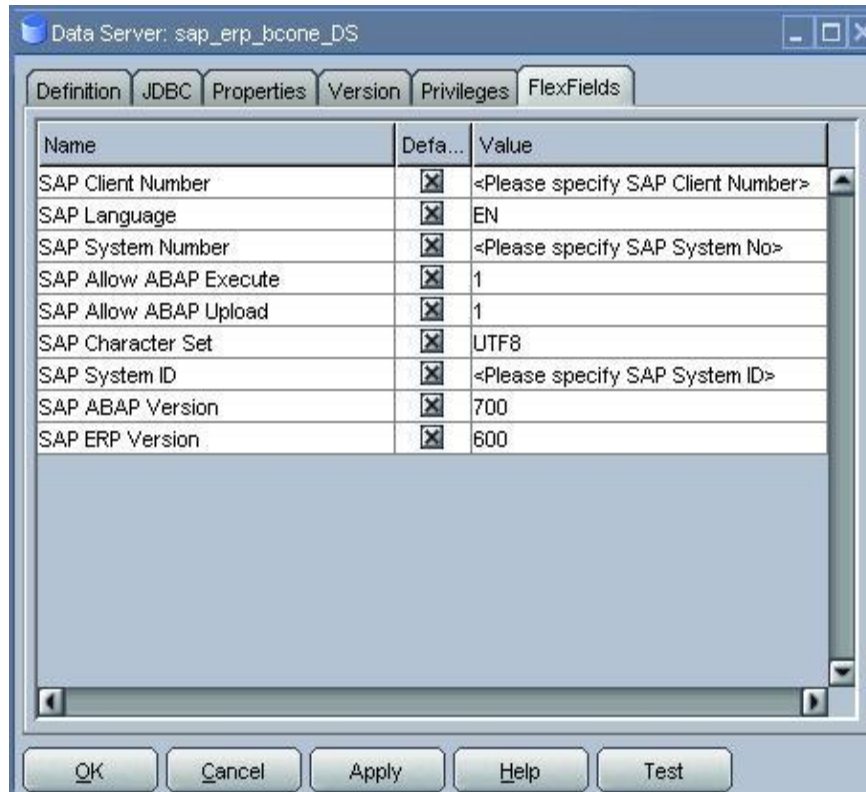
Figure 19.



3. Set the Flex field values for this data server in the Flex fields tab.

- SAP Language: Code of the language used when logging in. For example EN for English, DE for German.
- SAP Client Number: The three-digit number assigned to the self-contained unit which is called Client in SAP. A Client can be a training, development, testing or production client or represent different divisions in a large company.
- SAP System Number: The two-digit number assigned to a SAP instance which is also called Web Application Server or WAS.
- SAP System ID: The three-character, unique identifier of a SAP system in a landscape.
- SAP Character Set: The character set is only required if your SAP system is not a UNICODE system.
- SAP ERP Version: This is for the future release of SAP versions
- SAP ABAP Version: This is the SAP version i.e. either SAP ECC6.0 or 4.6C. By default it is 700 for SAP ECC6.0 and 46C for SAP 4.6C
- SAP Allow ABAP Upload: Setting of the Upload option to 0 will not upload ABAP Programme . Setting of the Upload option to 1 results in the uploading of ABAP Programme to SAP System.
- SAP Allow ABAP Execute: Setting of the Execute option to 0 will not execute ABAP Programme . Setting of the Execute option to 1 results in the execution of ABAP Programme to SAP System.

Figure 20.



Note: All the connection data listed above are mandatory and should be requested from the SAP Administrators. You may consider requesting support during connection setup from your SAP administrators.

4. Click OK.

Note: The Test button for validating SAP Connection definition is not supported for this connection. Please refer to Metalink Note 1384469 for authorizations..

Except for the data server name, all the parameters that you provide while defining the SAP Data server should be provided by the SAP Administrators. Refer the Oracle Data Integrator 10g Release 3 guide 'Getting Started with SAP ABAP ERP Adapter' for further information.

The SAP Adapter requires privileges to perform set up and execution operations. Please refer to myoracle support Note 1384469.1 for authorizations.

5. For setting up the FS MOUNT data server, go to Physical Architecture=> Technology=>File Double Click on File Server for SAP ABAP.

6. Do not specify the option for Host to the <HOST_Name> , < IP> and User& Password

Note: If user selects FTP file transfer mode then the FTP credentials will be required (Note: FTP transfer mode can be used up to 0.5 million records)

Figure 21.

Data Server: File Server for SAP ABAP

Definition JDBC Properties Version Privileges FlexFields

Name
File Server for SAP ABAP

Technology
File

Host (Data Server)

Connection

User

Password

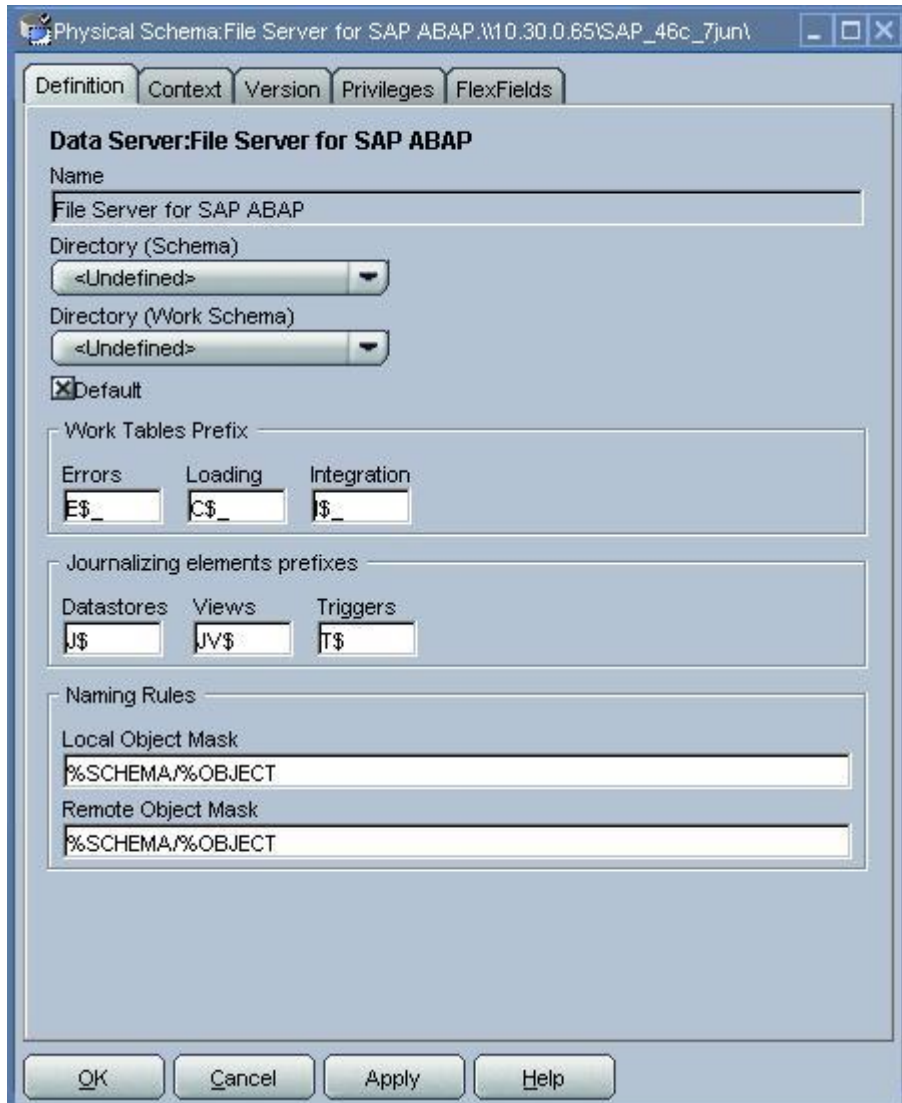
☐ JNDI Connection

Array Fetch Size
30

Batch Update Size
30

OK Cancel Apply Help Test

Figure 22.



7. Enter the information as given below:

Directory(Schema) : Specify the shared location which was shared for FSMOUNT for ODI server.

Directory(Work Schema): Specify the actual path of the shared folder for FSMOUNT for the ODI server.

How to set up the ORACLE_WORK_REP Data Server

To set up the ORACLE_WORK_REP Data Server:

1. Start Topology Manager and display the Oracle node in the Physical Architecture tab (for more information, see "How to display the Physical Architecture tab in Topology Manager").
2. Double click the ORACLE_WORK_REP node to display the Data Server: <Name> dialog.

Figure 23.

The screenshot shows the 'Data Server: ORACLE_WORK_REP' dialog box with the 'Definition' tab selected. The fields are as follows:

- Name:** ORACLE_WORK_REP
- Technology:** Oracle
- Instance / dblink (Data Server):** (empty)
- User:** (empty)
- Password:** (empty)
- JNDI Connection:** ☐
- Array Fetch Size:** 5000
- Batch Update Size:** 5000

Buttons at the bottom: OK, Cancel, Apply, Help, Test.

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

Table 12. *Data Server: ORACLE_WORK_REP dialog, Definition tab fields*

Field	Description
Name	Do not change the default value ORACLE_WORK_REP.
Technology	Do not change the default value Oracle.
Instance/dblink (Data Server)	Specify the name of the machine that hosts the ODI Master Repository. Use the Oracle SID name
User	Specify 'ODI_REP_7972'. This is the ODI Repository user name.
Password	Specify 'ODI_REP_7972'. This is default password for the ODI Repository user name.
Array Fetch Size	Specify a value suitable to your environment (a typical value is 30).
Batch Update Size	Specify a value suitable to your environment (a typical value is 30).

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

Figure 24.

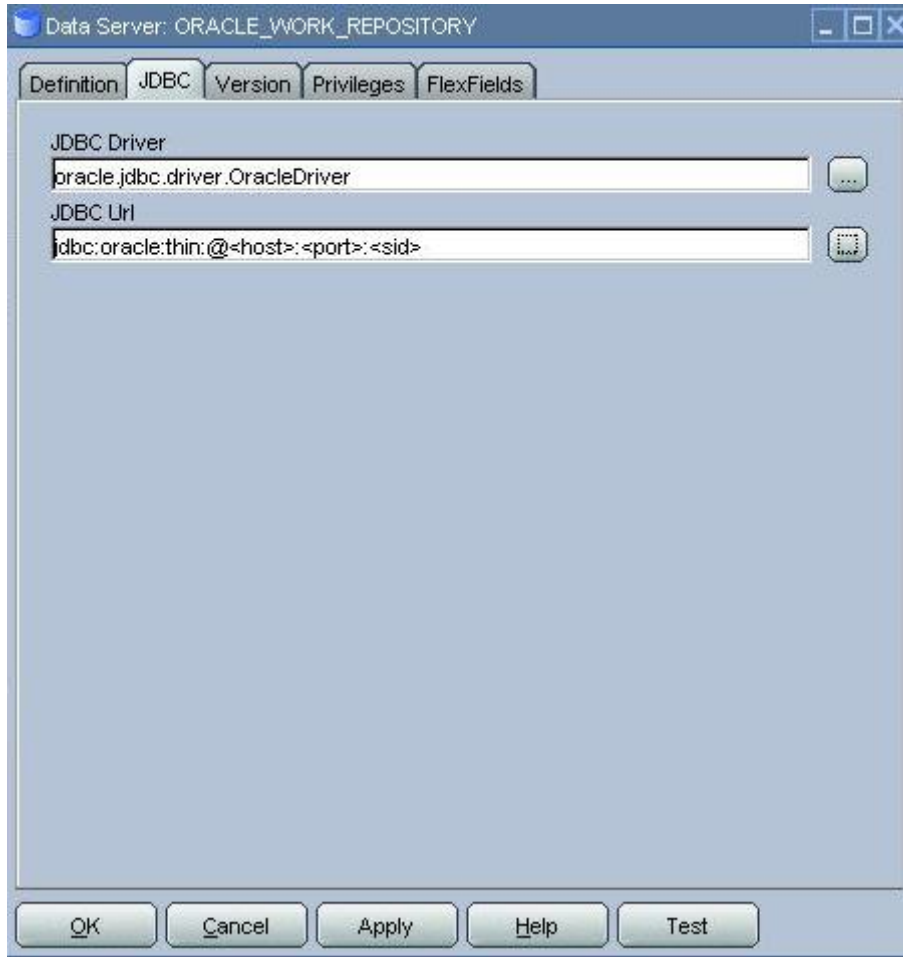


Table 13. *Data Server: ORACLE_WORK_REP dialog, JDBC tab fields*

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

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

How to set up the Physical Schema for the ORACLE_WORK_REP Data Server

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

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

3. Double click on ORACLE_WORK_REP.ODI_REP_7972 to display the Physical Schema: <Name> dialog.
4. Display the Definition tab, and enter the appropriate information, as described in the table below.

Figure 25

Physical Schema: ORACLE_WORK_REP.Schema

Definition Context Version Privileges FlexFields

Data Server: ORACLE_WORK_REP

Name
ORACLE_WORK_REP.Schema

Schema (Schema)
<Undefined>

Schema (Work Schema)
<Undefined>

☐ Default

Work Tables Prefix

Errors Loading Integration
E\$_ C\$_ I\$_

Journalizing elements prefixes

Datastores Views Triggers
J\$ JV\$ T\$

Naming Rules

Local Object Mask
%SCHEMA.%OBJECT

Remote Object Mask
%SCHEMA.%OBJECT@%DSERVER

OK Cancel Apply Help

Table 14. Physical Schema: <Name> dialog, Definition tab fields

Field	Description
Schema (Schema)	Make sure that ODI_REP_7972 is selected from the drop down list.
Schema (Work Schema)	Make sure that ODI_REP_7972 is selected from the drop down list.

Note: Do not change the other field values.

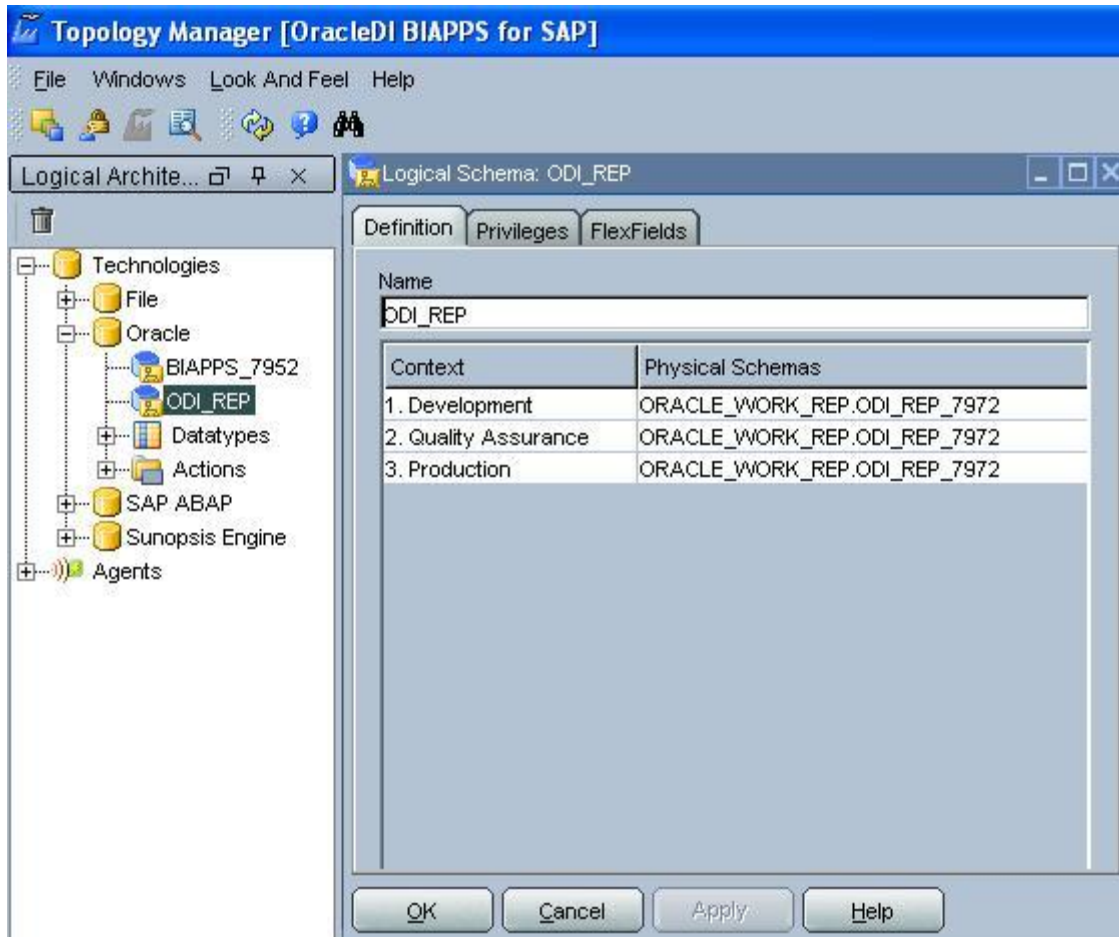
5. Click OK to save the details.

How to set up the Logical Data Servers

1. Start ODI Topology Manager.
2. Display the Logical Architecture tab.

3. Expand the Technologies node.
4. Expand the Oracle node to display the Logical Data Servers.

Figure 26.

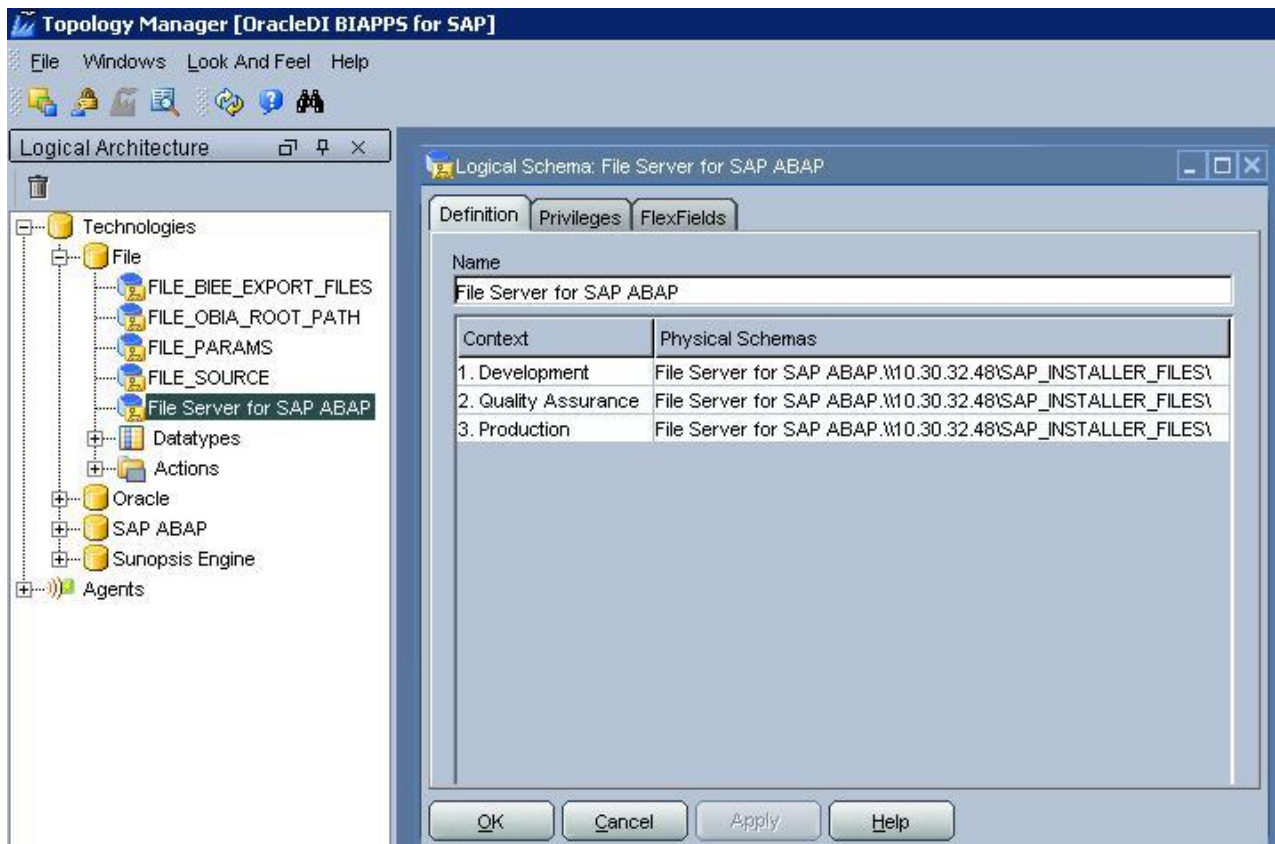


The Oracle node should contain the following Logical Data Servers:

- BIAPPS_7952
 - ODI_REP
5. Edit the BIAPPS_7952 Logical Data Server and make sure that for the appropriate Context (for example, Development), the value in the Physical Schemas column is set to 'ORACLE_BI_APPLICATIONS.DATA_BIAPPS'.
 6. Edit the ODI_REP Logical Data Server and make sure that for the appropriate Context (for example, Development), the value in the Physical Schemas column is set to 'ORACLE_WORK_REP.ODI_REP_7972'.
 7. Move to File under Technologies and select " File Server for SAP ABAP".

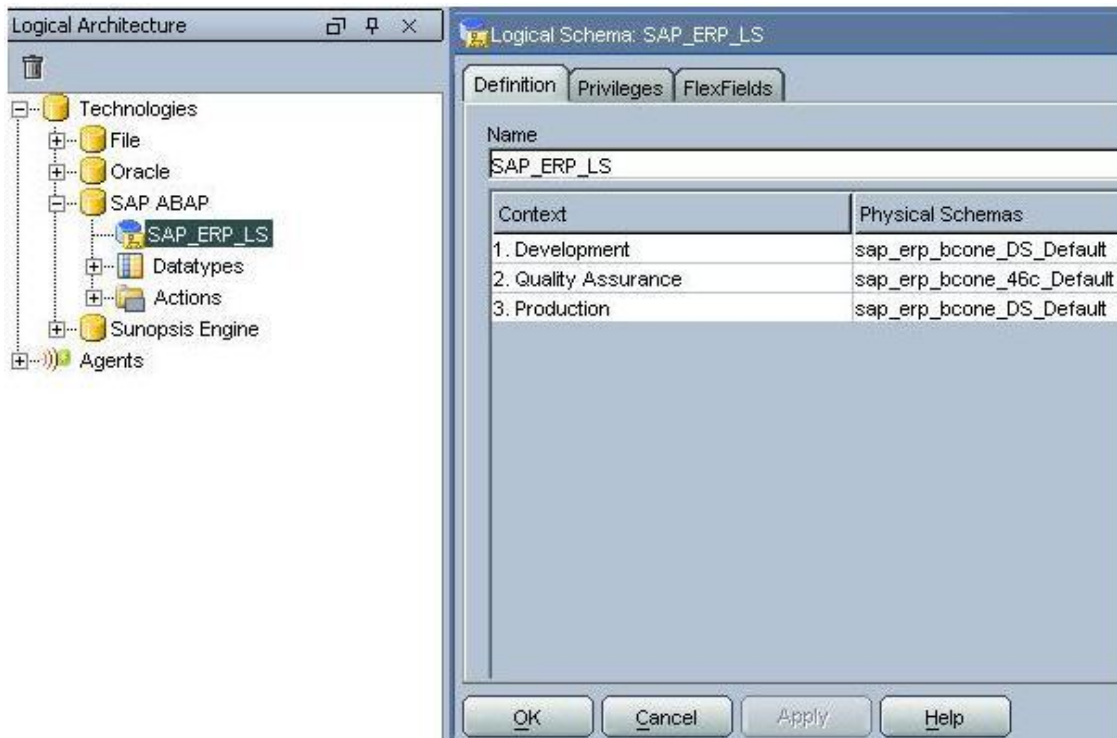
Configure Logical Schema called "File Server for SAP ABAP". This logical schema name is fixed and must be mapped to the physical schema created in the section " How to set up the Physical Schema for the FSMOUNT and SAP ERP Data Server for SAP ABAP".

Figure 27.



8. Expand SAP ABAP and double click on the SAP_ERP_LS logical schema for SAP ABAP in ODI Topology this must be mapped to the physical schema created for SAP ABAP.

Figure 28.

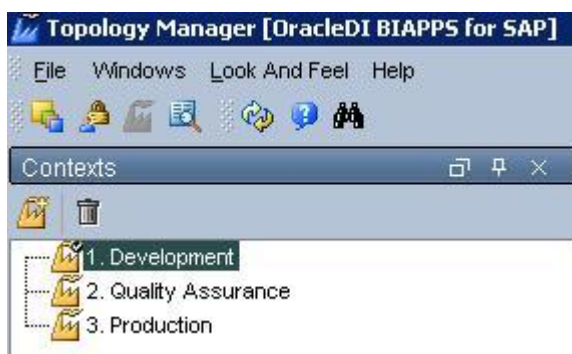


How to set up the Data Source Number

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

1. Start Topology Manager, and display the Contexts tab. The Contexts tab displays three contexts: Development, Quality Assurance, and Production.

Figure 29.



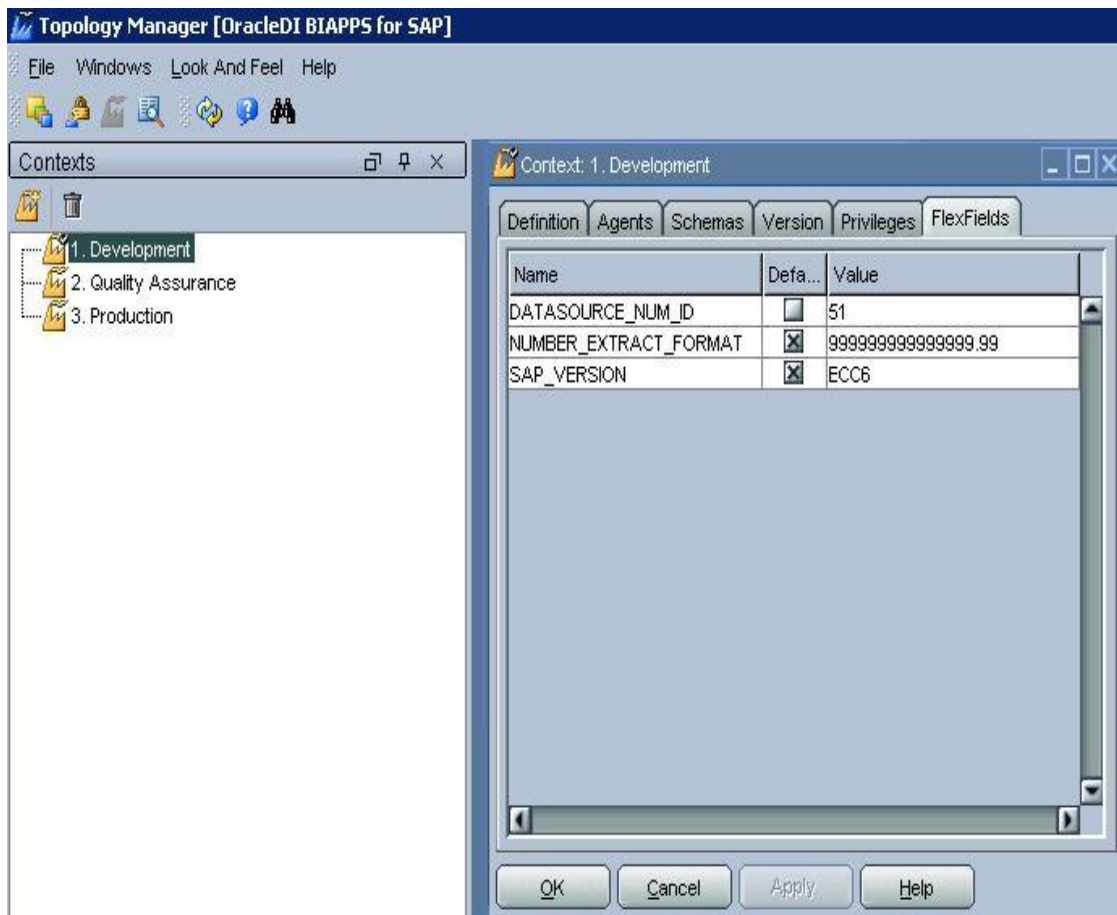
2. Edit each Context, and do the following:
 - e. Display the Flex fields tab, and enter the appropriate information, as described in the table below.

Table 15. *Context: <Name> dialog, Flex Fields tab fields*

Field	Description
DATASOURCE_NUM_ID	Specify the value according to your data source type. The default is 1 For SAP ECC6.0 the DATASOURCE_NUM_ID should be set as 51 and for SAP 4.6C it should be set as 50
SAP_VERSION	ECC6 or 46C
NUMBER_EXTRACT_FORMAT	Check the Default Check box

- a. Based on the Value entered in the SAP_VERSION flex field (either ECC6 or 4.6C) in the context the ODI variable SAP_VERSION would take the value from it and run the ECC6 or 4.6C packages accordingly when the user would select that particular context while running the ODI interfaces / master packages.

Figure 30.



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

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

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

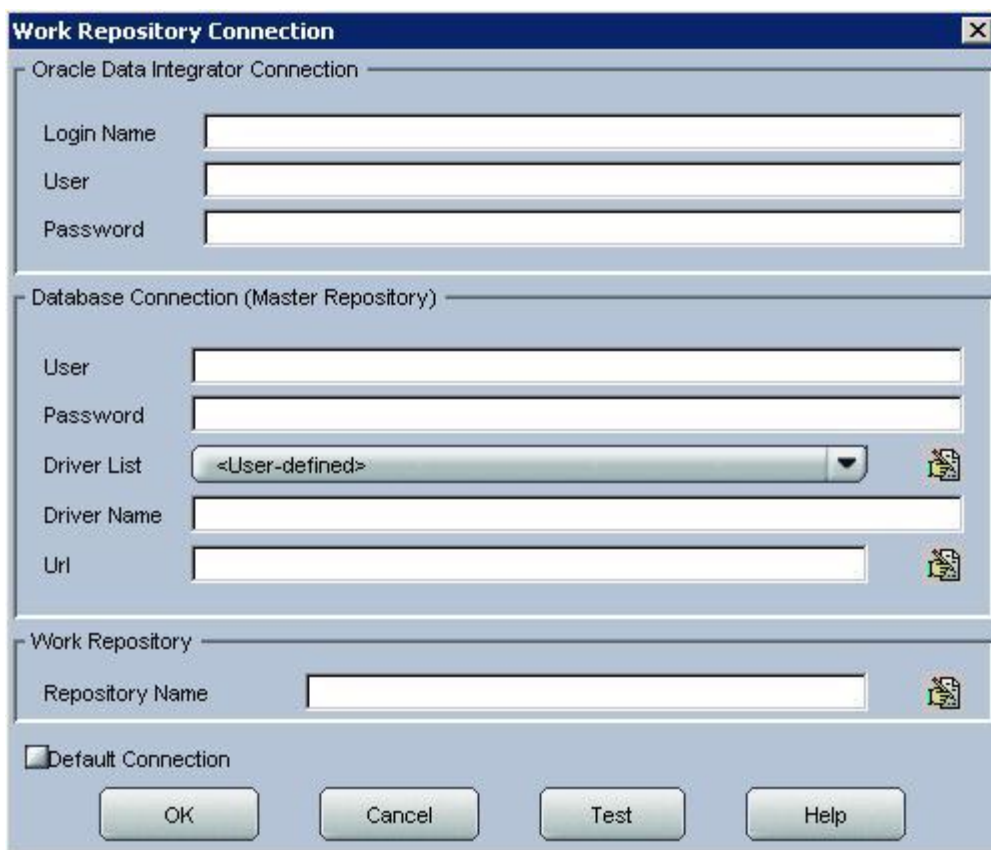
On Linux machine open Terminal, change current directory to <ODI_Home>/oracledi/bin/ then use ./designer.sh command to open Work Repository Connection dialog.

Figure 31.

The dialog box is titled "Oracle Data Integrator Login". It contains three input fields: "Login Name" (a dropdown menu), "User" (a text box), and "Password" (a text box). To the right of the "Login Name" field are three icons: a document with a star, a document with a pencil, and a trash can. At the bottom are three buttons: "OK", "Cancel", and "Help".

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

Figure 32.

The dialog box is titled "Work Repository Connection". It is divided into three sections. The first section, "Oracle Data Integrator Connection", has fields for "Login Name", "User", and "Password". The second section, "Database Connection (Master Repository)", has fields for "User", "Password", "Driver List" (a dropdown menu showing "<User-defined>"), "Driver Name", and "Url". The third section, "Work Repository", has a "Repository Name" field. At the bottom left is a checkbox labeled "Default Connection". At the bottom right are four buttons: "OK", "Cancel", "Test", and "Help".

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

Table 16. Work Repository Connection dialog fields

Field	Description
Login Name	Specify 'OracleDI 7972 Repository'.
User	Specify SUPERVISOR. This is the ODI Administrator user.
Password	Specify SUPERVISOR. This is the default password for the ODI Administrator user.

User	Specify ODI_REP_7972. This is the ODI Repository database user name.
Password	Specify ODI_REP_7972. This is the default password for the ODI Repository database user name.
Driver List	Select 'Oracle JDBC Driver'.
Driver Name	Specify 'oracle.jdbc.driver.OracleDriver'.
URL	Specify the JDBC URL to the Oracle Business Analytics Warehouse in the format jdbc:oracle:thin:@<host>:<port>:<sid>. Replace <host>, <port> and <sid> with your database installation specific values. For example, 'jdbc:oracle:thin:@US12345.us.company.com:1521:US12345'.
Repository Name	Select the icon with hand symbol and select WORKREP_7972.
Default Connection	Select this check box.

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

How to import the Oracle BI Applications Work Repository

This topic explains how to create the Oracle Business Intelligence Applications Work Repository by importing a zipped repository file into ODI. Importing this repository creates a project in ODI Designer called 'Oracle BI Applications 7.9.7.2'.

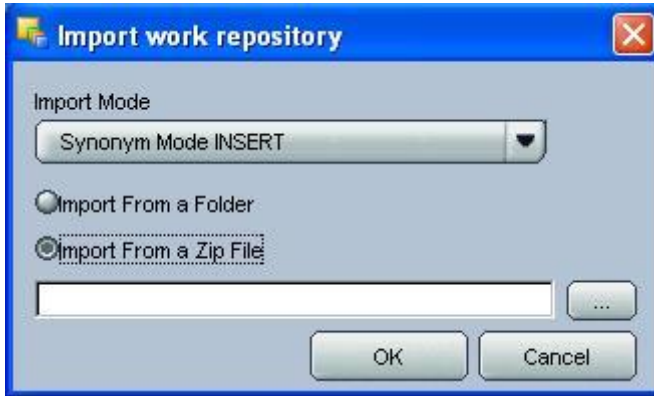
To import the Oracle BI Applications Work Repository:

1. In ODI Designer, select an appropriate Context from the Context drop down list. For example, if you are in a production environment, select Production.
2. Choose File, then Import, then Work Repository to display the Import work repository dialog.
3. In the 'Import work repository' dialog, enter the appropriate information, as described in the table below.

Table 17. Import work repository dialog fields

Field	Description
Import Mode	Select Synonym Mode INSERT from the drop down list.
Import From a Zip File	Select the Import from a Zip File radio button.
<Zip file path>	Use this field to specify the following path and ZIP file name: \$ODI_HOME/oracledi/biapps_odi/odifiles/importfiles/ODI_REP_7972_work.zip

Figure 33



4. Click OK to import the Work Repository.

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

How to start the ODI Agents

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

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

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

To start the ODI Agents on Windows:

1. On the ODI machine, open a command window and change directory to the \$ODI_HOME\oracledi\bin directory.
2. Execute the following commands one at a time:

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

Figure 34

```
AgentService.bat  
(c) Copyright Oracle. All rights reserved.  
Wrapper ! OracleBI BI Agent Scheduler WORKFLOW installed.  
Starting Agent Scheduler Service...  
SERVICE_NAME: SnpsAgentScheduler\WORKFLOW  
        TYPE               : 10  WIN32_OWN_PROCESS  
        STATE                : 2  START_PENDING  
                           (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)  
        WIN32_EXIT_CODE        : 0  (0x0)  
        SERVICE_EXIT_CODE     : 0  (0x0)  
        CHECKPOINT             : 0x0  
        WAIT_HINT              : 0x2d0  
        PID                   : 3464  
        FLAGS                   :
```


The above commands create the following agents:

Table 18. ODI Agents for Oracle Business Intelligence Applications

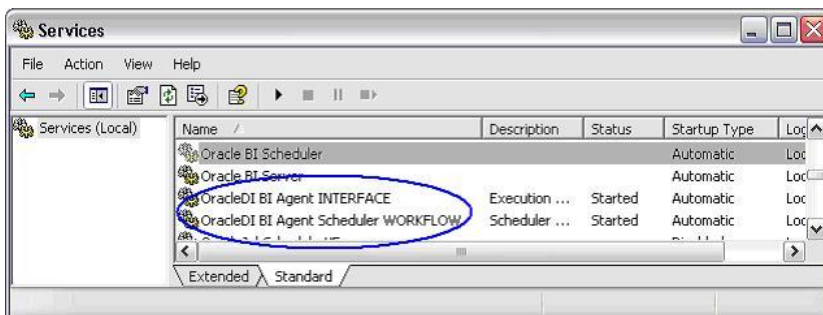
Name of Agent	Windows Service Name	Description	Mode	Port
WORKFLOW	OracleDI BI Agent WORKFLOW	Scheduler agent for ODI sessions.	Scheduler	20910
INTERFACE	OracleDI BI Agent INTERFACE	Listener agent for ODI sessions. When you use ODI Designer to execute Packages and Interfaces, you always specify the INTERFACE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

3. Display the Windows Services dialog.

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

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

Figure 35



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

To install and start the ODI Agents on Unix:

1. On the ODI machine, open a command window and change directory to the \$ODI_HOME/oracledi/bin directory.
2. Execute the following commands one at a time:

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

Notes

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

```
agent.bat "-v=5" > trace.txt
agent.sh -v=5 . trace.txt
```

- If you restart the machine on which you are running the ODI Agents, you must re-start the ODI Agents.

How to set up the Agents

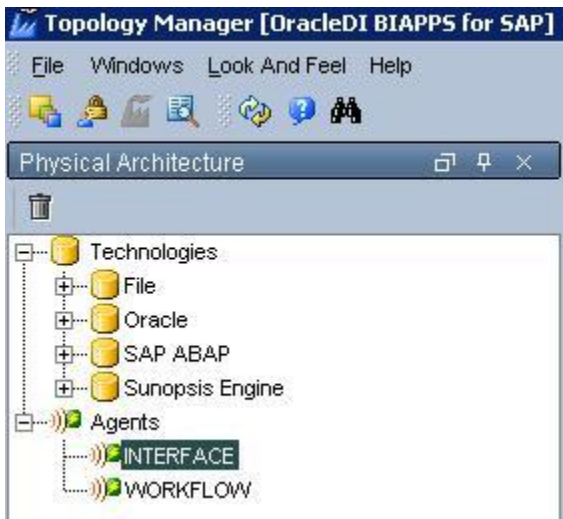
This topic explains how to set up the INTERFACE agent and the WORKFLOW agent in Topology Manager. To set up the Agents, do the following:

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

How to set up the INTERFACE Agent

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

Figure 36.



2. Double-click the INTERFACE node to display the Agent: <Name> dialog.

Figure 37.

The screenshot shows a Windows-style dialog box titled "Agent: INTERFACE". It has five tabs: "Definition", "Load balancing", "Version", "Privileges", and "FlexFields". The "Definition" tab is selected. Inside the dialog, there are four input fields: "Name" with the value "INTERFACE", "Host" which is empty, "Port" with the value "20911", and "Maximum number of sessions" with the value "3". Below these fields are three buttons: "Scheduling information", "Update Scheduling", and "Test". At the bottom of the dialog are four buttons: "OK", "Cancel", "Apply", and "Help".

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

Table 19. Agent: <Name> dialog, Definition tab fields

Field	Description
Name	Do not change the default value INTERFACE.
Host	The host name of the machine that is running the Agent, typically the ODI installation machine. For example, localhost, or US12345.us.company.com.
Port	Do not change the default value. If you encounter a port conflict when you run the ODI Agents, you might need to use a different port number (for more information, see How to Resolve Conflicts in ODI Agent Port Numbers.
Maximum number of sessions	The maximum number of interfaces that can run in parallel (default is 3)..

Click Test to verify the details.

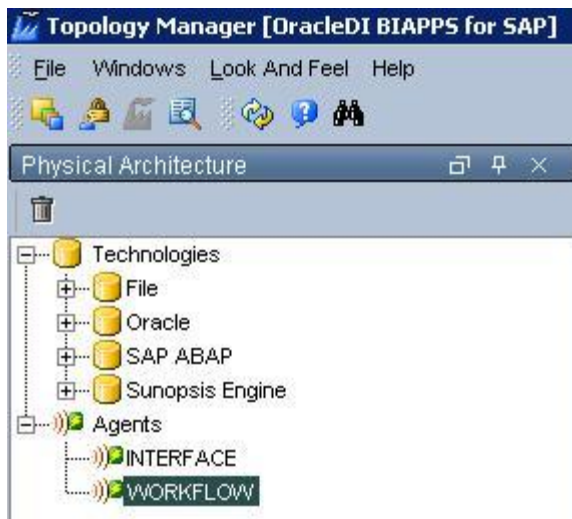
4. Click OK to close the Test window.
5. Click Apply, then click OK to save the details.

How to set up the WORKFLOW Agent

Setting Up the Oracle Business Intelligence Applications and ODI Components

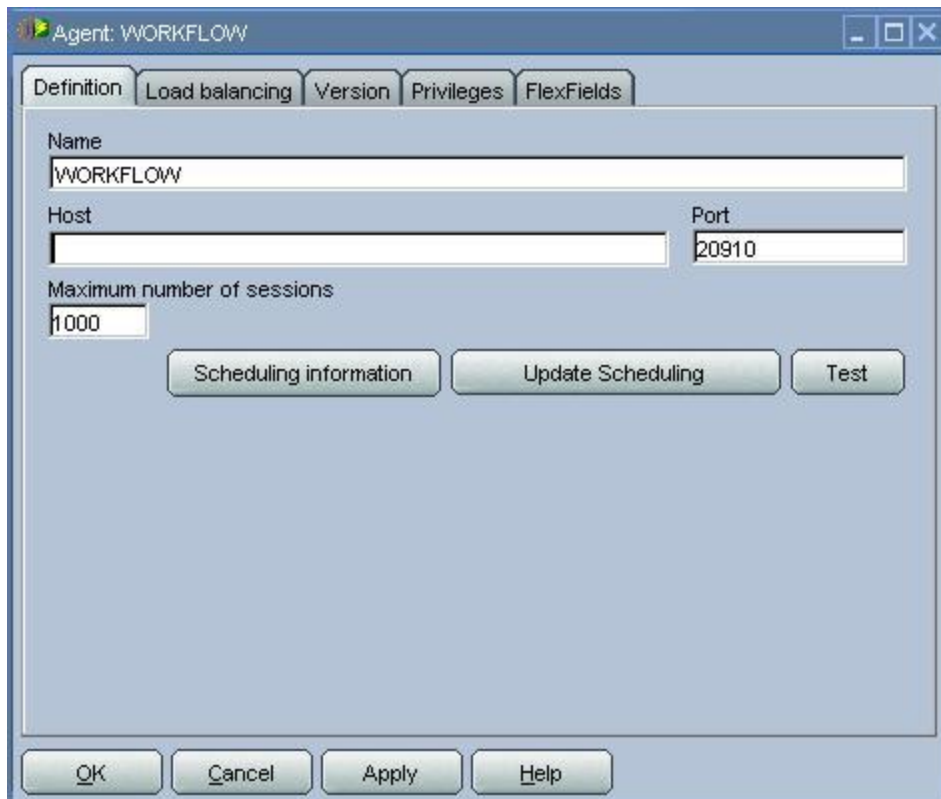
1. Start Topology Manager, display the Physical Architecture tab, and expand the Agents node.

Figure 38.



2. Double-click the WORKFLOW node to display the Agent: <Name> dialog.

Figure 39.



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

Table 20. **Agent: <Name> dialog, Definition tab fields**

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

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

Add the Open Tool

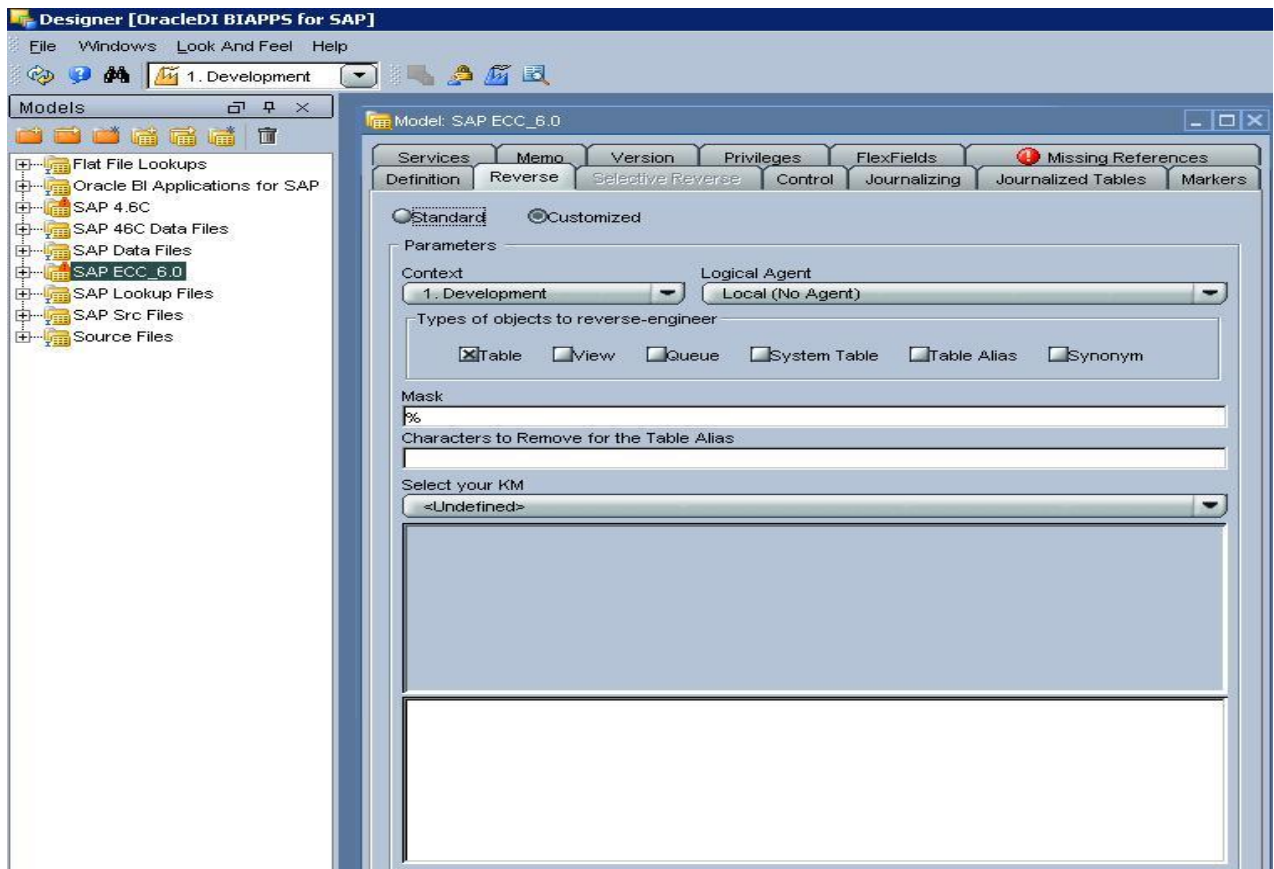
The complete process of installing and adding an Open Tool to Oracle Data Integrator is described in the Oracle Data Integrator Tools Reference guide. This topic details only the SAP ABAP specific steps.

1. Connect to Designer.
2. Select File > Add/Remove Open Tools. . .
3. In the Add/remove Open Tools window, enter the following name in the Open Tool class name field:
oracle.odl.sap.km._OdiSapAbapExecute
4. Click Add Open Tool.
5. Click OK.

How to install Base programs in SAP using RKM

In Designer display the Models view that contains the work repository objects. After importing the master and work repository in the data models (SAP ECC_6.0 or SAP 4.6C) designer shows warning icons on the "Missing References" tab in the Screen shot below.

Figure 40.



1. Double click on the model to open it up for editing
2. Go to "Reverse" tab, under the "Select Your KM" drop down Box, select the RKM then Click Apply.
3. As you do the above, you will see the warning icon disappeared.
4. Double click on the Model SAP ECC_6.0 to open it in edit mode.
5. On the reverse tab set the mentioned options as below:

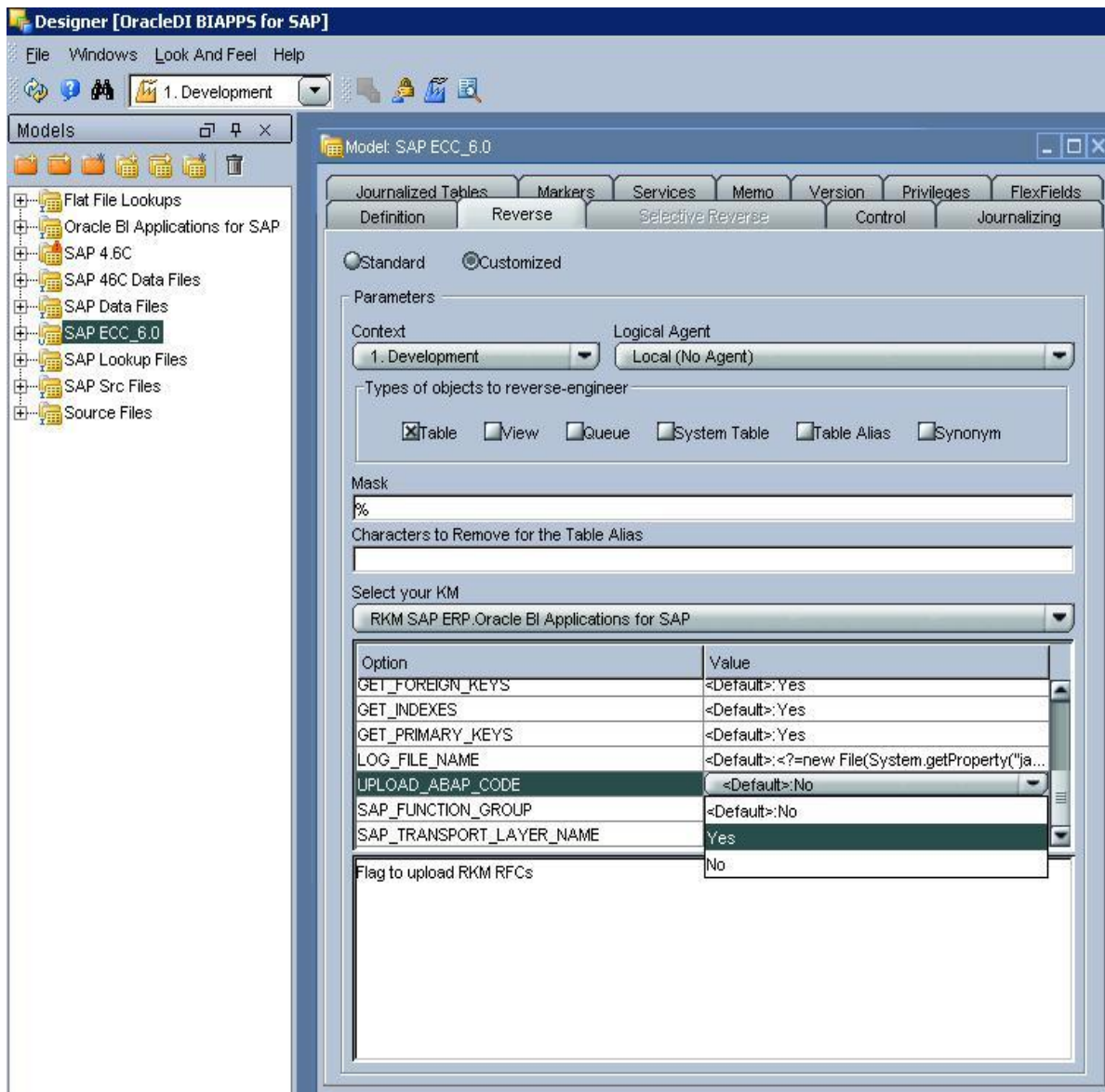
USE_GUI: No

SAP_TABLES_NAME: <Any_table_name>Example:MARA

UPLOAD_ABAP_CODE: Yes

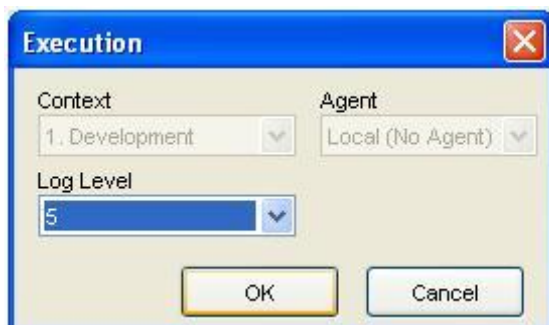
Leave rest of the options as default.

Figure 41



Click on the reverse button at the bottom. This will popup the execution window. Click on OK.

Figure 42



Check the operator log to make sure the operation has finished successfully. The process can take 1-5 minutes depending on SAP server and DB performance. After the execution has finished change the RKM options to default in the model.

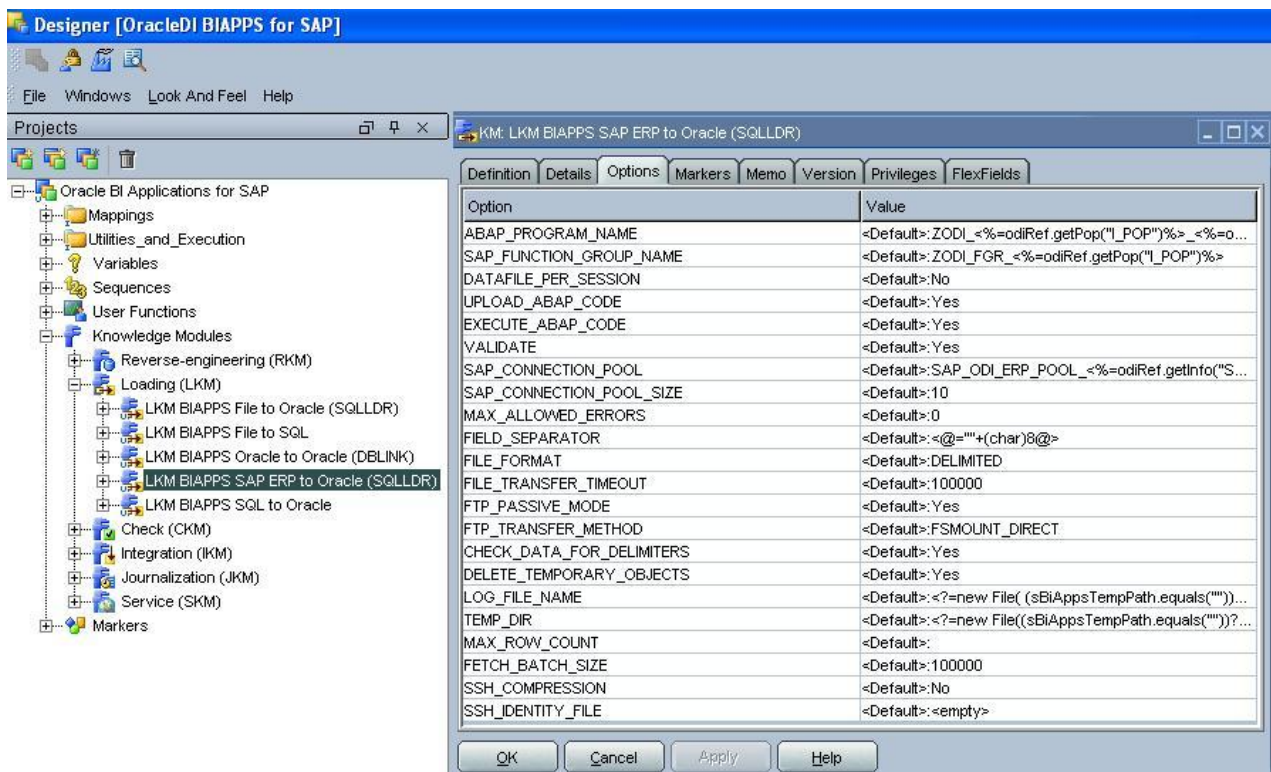
Please refer to the Getting started with SAP Adapter guide for more details about what happens in the background and why we need these steps.

Validating the LKM Option for file transfer

FS MOUNT option can be used for writing the files from SAP into a shared directory.

FTP_TRANSFER_METHOD parameter of LKM SAP ERP to Oracle (SQLLDR) has to be set to 'FSMOUNT_DIRECT'. Please validate the following options and do not change the default options.

Figure 43 .



Indexes Recommendation for Performance improvement on SAP Server

Before executing Master Package for SAP Adaptor the following indexes are recommendation for Performance improvement during extraction from SAP.

The SAP BASIS administrator will create these indexes in required SAP system.

Figure 44

Serial No.	OBIEE Table Name	Load Type	SAP Table Name	List of Fields in Index in Order
1	W_AP_XACT_FS	Full Load	BSAK	MANDT, AUGDT,BSTAT
2	W_AR_XACT_FS	Full Load	BSAD	MANDT, AUGDT,BSTAT
3	W_GL_COGS_FS	Full Load	BSIS	MANDT, BLART,BSCHL
4	W_GL_REVN_FS	Full Load	BSIS	MANDT,BUKRS,HKNOT
5	W_GL_OTHER_FS (BSAS)	Full Load	BSAS	MANDT,BUDAT
6	W_GL_REVN_FS	Incremental Load	BSIS	MANDT,CPUDT
7	W_ACCT_BUDGET_FS	Incremental Load	FMBDP	MANDT,CPUDT
8	W_GL_REVN_FS(ASSET)	Incremental Load	ANEK	MANDT,CPUDT
9	W_SALES_CYCLE_LINE_F_HOLD_TMP	Full Load	VBAK	LIFSK,FAKSK
10	W_SALES_CYCLE_LINE_F_HOLD_TMP	Incremental Load	VBUK	VBELN,CMGST
11	W_SALES_INVOICE_LINE_FS	Full Load	BSAD	SHKZG, VBELN, CPUDT, BSCHL
12	W_SALES_ORDER_LINE_FS	Full Load	VBAK	AUDAT
13	W_PURCH_COST_FS	Full Load	EKKO	BSTYP, MEMORY, AEDAT
14	W_PURCH_COST_FS	Incremental Load	KEPO	BSTYP, PSTYP

How to Upload SAP ABAP programs (PART 1 of 2)

This topic explains how to install the pre-defined SAP ABAP programs in the SAP source system. This is applicable for both ECC6.0 and 4.6c system.

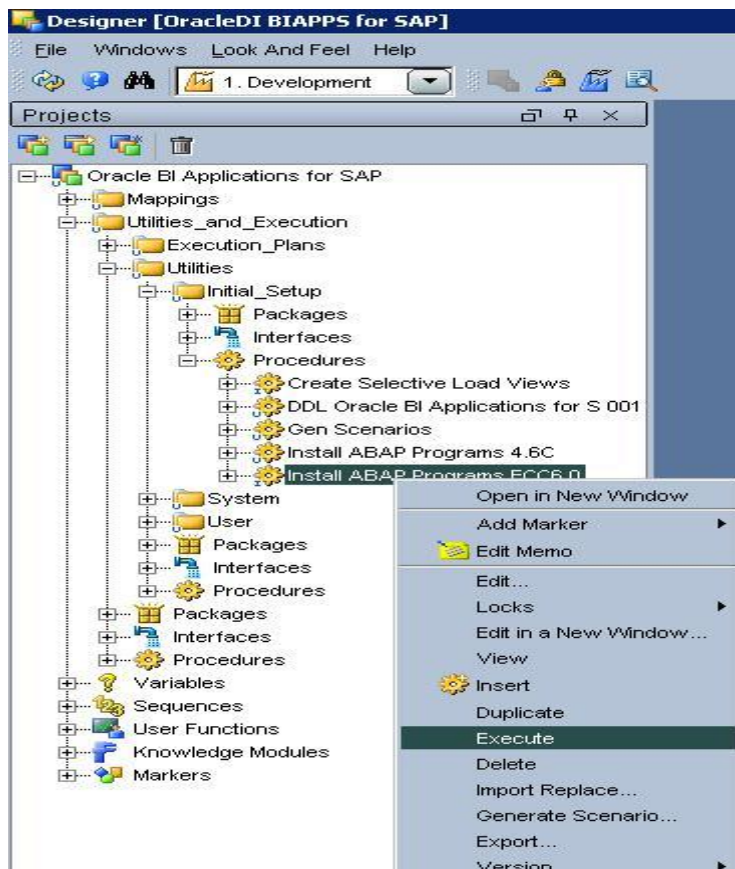
1. In ODI Designer, log in as SUPERVISOR, and display the Projects View.
2. Expand the Oracle BI Applications for SAP project.

Figure 45.



3. Select Utilities_and_Execution, then Utilities, then Initial_Setup, then Procedures.
4. Based on the version of your SAP .Right click on 'Install ABAP Programs <SAP_VERSION>' code and select execute to start installing ABAP programs required for SAP Adaptor of Oracle Business Intelligence Applications.

Figure 46.



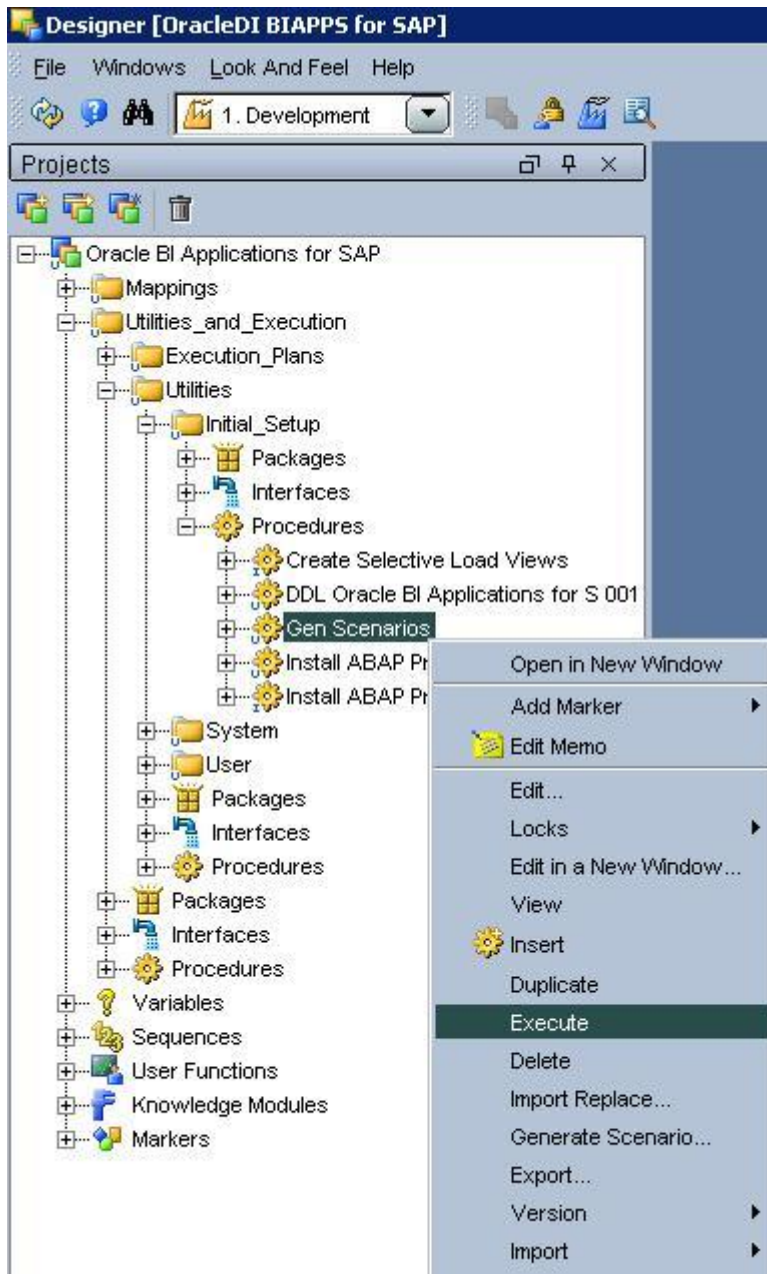
Note:For deploying in QA and Production environments, this procedure should not be executed as the code should be transported from SAP development system to SAP QA and Production system.

How to generate the required ODI Scenarios

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

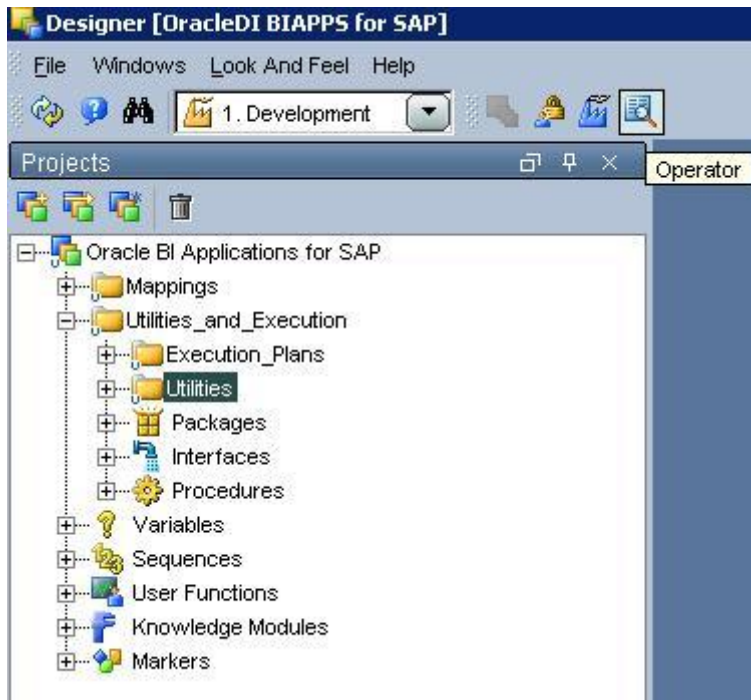
1. In ODI Designer, log in as SUPERVISOR, and display the Projects view.
2. Expand the Oracle BI Applications for SAP project.
3. Select Utilities_and_Execution, then Utilities, then Initial_Setup, then Procedures.
4. Right click on Gen Scenarios and select Execute to start creating the scenarios required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

Figure 47.



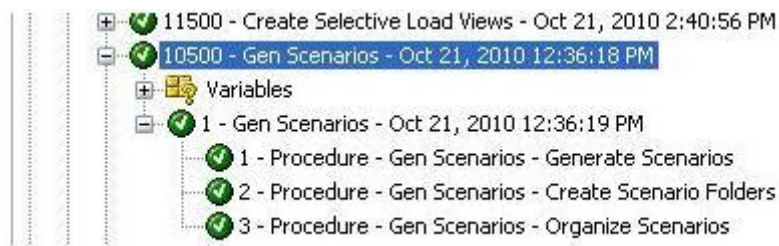
5. Use the Execution dialog to specify the appropriate context in the Context field, and specify Local (No Agent) as the Agent value. You use ODI Operator to monitor executed processes.
6. To display ODI Operator, click on the Operator icon on the ODI Designer tool bar.

Figure 48.



7. In ODI Operator, display the Session List tab, expand the Sessions node, and verify the session running the Gen Scenario procedure.
8. Make sure that all of the procedures on the Gen Scenarios node execute successfully.

Figure 49.



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

This section explains how to install the Schema Objects required by the Oracle Business Analytics Warehouse. To install the Schema Objects, do the following:

1. Generate the required procedures (for more information, see "How to Generate the Data Warehouse DDL Procedure").
2. Execute the required procedures (for more information, see "Execute the Data Warehouse DDL Procedure").
3. Execute the Load Control Seed Data Package (for more information, see "Running Load Control Seed Data Package").
4. Verify the INDEX_TABLESPACE setting in ODI (for more information, see "Verifying the INDEX_TABLESPACE Setting").

Notes

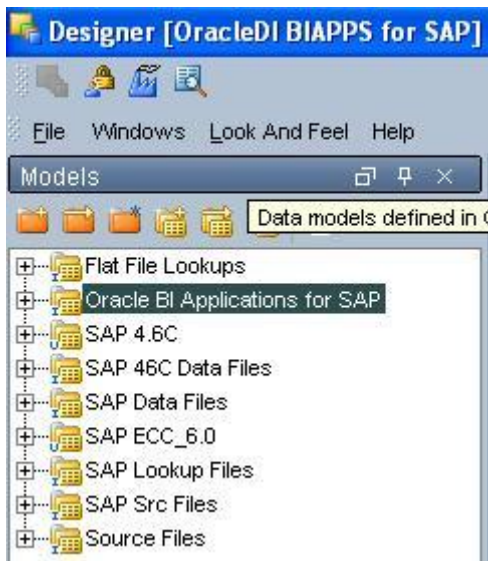
- Before performing the following tasks, make sure that the Oracle Business Analytics Warehouse schema is empty.
- To maximize performance, Oracle recommends that you implement range partitioning for fact tables before you create the Oracle Business Analytics Warehouse Target Schema Objects. For more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

How to Generate the Data Warehouse DDL Procedure

To generate the required DDL Procedure:

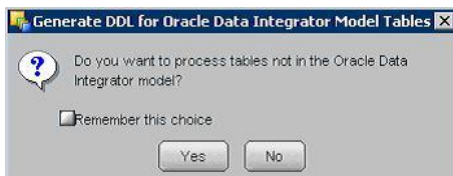
1. In ODI Designer, log in as SUPERVISOR, and display the Models view.

Figure 50



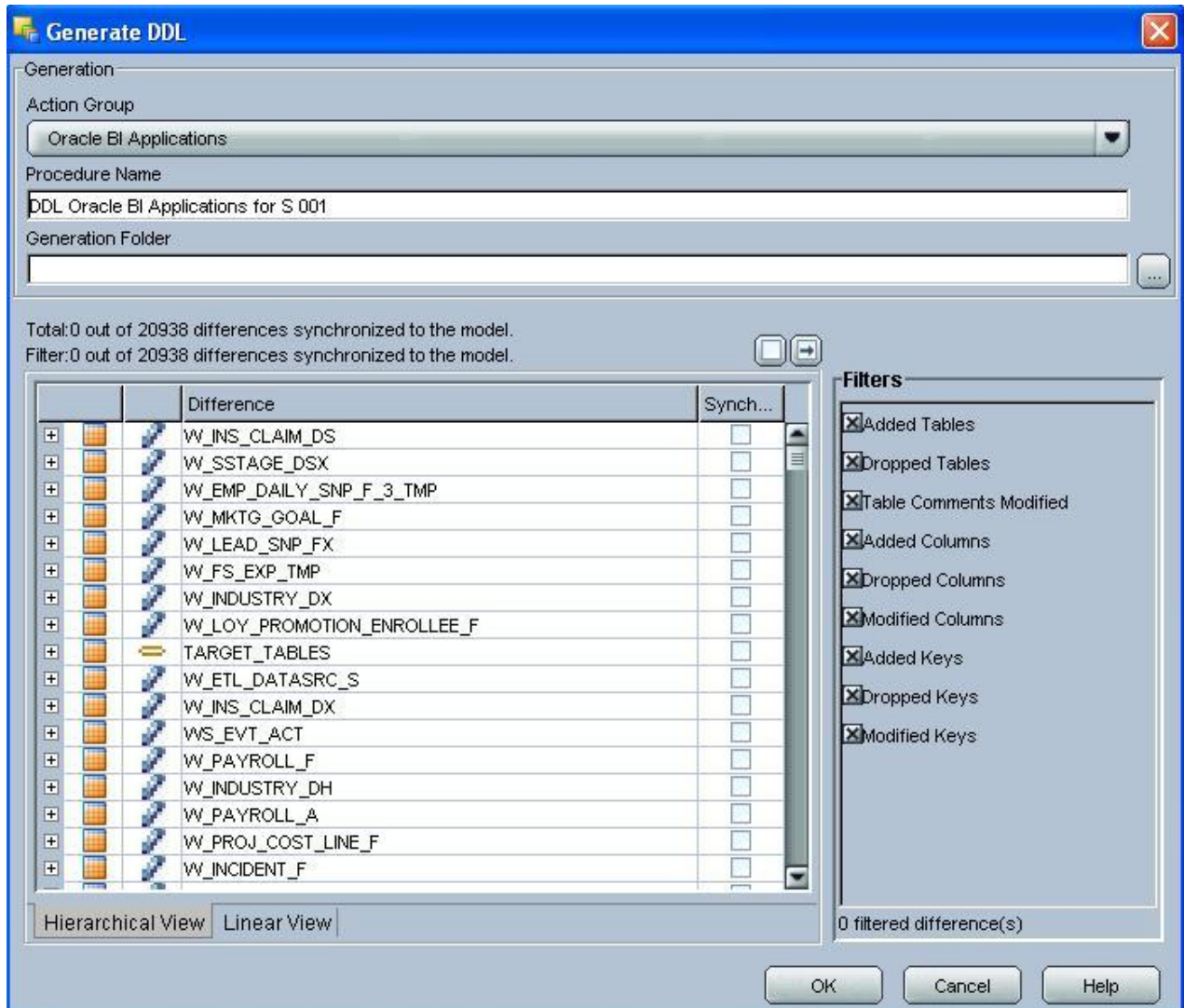
2. Right-click on the Oracle BI Applications for SAP node, then select Generate DDL from the right-click menu to display the Generate DDL for Oracle Data Integrator Model Tables dialog.

Figure 51



3. Click No to display the Generate DDL dialog.

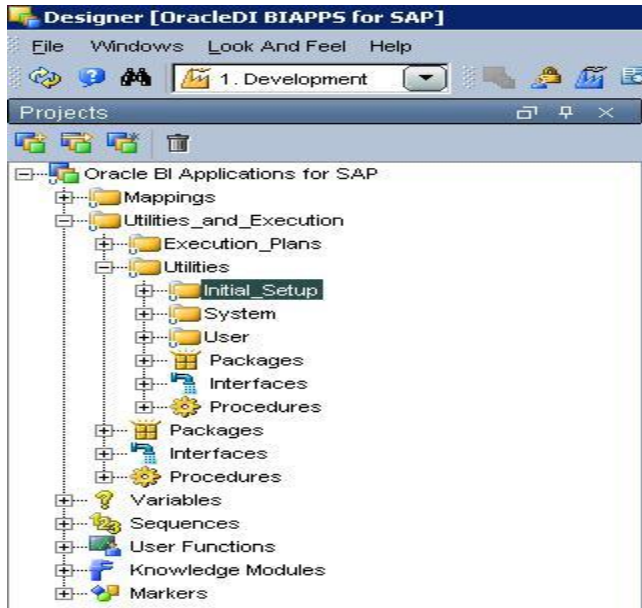
Figure 52



To select all objects for generation, click the right arrow (→) in the top right hand corner of the list of objects. A right arrow (→) is displayed in the Synchronization column for each row in the objects table, to indicate that all objects are selected.

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

Figure 53



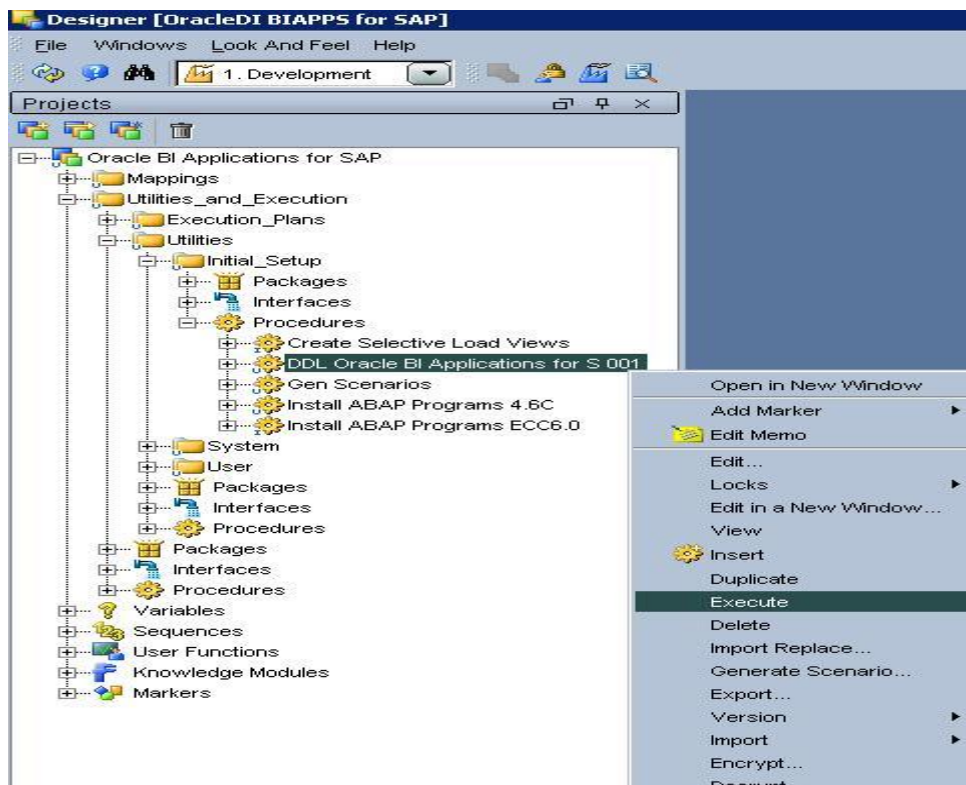
5. Click OK to start the generation of DDL procedures.

Execute the Data Warehouse DDL Procedure

To execute the required procedures:

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

Figure 54

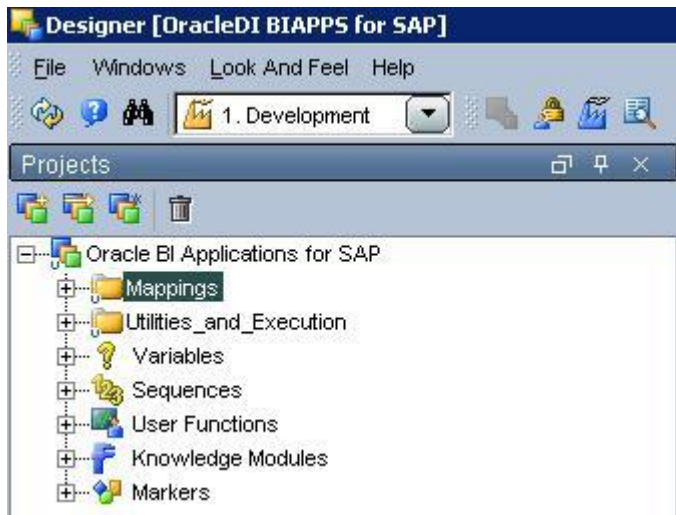


4. Right click on 'DDL Oracle BI Applications for SAP' and select Execute.

5. Right click on 'Create Selective Load Views' and select Execute to display the Execute dialog.
6. Use the Execution dialog to specify the appropriate context in the Context field, and specify INTERFACE as the Agent value. You use ODI Operator to monitor executed processes.

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

Figure 55



Running Load Control Seed Data Package

To run the Load Control Seed Data Package:

1. In ODI Designer, log in as SUPERVISOR, and display the Projects view.
2. Expand the Oracle BI Applications for SAP project.
3. Select Utilities_and_Execution, then Utilities, then Initial_Setup, then Packages.
4. Right click on 'Load Control Seed Data' and select Execute.
5. You use ODI Operator to monitor executed processes.

Verifying the INDEX_TABLESPACE Setting

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

To verify the INDEX_TABLESPACE setting:

1. In ODI Designer, log in as SUPERVISOR, and display the Models view.
2. Edit the Oracle BI Applications for SAP project, to display the Model:<Name> dialog.
3. Display the FlexFields tab.
4. Make sure that the INDEX_TABLESPACE FlexField is set with a value BIAPPS_INDEX.
5. This FlexField is used by Index Management procedures in related Knowledge Modules (KM).

Setting Up Oracle BI Applications Configuration Manager

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

- Create and manage Execution Plans for E-LT executions.
- Set E-LT parameters.

- View Master Package structures.
- Run and monitor Execution Plans.

You can also set up connections to different environments, and use a single deployment of Oracle BI Applications Configuration Manager to manage them. The source files for Oracle BI Applications Configuration Manager are installed when you install Oracle Business Intelligence Applications on a machine. This topic explains how to use those files to set up and run Oracle BI Applications Configuration Manager, and contains the following topics:

- Overview to Setting Up Oracle BI Applications Configuration Manager
- How to Execute the Oracle BI Applications Configuration Manager Setup Script
- How to Set Up Oracle BI Applications Configuration Manager on Windows
- How to Set Up Oracle BI Applications Configuration Manager on UNIX/Linux
- About accessing the WebLogic Administration Console
- How to Launch Oracle BI Applications Configuration Manager
- How to Launch Oracle BI Applications Configuration Manager in Debug Mode
- How to create a Connection For Your Development Environment
- How to Login to Oracle BI Applications Configuration Manager Using A Connection

For information about using Oracle BI Applications Configuration Manager, see [Using Oracle BI Applications Configuration Manager](#)

Overview to Setting Up Oracle BI Applications Configuration Manager

Oracle BI Applications Configuration Manager uses a Web Logic application server with Oracle Application Development Framework (ADF) runtime that you must set up after you have installed Oracle Business Intelligence Applications on a machine. After you have run the Oracle Business Intelligence Applications installer on a machine, the Oracle BI Applications Configuration Manager files are installed in the Oracle_BI1\biapps7972\odi\biapps_configmgr directory. To install Oracle BI Applications Configuration Manager on a different machine (for example, a UNIX or Linux machine), you copy the appropriate files from the \Oracle_BI1\biapps7972\odi\biapps_configmgr directory to the target machine. On the target machine, you then set up Oracle BI Applications Configuration Manager as described in this topic. When deciding where to install WebLogic Server and deploy Oracle BI Applications Configuration Manager, note the following:

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

To set up Oracle BI Applications Configuration Manager:

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

How to Execute the Oracle BI Applications Configuration Manager Setup Script

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

To execute the Oracle BI Applications Configuration Manager setup script:

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

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

The default values are defined as follows:

```
/*  DEFINES for default schema names  */  
  
--ODI Repository default schema  
  
DEFINE L_ODI_SCHEMA    = ODI_REP_7972  
  
--OBIA Data warehouse default schema  
  
DEFINE L_OBIA_SCHEMA   = DATA_BIAPPS  
  
--Configuration Manager default schema  
  
DEFINE L_TX_SCHEMA     = DATA_BIAPPSTX
```

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

How to Set Up Oracle BI Applications Configuration Manager on Windows

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

To set up Oracle BI Applications Configuration Manager on Windows:

1. On the target machine, install JDK 1.6.0_05 or a higher 1.6.x version on the machine where Oracle WebLogic Server will be installed. Do not use spaces in the directory path name. If you are installing Oracle BI Applications Configuration Manager on the machine on which the Oracle BI Applications installer was run, skip Step 2 and go straight to Step 3.
2. If you are installing Oracle BI Applications Configuration Manager on a machine different from the one on which the Oracle BI Applications installer was run, copy the following files from

the\Oracle_BI1\biapps7972\odi directory on the Oracle Business Intelligence Applications installation machine to a directory on the target machine, as follows:

- a. On the Oracle Business Intelligence Applications installation machine, navigate to the \Oracle_BI1\biapps7972\odi directory.
- b. Copy the following files from the \Oracle_BI1\biapps7972\odi\ directory to an equivalent directory on the target machine:

biapps_configmgr.ear

jdevstudio11110install.jar

silent.xml

WIN_1_HowToInstallBIAppsConfigMgr.txt

WIN_2_RunJDevInstallerSilent.cmd

WIN_3_CreateDomain.txt

WIN_4_DeployBIAppsConfigMgr.cmd

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

- a. Edit silent.xml and specify the full path name to the WebLogic (BEA) Home directory by setting the value for the data-value name 'BEAHOME'. This is the location where Oracle WebLogic Server will be installed. For example, D:\Weblogic\.
- b. Edit WIN_2_RunJDevInstallerSilent.cmd, and set the JAVA_LOCATION variable to the path to the java.exe (JDK 1.6.0_05 or higher 1.6 version) on your machine, then save the changes to the file. For example, D:\jre1.6.0_02\bin.
- c. Execute the WIN_2_RunJDevInstallerSilent.cmd file. This script installs WebLogic Server + Application Developer Framework and ADF Runtime Libraries. WebLogic is installed in the directory specified by 'BEAHOME' in silent.xml.

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

- a. Launch the Oracle WebLogic Configuration Wizard by selecting Start >Programs > Oracle Fusion Middleware BEAHOME > Weblogic > Tools >Configuration Wizard.
- b. In the Welcome screen, select the Create a New WebLogic domain radio button, then click Next.
- c. In the Select Domain Source screen, select the Generate a domain configured automatically to support the following products: radio button.
- d. Select the Application Development Framework check box, then click Next.

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

- e. In the Configure Administrator Username and Password screen, enter User name and User Password, confirm the password, and then click Next.
- f. In the Configure Server Start Mode and JDK screen, retain the default selection of Development Mode for the WebLogic Domain Startup Mode.
- g. In the JDK Selection box, select the location of the JDK 1.6 installation on the machine, then click Next.
- h. In the Customize Environment and Services settings screen, select the No radio button, then click Next.
- i. In the Create WebLogic Domain screen, specify the Domain name.

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

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

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

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

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

- a. Edit the WIN_4_DeployBIAppsConfigMgr.cmd file.
- b. Set BEAHOME to the install location of Oracle WebLogic Server as specified in the silent.xml file in Step 3.
- c. Set USER_DOMAIN to the name of the WebLogic domain that you created in step 4.
- d. Set EAR_FILE_PATH to the location of the biapps_configmgr.ear file.
- e. Save changes to the file.
- f. Execute the WIN_4_DeployBIAppsConfigMgr.cmd file by double-clicking the file or by executing from a command prompt window.

This command file will deploy the Oracle BI Applications Configuration Manager application using the biapps_configmgr.ear file to the Oracle WebLogic Server installed in Step 3 and in the domain you have created in Step 4. The script starts the Oracle WebLogic Server after the BI Applications Configuration Manager is deployed. A notice similar to "<Notice> <WebLogicServer> <BEA-000360> <Server started in RUNNING mode>" is displayed in the command line window to indicate that the WebLogic Server is running and listening at the default port 7001.

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

For information on how to log in and use Oracle BI Applications Configuration Manager, see [How to Launch Oracle BI Applications Configuration Manager](#).

The URL for the WebLogic Console is:

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

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

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

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

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

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

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

2. Copy the following files from the <OBIEE11G Home>\biapps7972\odi\biapps_configmgr\ directory on the Oracle Business Intelligence Applications installation machine to a directory on the target machine, as follows:

- a. On the Oracle Business Intelligence Applications installation machine, navigate to the <OBIEE11G Home>\biapps7972\odi\biapps_configmgr\ directory.
- b. Copy the following files from the <OBIEE11G Home>\biapps7972\odi\biapps_configmgr\ directory to an equivalent directory on the target machine:

biapps_configmgr.ear

jdevstudio11110install.jar

silent.xml

UNIX_1_HowToInstallBIAppsConfigMgr.txt

UNIX_2_RunJDevInstallerSilent.sh

UNIX_3_CreateDomain.txt

UNIX_4_DeployBIAppsConfigMgr.sh

Note: The UNIX_2_RunJDevInstallerSilent.sh and UNIX_4_DeployBIAppsConfigMgr.sh scripts must have read, write, and execute permissions. If the files do not have these permissions, modify them using the chmod command. When you copy across these files, use a MS-DOS to UNIX conversion tool, convert the script files to UNIX format (that is, remove the carriage return and line feed characters). There are many MS-DOS to UNIX conversion tools that are freely available for download on the Internet. Alternatively, you can manually remove the carriage return and line feed character (/r) from the script files by running the following commands:

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

```
mv tmp.sh UNIX_2_RunJDevInstallerSilent.sh
```

Repeat for the UNIX_4_DeployBIAppsConfigMgr.sh file.

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

- a. Edit silent.xml and specify the full path name to the WebLogic (BEA) Home directory by setting the value for the data-value name 'BEAHOME'. This is the location where Oracle WebLogic Server will be installed.
- b. Edit UNIX_2_RunJDevInstallerSilent.sh, and set the JAVA_LOCATION variable to the path to the java binary (JDK 1.6.0_05 or higher 1.6 version) on your machine, then save the changes to the file. For example, /jre1.6.0_02/bin.
- c. Execute the command using:

./UNIX_2_RunJDevInstallerSilent.sh

Note: The script should be run under the bash shell. This script installs WebLogic Server + Application Developer Framework and ADF Runtime Libraries. WebLogic is installed in the directory specified by 'BEAHOME' in silent.xml.

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

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

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

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

- b. At the Welcome prompt, choose "1 Create a new WebLogic domain" by typing 1, then press Enter.
- c. At the Select Domain Source prompt, select "1 Choose Weblogic Platform components" by typing 1, then press Enter.
- d. At the Application Template Selection prompt, select "Application Development Framework" by typing 1, then press Enter.

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

- e. At the Application Template Selection prompt, type next and press Enter.
- f. At the Configure Administrator Username and Password prompt, create a user to be assigned to the Administrator role. This user is the default administrator used to start development mode servers.
- g. Follow the on screen instructions to set the user password and confirm user password. Optionally, change the default user name.
- h. At the Domain Mode Configuration prompt, select "1 Development Mode" by typing 1, then press Enter.
- i. At the Java SDK Selection prompt: specify Java SDK 1.6.0_05 or higher JDK 1.6 location.
- j. At the Choose Configuration Option prompt: do not change the default settings by typing 1, then press Enter.
- k. At the Select the target domain directory for this domain prompt, accept the default target location for the domain directory.

Note: Do not change the default value.

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

Note: Do not change the default value.

- m. At the Edit Domain Information prompt, select the default name 'base_domain'.
Alternatively, specify a different name if required.
- n. Complete the domain creation.

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

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

Edit the UNIX_4_DeployBIAppsConfigMgr.sh file.

- a. Set BEAHOME to the install location of Oracle WebLogic Server as specified in the silent.xml file in Step 3.
- b. Set USER_DOMAIN to the name of the WebLogic domain that you created in step 4.
- c. Set EAR_FILE_PATH to the location of the biapps_configmgr.ear file.
- d. Save changes to the file.
- e. Execute the UNIX_4_DeployBIAppsConfigMgr.sh file using:

`./UNIX_4_DeployBIAppsConfigMgr.sh`

Note: The script should be run under the bash shell. This command file will deploy the Oracle BI Applications Configuration Manager application using the `biapps_configmgr.ear` file to the Oracle WebLogic Server installed in Step 3 and in the domain you have created in Step 4.

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

For information on how to log in and use Oracle BI Applications Configuration Manager, see [How to Launch Oracle BI Applications Configuration Manager](#).

The URL for the WebLogic Console is: `http://<Weblogic Server Host>:7001/console`

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

About accessing the WebLogic Administration Console

The WebLogic Admin Console is accessed from the URL:

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

The default user name and password is `weblogic/weblogic`.

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

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

The default username and password is `data_biappstx/data_biappstx`.

How to Launch Oracle BI Applications Configuration Manager

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

For a list of supported browsers and requirements, see [Certification Matrix for Oracle Business Intelligence Applications for Oracle Business Intelligence Applications](#).

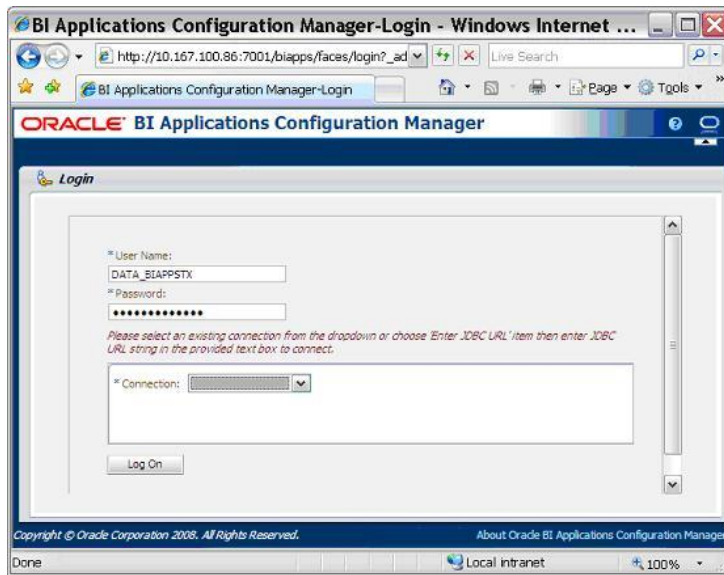
To start Oracle BI Applications Configuration Manager for the first time:

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

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

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

Figure 56



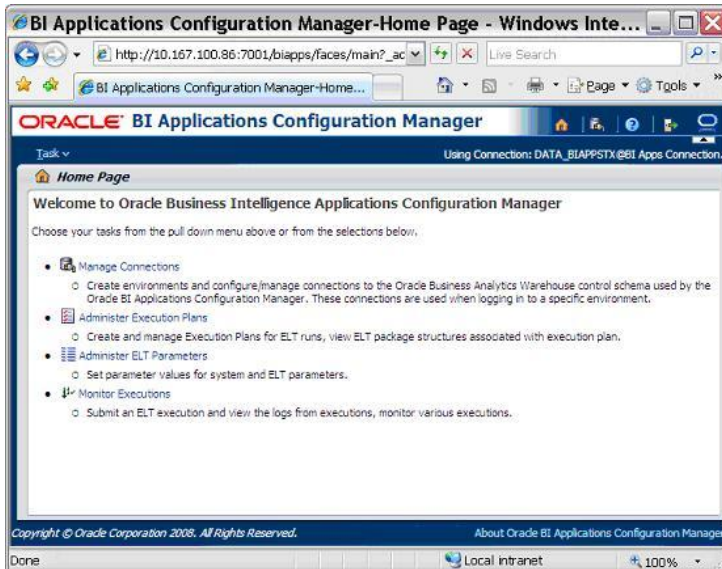
3. Enter appropriate login information, as described in the table below, then click Log On.

Table 21. Oracle BI Applications Configuration Manager Login page fields

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

The main Oracle BI Applications Configuration Manager page is displayed.

Figure 57



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

How to Launch Oracle BI Applications Configuration Manager in Debug Mode

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

To run Oracle BI Applications Configuration Manager in debug mode:

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

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

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

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

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

How to create a Connection For Your Development Environment

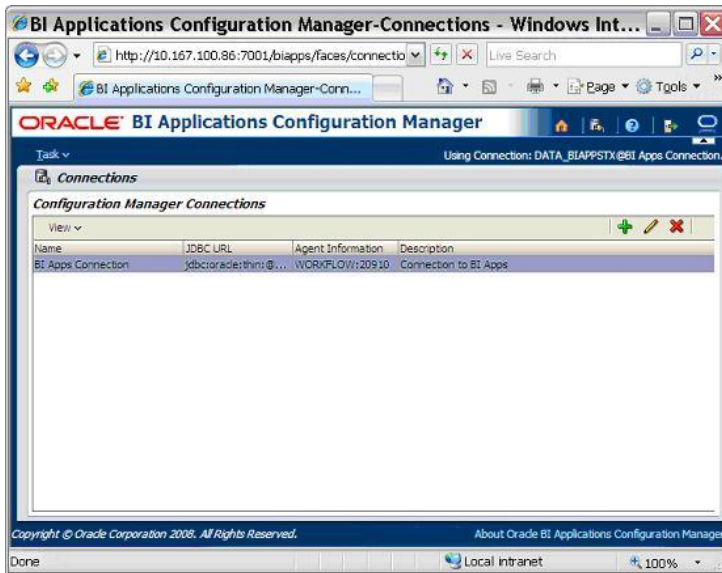
This topic explains how to create a connection for your development environment in Oracle BI Applications Configuration Manager. A connection is a stored set of environment details that includes a JDBC connect string to the Oracle Business Analytics Warehouse, an ODI agent name and port, and an ODI context. For example, you might create separate connections for Development, Test, and Production environments.

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

To create a connection in Oracle BI Applications Configuration Manager:

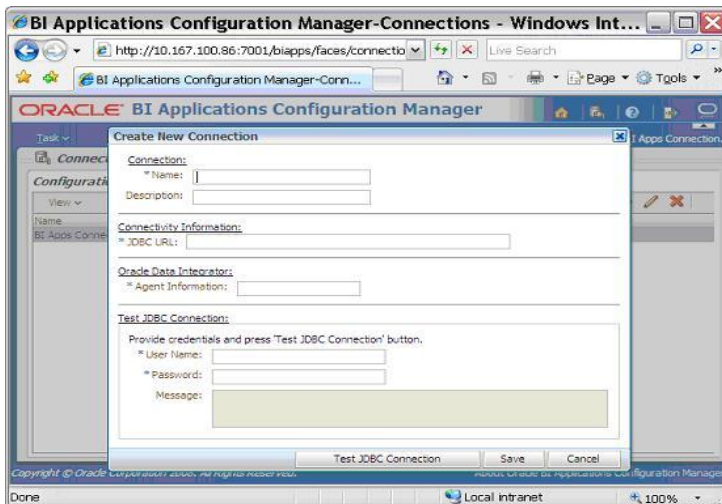
1. Start Oracle BI Applications Configuration Manager (for more information, see "How to Launch Oracle BI Applications Configuration Manager").
2. Select the Manage Connections link to display the Configuration Manager Connections page.

Figure 58



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

Figure 59



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

Table 22. Create New Execution Plan page fields

Field	Description
Name	A user-specified short descriptive name to identify the connection. For example, 'DEV Connection APAC'.

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

5. Click Test JDBC Connection to test the connection details. When a valid set of connection details are specified, a Successful! message is displayed.

6. Click Save.

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

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

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

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

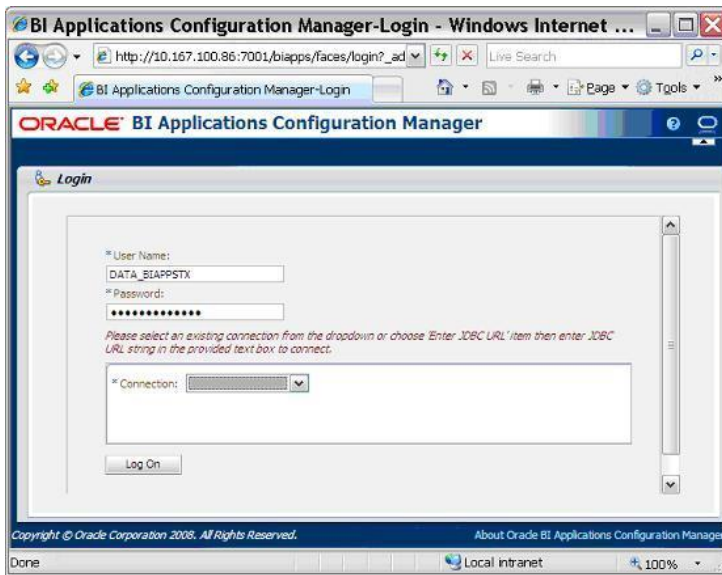
To start Oracle BI Applications Configuration Manager using a connection:

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

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

<http://mymachine.us.company.com:7001/biapps>

Figure 60



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

Table 23. *Oracle BI Applications Configuration Manager Login page fields*

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

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

Using Oracle BI Applications Configuration Manager

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

- Overview to Oracle BI Applications Configuration Manager
- Managing Connections
- Managing E-LT Parameters
- Managing Execution Plans
- Monitoring E-LT Processes

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

Figure 61.



Overview to Oracle BI Applications Configuration Manager

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

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

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

Managing Connections

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

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

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

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

Editing and Deleting Connections

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

Managing E-LT Parameters

This section explains how to manage E-LT Parameters in Oracle BI Applications Configuration Manager. The Parameters Administration page contains a tab for each category of parameters, as follows:

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

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

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

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

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

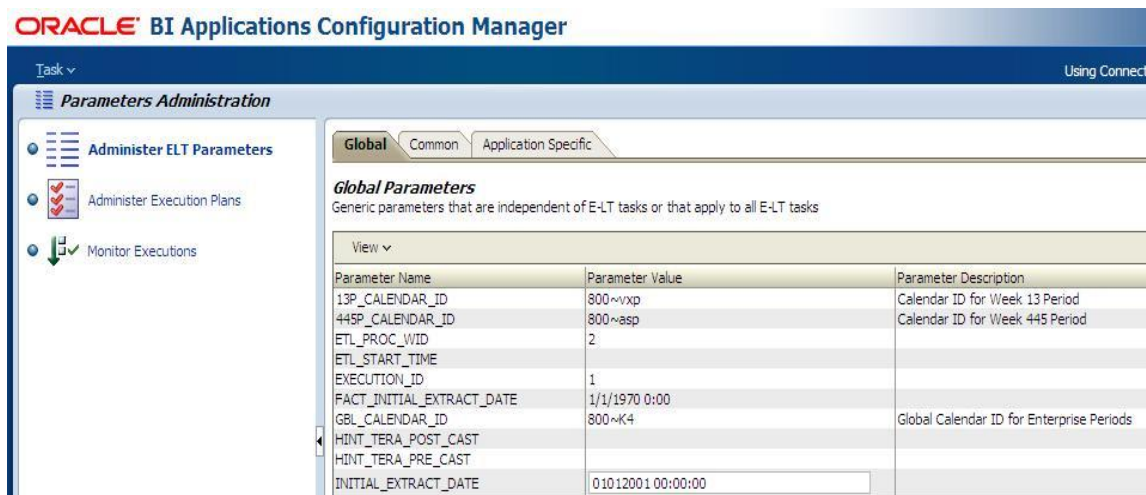
This topic provides generic instructions for modifying E-LT parameters. Part III Configuring Your Analytical Applications describes how to use Oracle BI Applications Configuration Manager to configure specific applications.

Note: Oracle BI Applications Configuration Manager does not perform validation on the parameter values that are entered. For parameters that must be entered in a specific format, follow the formatting instructions noted in the Description column for the parameter. For example, date parameter values must be entered in MMDDYYYY HH24:MI:SS format. Refer to the parameter Description information for the parameter to find out the format to use.

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

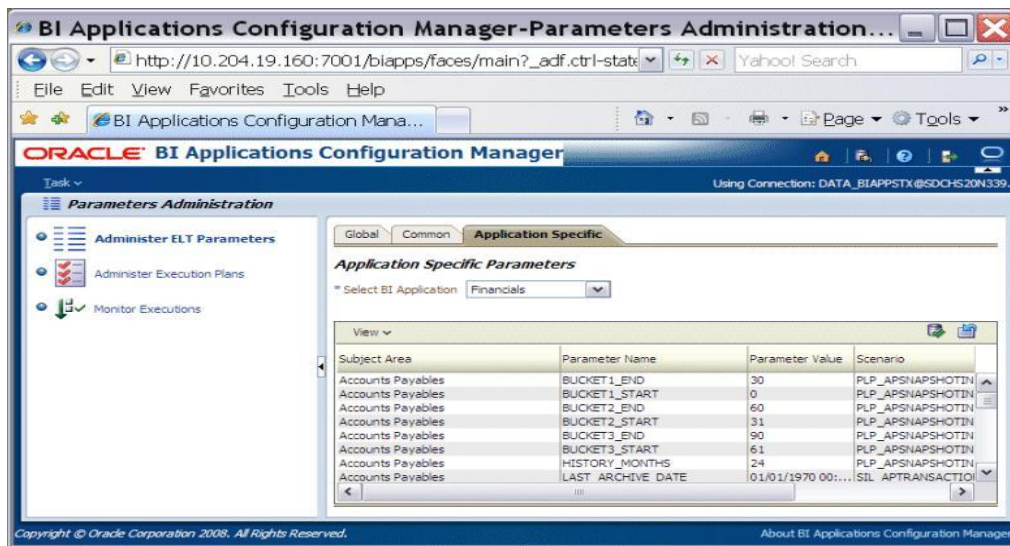
1. Start Oracle BI Applications Configuration Manager (for more information, see "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the Administer ELT Parameters link to display the Parameters Administration page.
3. Use the Global, Common, and Application Specific tabs to change the default values of the parameters.

Figure 62.



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

Figure 63.



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

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

This section explains how to reset database passwords, schema names, and connection details for the components in Oracle Business Intelligence Applications, and what related components you need to change to keep your components synchronized. For example, if you want to use a different database schema name to the recommended 'ODI_REP_7972', you also need to modify ORACLE_WORK_REP schema in ODI Topology Manager.

This section contains the following topics:

- How to Change the ODI Topology Connection Details

- How to Change the Encoded Passwords in odiparams.bat
- How to Change the Password for the ODI SUPERVISOR User
- Using Different Schema and Tablespace Names
- Changing the ODI Repository Schema Name

How to Change the Default Database Passwords

This topic explains how to change the default database passwords for the databases that you set up in the topic: Creating the Required Databases and Tablespaces. Use an appropriate SQL client to change the database password.

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

```
ALTER USER ODI_REP_7972 IDENTIFIED BY <New Password> REPLACE <Old Password>
```

Notes

- If you change the ODI_REP_7972 database password, you also need to modify the connection details for ORACLE_WORK_REP work repository as described in "How to Change the ODI Topology Connection Details".
- If you change the DATA_BIAPPS database password, you also need to modify the connection details for ORACLE_BI_APPLICATIONS as described in "How to Change the Encoded Passwords in odiparams.bat".

How to Change the ODI Topology Connection Details

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

Table 24. Database Schemas and Related Physical Data Servers

Database Schema Name	Related Physical Data Server	Action Required to Synchronize
ODI_REP_7972	ORACLE_WORK_REP	If you change the ODI_REP_7972 schema password, you need to modify the Oracle BI Applications Configuration Manager setup.
TEMP_BIAPPS	Not applicable.	Not applicable.
DATA_BIAPPS	ORACLE_BI_APPLICATIONS	If you change the DATA_BIAPPS schema password, you need to modify the Oracle BI Applications Configuration Manager setup.
DATA_BICONSOLE	Not applicable.	If you change the DATA_BICONSOLE schema password, you need to modify the Oracle BI Applications Configuration Manager setup.

To change the Work Repository connection details:

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

2. Expand the Work Repositories node.
3. Edit the repository WORKREP_7972, to display the Work Repository: <Repository Name> dialog.
4. Click Connection to display the Data Server: <Repository Name> dialog.
5. Use the Password field to modify the password, then click Test to verify the new password.
6. Save the details.

How to Change the Encoded Passwords in odiparams.bat

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

```
set ODI_SECU_USER=ODI_REP_7972
set ODI_SECU_ENCODED_PASS=dpfHHIqHwW.v9VGofVVd6eG0x
set ODI_USER=SUPERVISOR
set ODI_ENCODED_PASS=d,ypFC5Tzt5plXN82JwXASAUp
```

You might want to change the password for ODI_REP_7972 user or the SUPERVISOR user.

To encode a password:

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

2. Enter the following command:

```
agent ENCODE <password>
```

Where <password> is either ODI_REP_7972 or SUPERVISOR.

The 'agent ENCODE' command returns an encoded password for ODI_REP_7972 or SUPERVISOR and displays it on screen.

3. Note down or copy into memory the encoded password returned by the 'agent ENCODE' command.
4. Open the file \$ODI_HOME\oracledi\bin\odiparams.bat in a text editor.
5. If you obtained a new encoded password for the ODI_REP_7972 user, type in or paste in the encoded password that you obtained in step 3 as the value of the set 'ODI_SECU_ENCODED_PASS=' parameter.
6. If you obtained a new encoded password for the SUPERVISOR user, type in or paste in the encoded password that you obtained in step 3 as the value of the set 'set ODI_ENCODED_PASS=' parameter.
7. Save the odiparams.bat file.

How to Change the Password for the ODI SUPERVISOR User

This topic explains how to change the ODI SUPERVISOR password.

To change the passwords for the ODI SUPERVISOR user:

1. Log into ODI Designer or ODI Topology Manager.
2. Choose File, then Change Password to display the 'Password change' dialog.
3. Use the 'Password change' dialog to specify the current password, and the new password.
4. Save the details.

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

Using Different Schema and Tablespace Names

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

Changing the Oracle Business Analytics Warehouse Schema Names

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

1. Create new database account names for DATA_BIAPPS or TEMP_BIAPPS. For more information about creating database accounts, see "Creating the Required Databases and Tablespaces".
2. In ODI Topology, replace the database account names for DATA_BIAPPS or TEMP_BIAPPS to the new database account names that you created in step 1.
3. Modify the biappstx_setup.sql file as follows:
 - a. On the ODI machine, open the biappstx_setup.sql in a text editor or SQL tool. The biappstx_setup.sql file is located in the \$ODI_HOME\oracledi\biapps_odi\dbfiles\ directory. For example, D:\OraHome_1\oracledi\biapps_odi\dbfiles\.
 - b. Modify the value of the 'DEFINE L_OBIA_SCHEMA = 'statement'.
For example, you might change this statement to DEFINE L_OBIA_SCHEMA= DATA_TESTSCHEMA.
 - c. Save the file.

Changing the ODI Repository Schema Name

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

1. Create new database account names for ODI_REP_7972. For more information about creating database accounts, see "Creating the Required Databases and Tablespaces".
2. Modify the odiparams.bat or odiparams.sh file and change the following parameters:
ODI_SECU_USER
ODI_SECU_ENCODED_PASS

For information about obtaining encoded passwords, see "How to Change the Encoded Passwords in odiparams.bat".

3. Edit the ODI Master Repository to use the new database account name that you created in step 1.
4. In ODI Topology, replace the database account name for ODI_REP_7972 to the new database account name that you created in step 1.
5. Modify the biappstx_setup.sql file as follows:
 - a. On the ODI machine, open the biappstx_setup.sql in a text editor or SQL tool. The biappstx_setup.sql file is located in the \$ODI_HOME\oracledi\biapps_odi\dbfiles\ directory. For example, D:\OraHome_1\oracledi\biapps_odi\dbfiles\.
 - b. Modify the value of the 'DEFINE L_ODI_SCHEMA = 'statement'. For example, you might change this statement to DEFINE L_ODI_SCHEMA =ODI_REP_1.

- c. Save the file.

Changing the Schema Name for Oracle BI Applications Configuration Manager

To use a different Schema name, do the following:

1. Create a new database account name for DATA_BIAPPSTX. For example, you might create a new database account name DATA_CONFTX that has the same privileges as DATA_BIAPPSTX. For more information about creating database accounts, see "Creating the Required Databases and Tablespaces".
2. In Oracle BI Applications Configuration Manager, replace the database account name for DATA_BIAPPSTX to the new database account name that you created in step 1.
3. Modify the biappstx_setup.sql file as follows:
 - a. On the ODI machine, open the biappstx_setup.sql in a text editor or SQL tool. The biappstx_setup.sql file is located in the \$ODI_HOME\oracledi\biapps_odi\dbfiles\ directory. For example, D:\OraHome_1\oracledi\biapps_odi\dbfiles\.
 - b. Modify the value of the 'DEFINE L_TX_SCHEMA = 'statement. For example, you might change this statement to DEFINE L_TX_SCHEMA =DATA_CONFTX.
 - c. Save the file.

Changing the Schema Name for the Index Tablespace

To use a different index tablespace schema name, do the following:

1. Create a new database account name for BIAPPS_INDEX. For example, you might create a new database account name BIAPPS_INDEX_1 that has the same privileges as BIAPPS_INDEX. For more information about creating database accounts, see "Creating the Required Databases and Tablespaces".
2. In ODI Designer, modify the Oracle BI Applications for SAP model as follows:
 - a. Log into ODI Designer and display the Models view.
 - b. Double click on the Oracle BI Applications for SAP model to display the Model: <Name> dialog.
 - c. Display the FlexFields tab.
 - d. Replace the value of the INDEX_TABLESPACE field with the new database account name that you created in step 1.
3. Save the details, and re-start the ODI Agents if necessary.

Using A Different ODI Work Repository Name

Note: Before you modify the ODI Work Repository Name, make sure that all ODI processes have stopped. To use a different ODI Work Repository name, do the following:

1. Stop the ODI Agents INTERFACE and WORKFLOW.
2. In ODI Topology Manager, display the Repositories tab.
3. Expand the Work Repositories node and double-click on WORKREP_7972 to display the Work Repository: <Name> dialog.
4. Use the Name field to change the name of the ODI Work Repository, and save the details.

5. Log out of ODI Topology Manager.
6. Start ODI Designer to display the Oracle Data Integrator Login dialog.
7. Use the Login Name list to select the connection details for the environment that you are working in.
8. Click the Edit button to display the Work Repository Connection dialog.
9. Use the Repository Name field to change the name of the ODI Work Repository, and save the details.
10. In a text editor, edit the odiparams.bat or odiparams.sh file and modify the value of the 'set ODI_SECU_WORK_REP=' command.
11. For example, you might change the command to set ODI_SECU_WORK_REP=WORKREP_EMEA.
12. Re-start the ODI Agents WORKFLOW and INTERFACE.

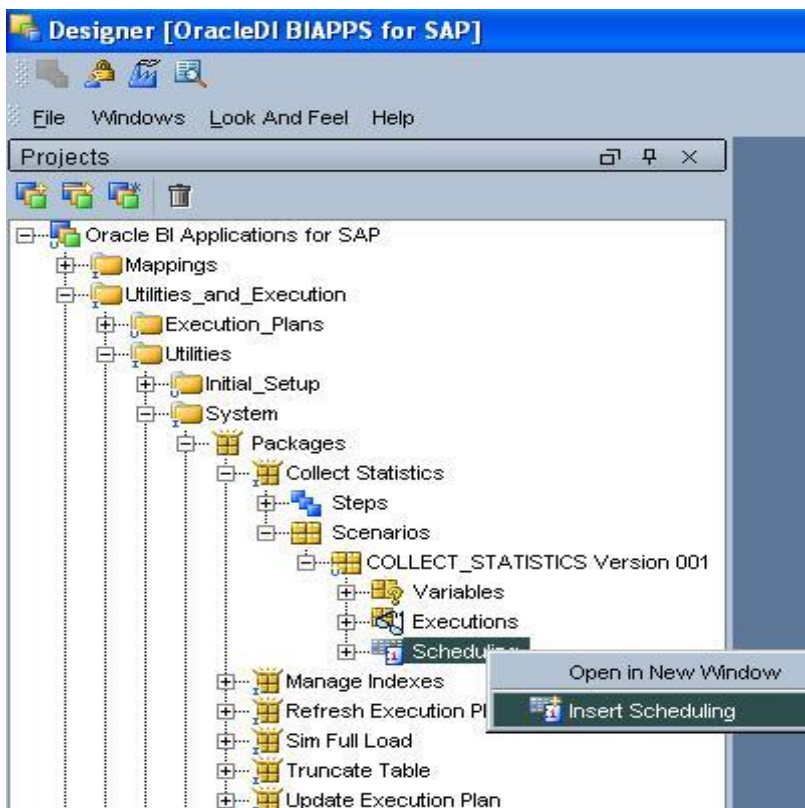
How to configure and activate Automated Database Statistics Collection

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

To activate and configure the 'Collect Statistics' Package:

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

Figure 64.



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

Figure 65.

Scenario Scheduling: PROD

Definition | Execution Cycle | Variables | Privileges | Version

Scenario: 0_MASTER_DELETE / 001

Context: 3. Production Agent: <Undefined>

Log Level: <Undefined>

Status:

☒ Active

☐ Inactive

☐ Active for the period:

Starting: the 21-Jan-2009 at 13:50:30

Ending: the 21-Jan-2009 at 13:50:30

Every day at interval: from 13:50:30 to 13:50:30

Except these days of the month:

Except these days of the week: Monday Tuesday Wednesday Thursday Friday Saturday Sunday

Execution:

☐ On startup

☒ Simple

☐ Hourly

☐ Daily

☐ Weekly

☐ Monthly (day of the month)

☐ Monthly (week day)

☐ Yearly

Simple

Run on 21-Jan-2009 at 13:50:30

OK Cancel Apply Help

5. Use the fields on the Scenario Scheduling: <Name> dialog to modify the schedule, as follows:
 - Use the Status area to activate or deactivate the package. For example, select the Active radio button to activate the package.
 - Use the Context field to select the 'Production' context.
 - Use the Agent field to select the 'INTERFACE' agent.
 - Use the other fields to specify the time and frequency of the refresh.
6. Save the details.

How to configure and activate Automated Delete Handling

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

To activate and configure the 'Collect Statistics' Package:

1. In ODI Designer, display the Projects view.
2. Open the project Oracle BI Applications 7.9.7.2
3. Navigate to
\\Utilities_and_Execution\\Execution_Plans\\Optional_Execution\\Delete_Master_Packages\\Packages\\0_Master_Delete\\Scenarios\\0_MASTER_DELETE Version 001\\Scheduling.
4. Right click on the Scheduling node and choose Insert Scheduling to display the Scenario Scheduling: <Name> dialog.

Figure 66.

5. Use the fields on the Scenario Scheduling: <Name> dialog to modify the schedule, as follows:
 - Use the Status area to activate or deactivate the package. For example, select the Active radio button to activate the package.
 - Use the Context field to select the 'Production' context.
 - Use the Agent field to select the 'INTERFACE' agent.
 - Use the other fields to specify the time and frequency of the refresh.
6. Save the details.

About Data Warehouse Loads

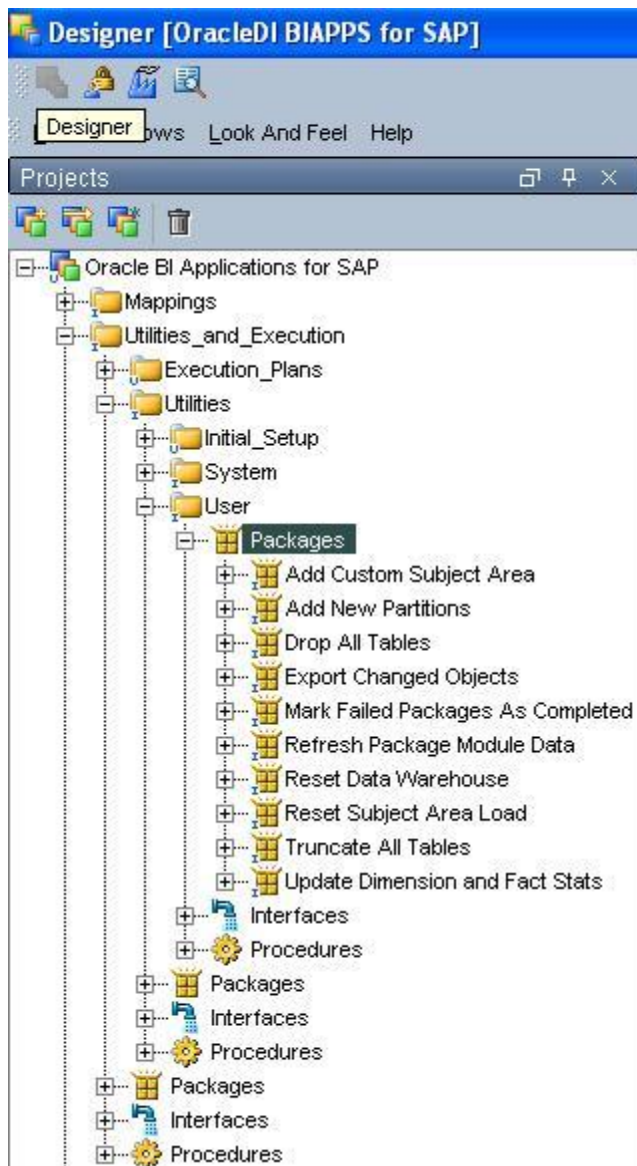
This topic explains how to use loads in ODI to manage your Oracle Business Intelligence Applications environment. The Master Packages are pre-programmed to handle incremental load logic implicitly. All the runs initiated after the initial full load is successfully done are incremental by default.

How to reset the Oracle Business Analytics Warehouse for Full Load

To reset the Oracle Business Analytics Warehouse for full load:

1. In ODI Designer, display the Projects view.
2. Open the project Oracle BI Applications 7.9.7.2.
3. Navigate to Utilities_and_Execution\Utilities\User\Packages\.

Figure 67



4. Execute the 'Reset Data Warehouse' package.

How to add partitions to Data Warehouse tables

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

- About supported partitioning in this release
- An example of partitioning a table
- How to generate a DDL for a modified table
- How to add a partition to a table that is already partitioned

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

About supported partitioning in this release

When you set up partitioning, note the following:

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

About using NUM_PARTITIONS_INCR FlexFields

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

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

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

An example of partitioning a table

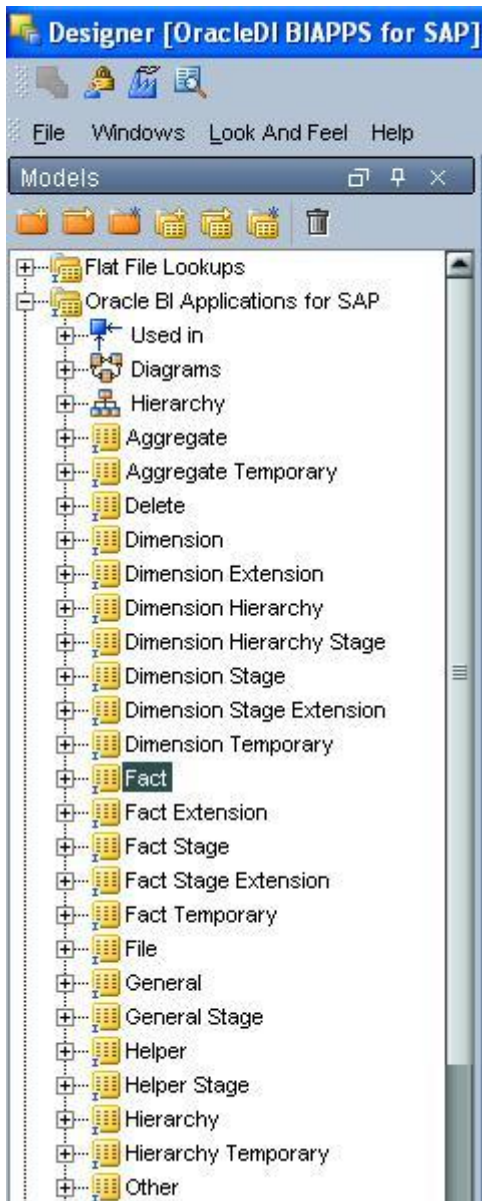
This topic uses an example to show how to partition a data warehouse table. Before starting this procedure, the following information is required:

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

Example of partitioning the W_REV_F fact table:

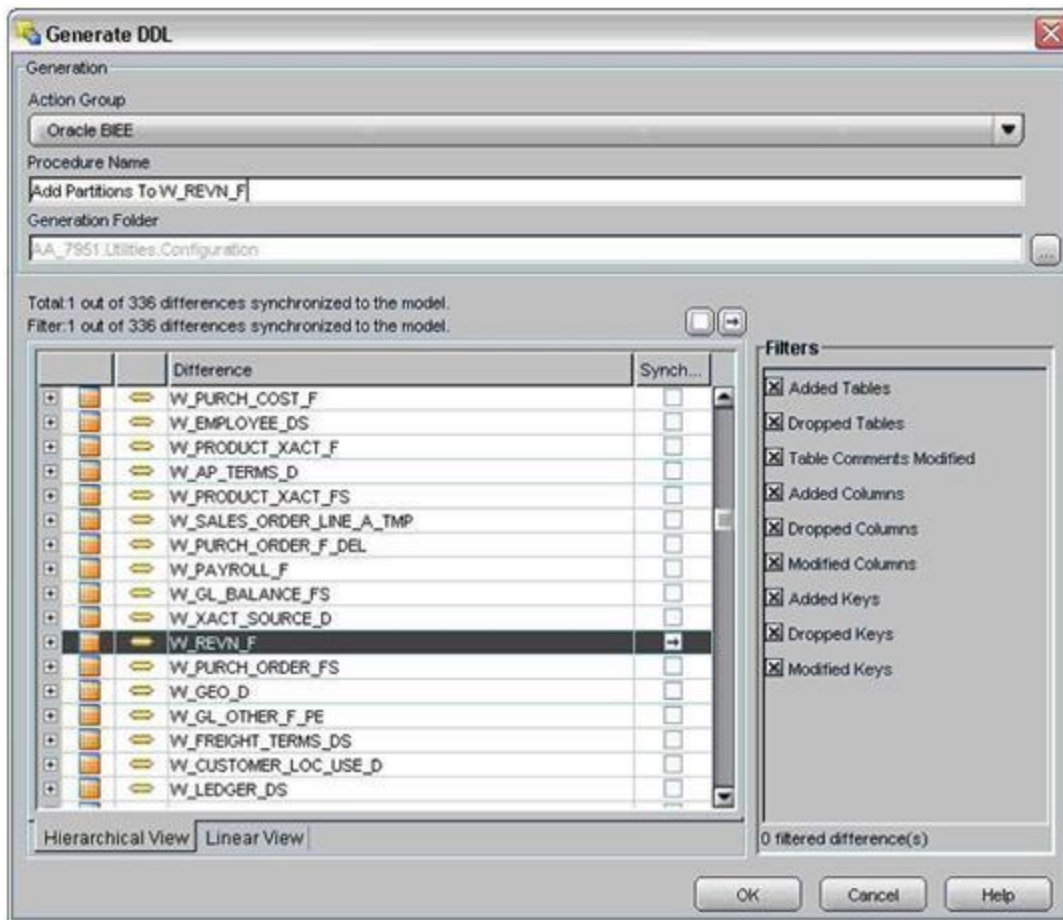
1. Use a SQL editor such as SQL*PLUS or SQL Developer to make a back-up of the fact W_REV_F fact table. This back-up version will enable you to repopulate the table in the later step in this topic: About Repopulating A Partitioned Table.
2. In ODI Designer, display the Models view.
Note: You must be logged in as the SUPERVISOR user.
3. Open the model Oracle BI Applications 7.9.7.2
4. Expand the Fact node.

Figure 68.



5. Locate the W_REV_F fact table.

Figure 69.



6. Double-click the W_REVN_F table to display the DataStore: <Name> dialog.

Note: If the Object Locking dialog is displayed, click Yes to unlock this object.

7. Display the FlexFields tab, and enter the appropriate information, as described in the table below.

Table 25. FlexFields Fields

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

8. Save the details.

9. Re-generate a DDL for this table by following the steps in this topic: How to generate a DDL for a modified table.

How to generate a DDL for a modified table

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

To generate a DDL for a modified table:

1. Using a SQL client tool (for example, Oracle SQL*Plus or Oracle SQL Developer), connect to your data warehouse as user DATA_BIAPPS.

2. To determine whether the W_REVN_F table is populated, enter the following SQL command:

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

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

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

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

```
DROP TABLE W_REVN_F;
```

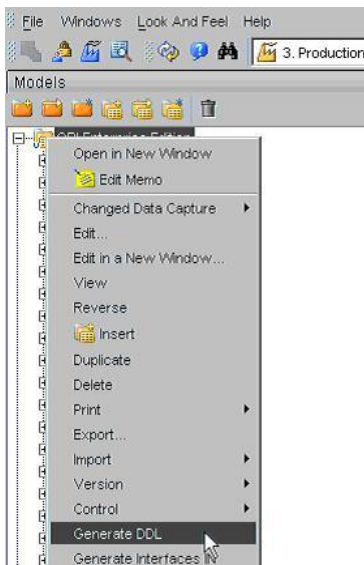
5. In ODI Designer, display the Models view.

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

6. Open the model ORACLE BI APPLICATIONS 7.9.7.2.

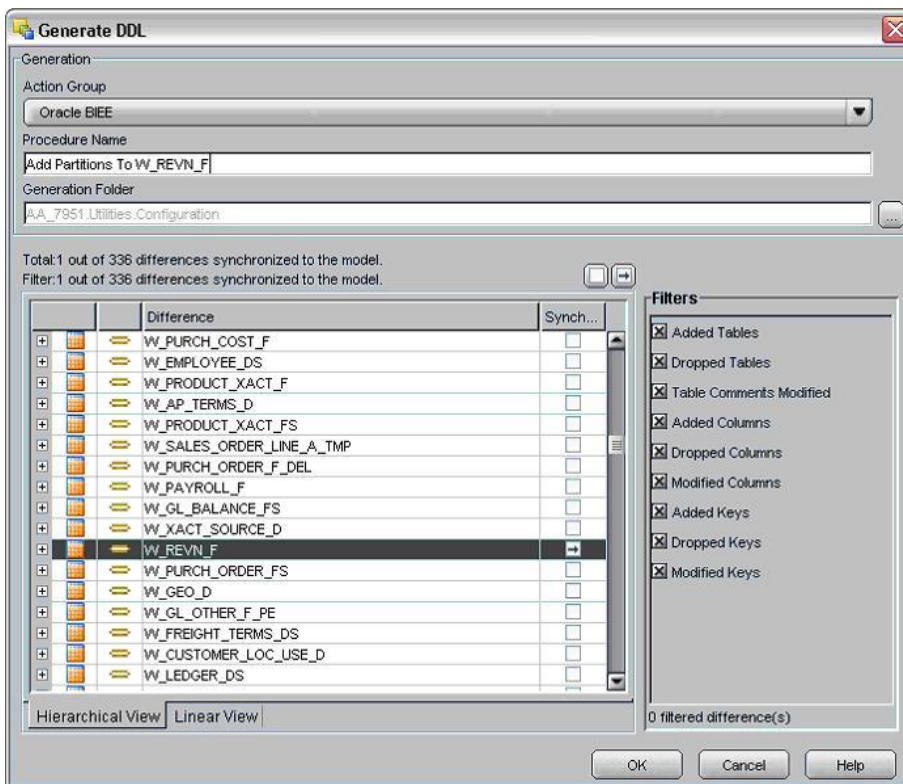
7. Right-click on Oracle BI Applications for SAP and select Generate DDL.

Figure 70.



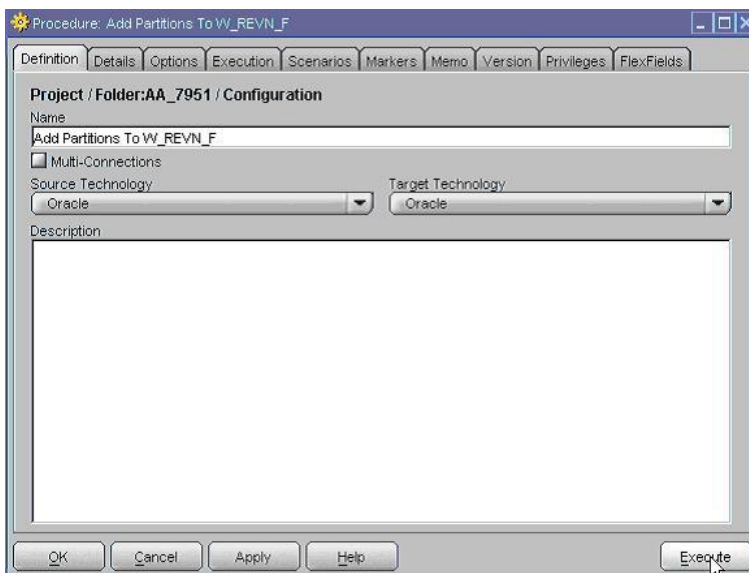
8. At the Generate DDL for Oracle Data Integrator Model dialog, click No to display the Generate DDL dialog.

Figure 71



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

Figure 72.



10. Select Oracle from the Source Technology drop down list.
11. Select Oracle from the Target Technology drop down list.
12. Click Execute to display a Confirmation dialog, and click Yes to display the Execution dialog.
13. Click OK to start the process.

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

About Repopulating A Partitioned Table

After you have partitioned a table (for an example using the fact table, see "An example of partitioning a table") you need to re-populate the table. Using a back-up copy of the table that you partitioned, repopulate the partitioned table. Use a SQL editor such as SQL*PLUS or SQL Developer to execute the 'INSERT INTO. . . SELECT FROM' SQL statement. Make sure that you execute a 'COMMIT' before closing the SQL session.

About Creating Indexes and Regenerating Statistics

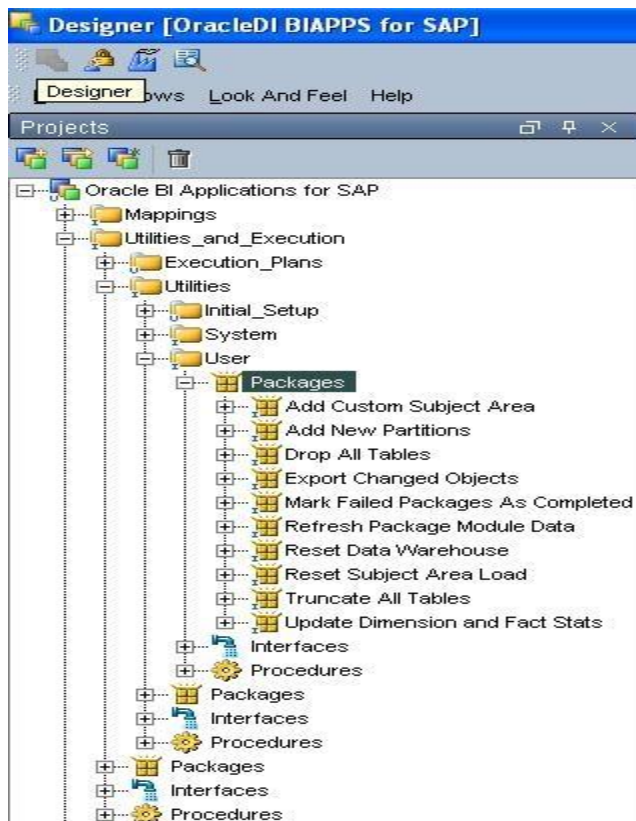
Oracle recommends that you manually create Partition indexes. Refer to Oracle Database documentation for details on how to create indexes on table partitions.

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

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

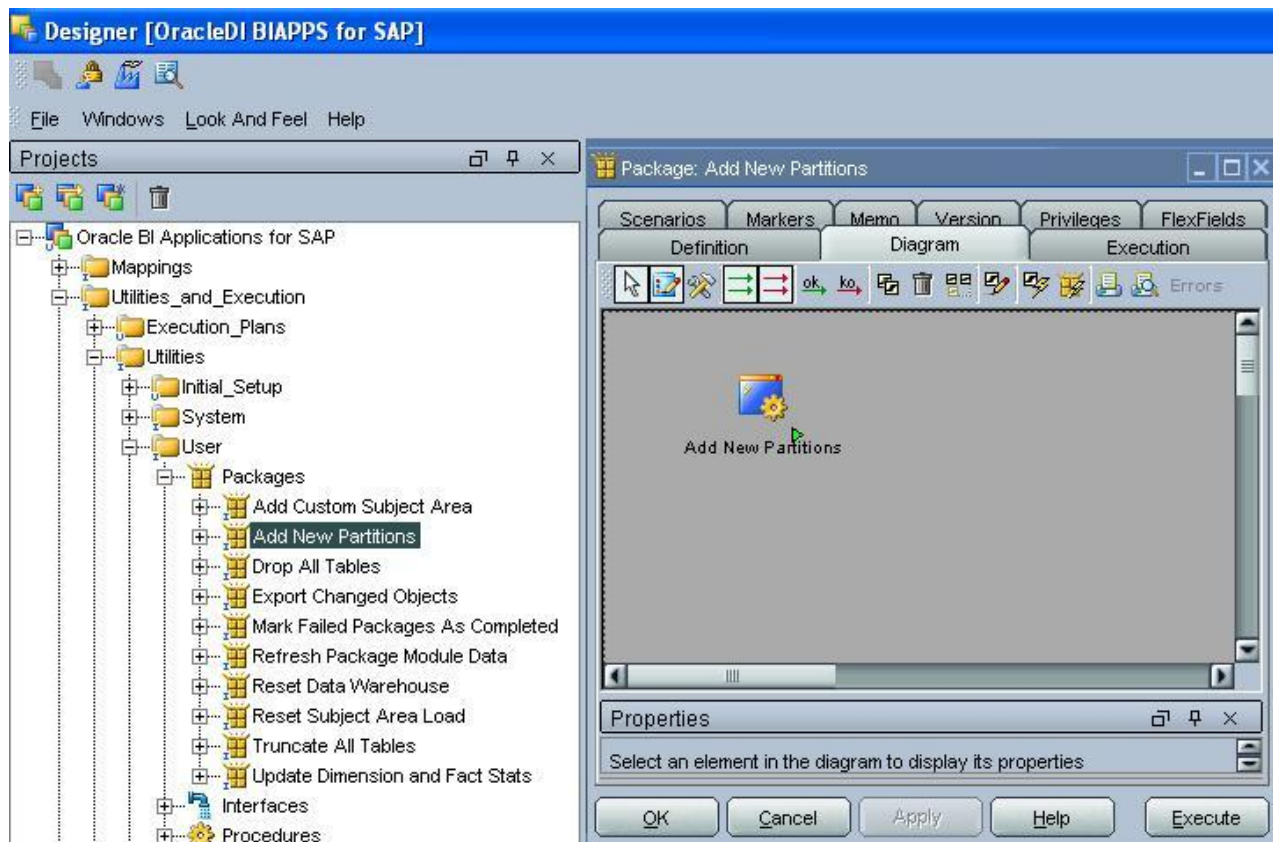
1. In ODI Designer, display the Projects view.
2. Open the project Oracle BI Applications 7.9.7.2.
3. Navigate to Utilities_and_Execution\Utilities\User\Packages\.

Figure 73.



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

Figure 74.



7. Specify the following values:

- TABLE NAME - W_REVN_F
- CURRENT_PARTITION_END_WID - 201012
- NEW_PARTITION_END_WID - 201112

8. Click Apply, then Execute to display the Execution dialog.

9. At the Execution dialog, select INTERFACE from the Agent drop down list, then OK.

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

About deploying ODI across multiple environments

In ODI deployments, you typically maintain different environments (known as Contexts) for Production (PROD), Development (DEV) and Quality Assurance (QA) activities. Oracle recommends that you install multiple ODI Repositories and create Topologies to support multiple environments. To avoid collision between different environments, it is also recommended that you install ODI separately on different servers for each of the environments.

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

- Create a Context for each of the additional environment i.e. DEV, QA, PROD.
- Create a database user for the ODI Staging schema.

- Create new data servers corresponding to the data source(s), Staging schema and Data Warehouse target.
- For each data server, create corresponding physical schemas.
- Create logical schemas and associate them with physical schemas in the contexts.
- Associate a DATASOURCE_NUM_ID with the new contexts.
- Create the physical agents for each agent running on a machine (as a listener, or in scheduler mode).
- Create logical agents and associate them with physical agents in the contexts.
- Create Physical and Logical servers corresponding to each of the data sources and data warehouse targets.

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

How to run an ODI Agent as a Unix background process

To run an ODI Agent as a Unix background process:

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

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

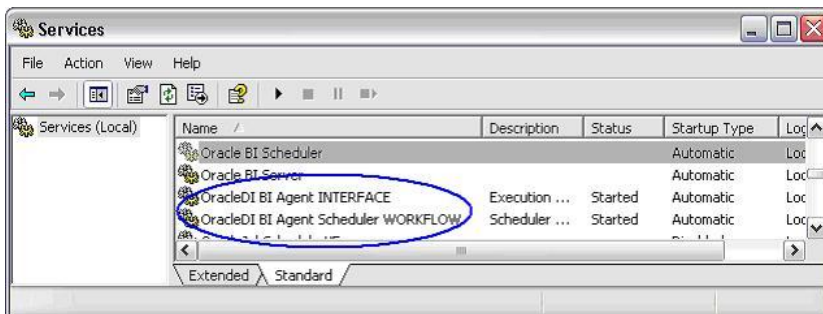
How to uninstall ODI Agent Windows Services

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

To uninstall the ODI Agent services:

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

Figure 75.



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

```
agentservice -r -s WORKFLOW 20910
```

```
agentservice -r -a INTERFACE 20911
```


Figure 76

```
Agentservice.bat
(c) Copyright Oracle. All rights reserved.
Wrapper ! OracleBI BI Agent Scheduler WORKFLOW installed.
Starting Agent Scheduler Service...
SERVICE_NAME: SnpAgentScheduler-WORKFLOW
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE                : 2   START_PENDING
                                (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE       : 0   (0x0)
        SERVICE_EXIT_CODE   : 0   (0x0)
        CHECKPOINT           : 0x0
        WAIT_HINI            : 0x7d0
        PID                  : 3464
        FLAGS                 :
```

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

How to Resolve Conflicts in ODI Agent Port Numbers

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

To resolve conflicts in ODI Agent port numbers:

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

How to activate or deactivate Flow Control in ODI

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

To activate or deactivate Flow Control:

1. In ODI Designer, display the Projects\Mappings tab and locate the Interface for which you want to modify the Flow Control behavior.
2. Edit the Interface to display the Interface: <Name> dialog, and display the Flow tab.
3. Use the IKM Selection field to select an appropriate knowledge module, and set the value of the FLOW_CONTROL option and the ANALYZE_FLOW_CONTROL option. When Flow Control is turned on, ODI starts to filter out error records from the data flow. These error records are saved into error tables internally. You can enable recycling (processing) of error records by setting the RECYCLE_ERRORS property to YES.

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

List of Log Files

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

Table 26. Oracle BI Configuration Manager Log Files

Log File Name	Log File Description
<root session number>_<date string>_<time string>_debug.log	<root_session_number> corresponds to the root of the session hierarchy as visible in ODI Operator. This log file must be submitted to Oracle Support for resolution of Oracle BI Configuration Manager Execution Plan related issues. Debug information from Oracle BI Applications Configuration Manager, which is stored in: <Stand-alone WebLogic installationfolder>/user_projects/domains/base_domain. Note: For more information, see "How to Launch Oracle BI Applications Configuration Manager in Debug Mode".
calcStatsFile_name.log	Database statistics collection log written by the Database Statistics collection utility
dropTableFile_name.log	Drop table log written by the Utility Package 'Drop All Tables' and the Procedure 'Drop Table'.
indexFile_name.log	Index Management log written by Knowledge Modules during Data Warehouse loads.
truncateFile_name.log	Truncate table log written by Knowledge Modules during Data Warehouse loads.
ZODI_<interface_id>_<sourceset_id>_<context>.log ZODI_<interface_id>_<sourceset_id>_<context>.ctl ZODI_<interface_id>_<sourceset_id>_<context>.out ZODI_<interface_id>_<sourceset_id>_<context>.bad	Corresponds to the log file of LKM when an ODI interface executes

Other log files:**Table 27. Other Log Files**

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

Loading Source Data Using an Execution Plan

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

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

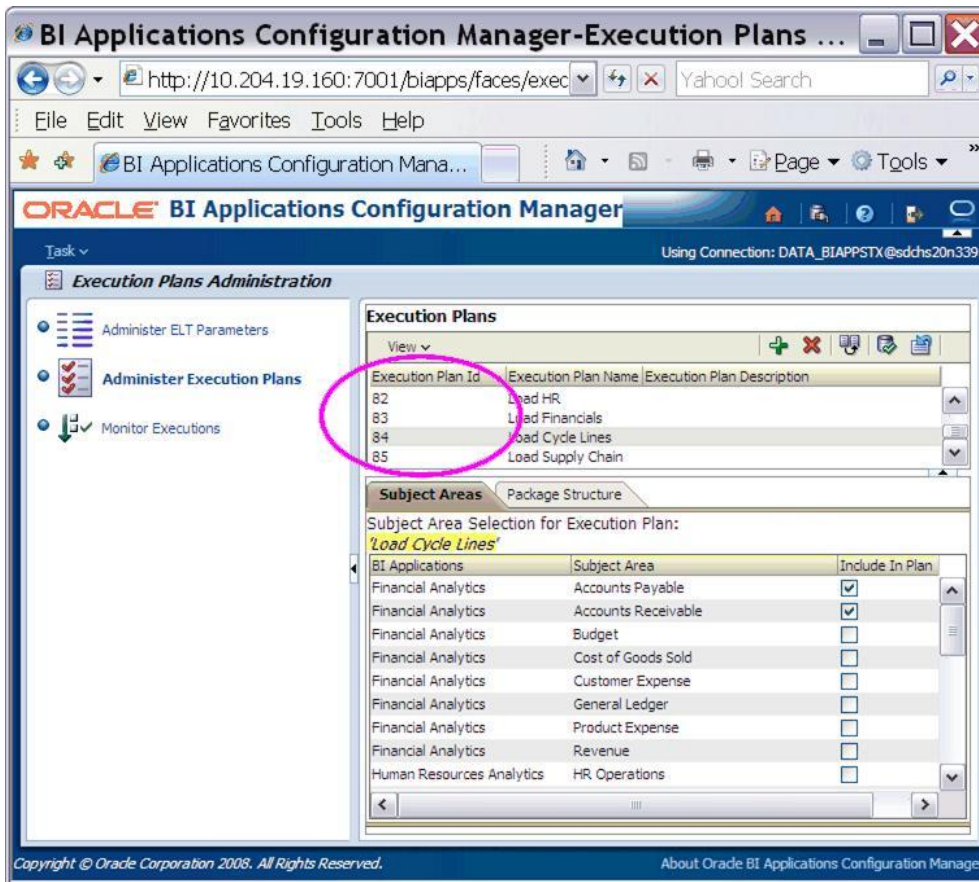
Note: Before you load your OLTP data, you typically configure your applications (for more information, see "Configuring Your Analytical Applications") and make customizations if required (for more information, see "Customizing Oracle Business Intelligence Applications").

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

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

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

Figure 77.



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

- Use Oracle BI Applications Configuration Manager to run the Execution Plan. For more information, see "How to Run an Execution Plan in Oracle BI Applications Configuration Manager".
- Use ODI Designer to run the Execution Plan either once, or scheduled to run more than once.

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

- Create a copy of the ODI master package supplied with Oracle Business Intelligence Applications, and set the value of the OBI_EXECUTION_PLAN setting to match the unique Execution Plan Id value assigned in Oracle BI Applications Configuration Manager. For more information, see "How to set up Master Packages to run an Execution Plan".
- Execute the Scenario for the copy of the master package that you created. For more information about using ODI Designer to run an Execution Plan, see "How to Run an Execution Plan in Oracle BI Applications Configuration Manager".

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

How to set up Master Packages to run an Execution Plan

This topic explains how to set up Master Packages in ODI Designer to run an Execution Plan. You set up Master Packages by creating a copy of the ODI master package supplied with Oracle Business Intelligence Applications, and setting the value of the OBI_EXECUTION_PLAN setting to match the unique Execution

Plan Id value assigned in Oracle BI Applications Configuration Manager when you create an Execution Plan.

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

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

Applications Configuration Manager".

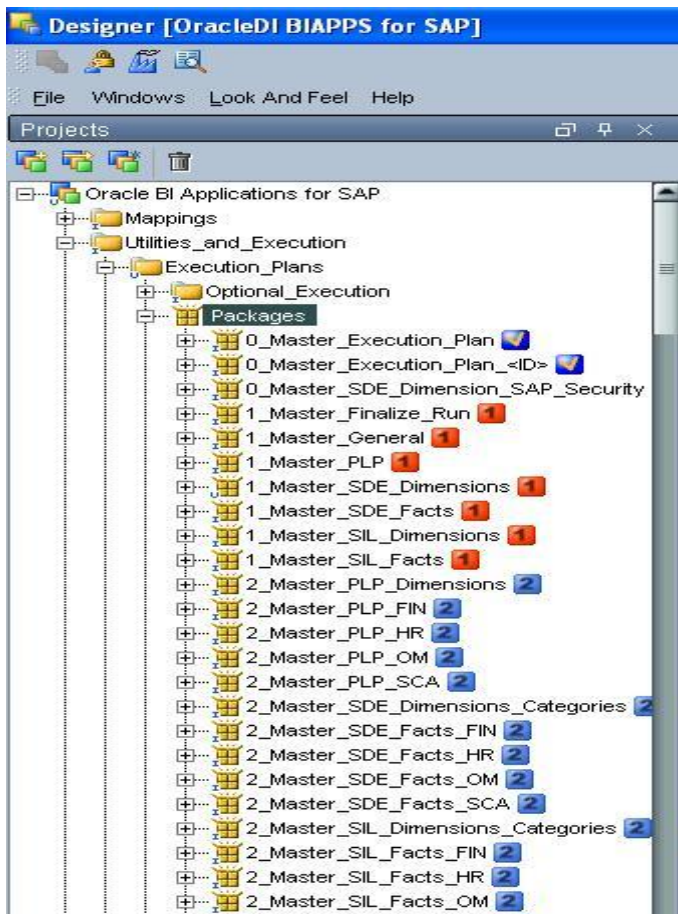
To set up Master Packages to run Execution Plans:

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

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

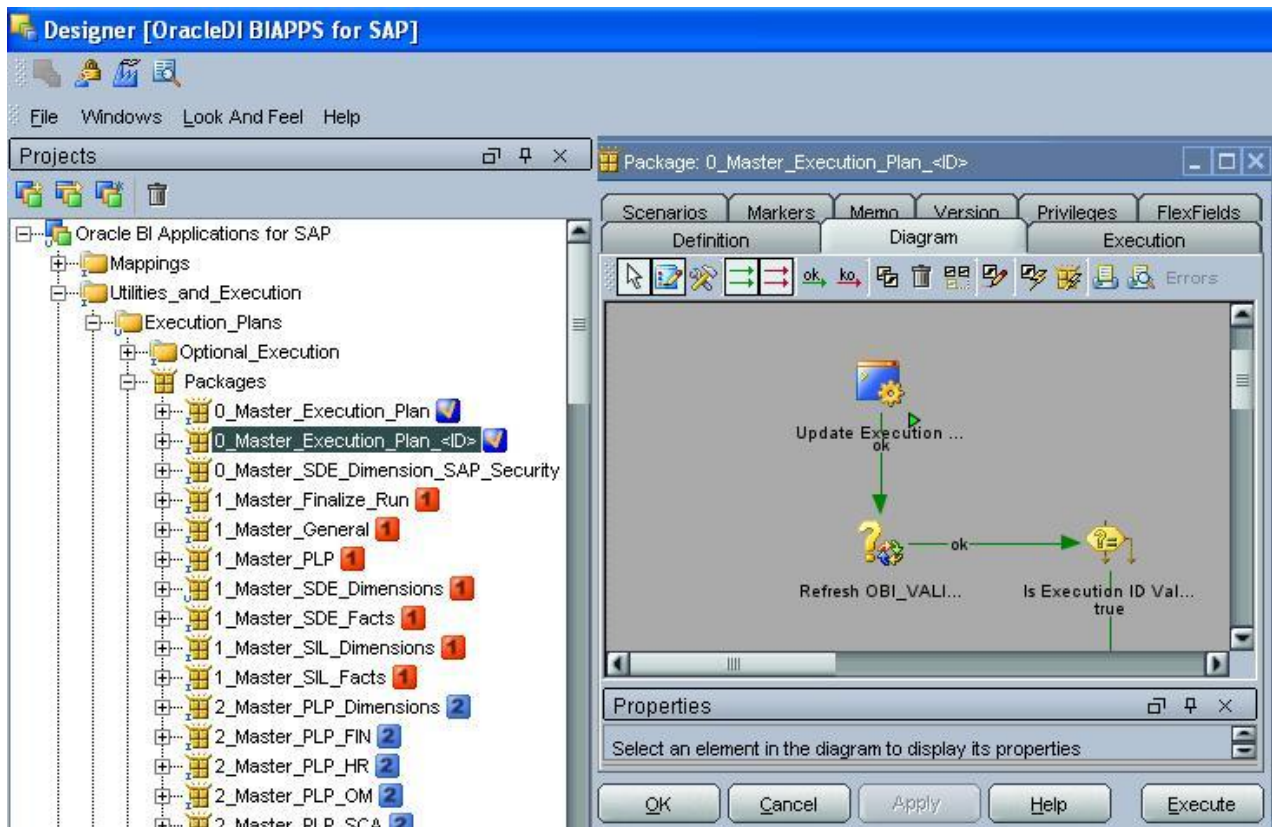
2. Expand the Oracle BI Applications for SAPproject.
3. Select Utilities_and_Execution, then Execution Plans, then Packages.

Figure 78.



4. Right-click on 0_Master_Execution_Plan_<ID> and choose Duplicate. ODI Designer creates a copy of the package '0_Master_Execution_Plan_<ID>' named 'Copy of 0_Master_Execution_Plan_<ID>'. You can optionally rename this copy with a suitable name, for example, 'Execution Plan for HR'.
5. Double-click on the copy of the package to display the Package: <Name> dialog for this package.
6. Display the Diagram tab.
7. Select the 'Update Execution Plan' procedure at the top of the Diagram pane, then display the Options tab on the Properties pane below.

Figure 79.



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

Loading Source Data Using an Execution Plan

8. On the Options tab, use the Value field to set the value of the OBI_EXECUTION_PLAN property. You set the value of the OBI_EXECUTION_PLAN field to the same as the unique Execution Plan Id value assigned to the Execution Plan in Oracle BI Applications Configuration Manager.

For example, if an Execution Plan that you create in Oracle BI Configuration manager is assigned the ID '470', to run that Execution Plan you set the value of the OBI_EXECUTION_PLAN field to '470'. For more information about Execution Plan IDs, see "How to Create an Execution Plan In Oracle BI Applications Configuration Manager").

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

Notes

- To run multiple execution plans, copy the O_Master_All_Exec_Plan_1 package, edit the package details, and set the OBI_EXECUTION_PLAN setting to the same as the unique Execution Plan Id value assigned to the Execution Plan in Oracle BI Applications Configuration Manager.

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

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

1. Configure E-LT Parameters to enable you to control E-LT processes (for more information, see "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager").
2. Create an Execution Plan (for more information, see "How to Create an Execution Plan In Oracle BI Applications Configuration Manager").
3. Run the Execution Plan (for more information, see "How to Run an Execution Plan in Oracle BI Applications Configuration Manager").

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

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

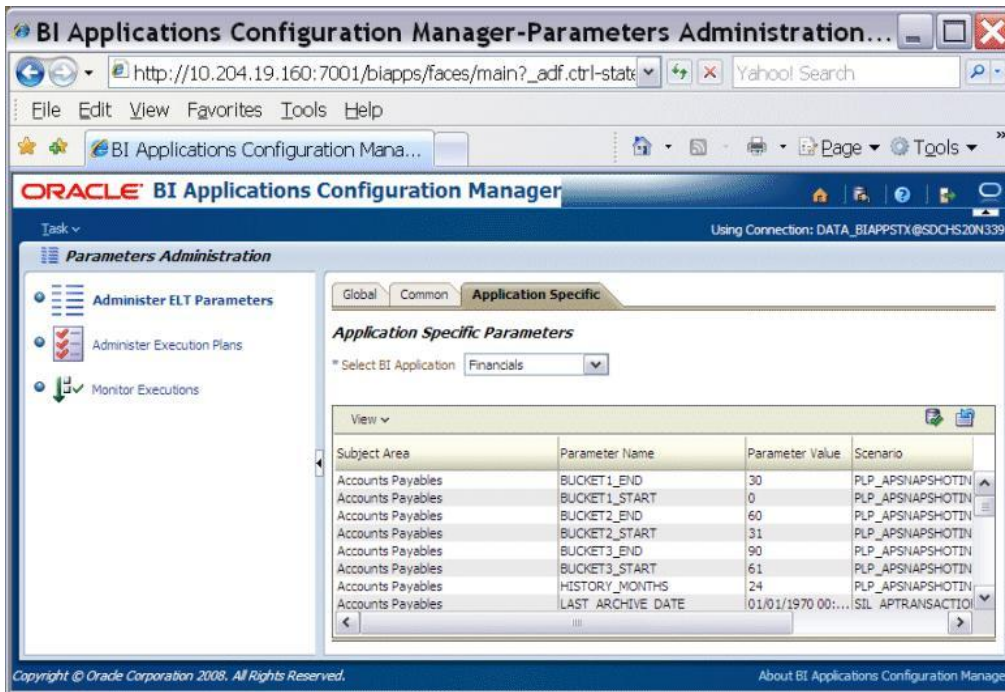
1. Start Oracle BI Applications Configuration Manager (for more information, see "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the Administer ELT Parameters link to display the Parameters Administration page.

Figure 80



3. Use the Global, Common, and Application Specific tabs to change the default values of the parameters. At the Application Specific tab, the parameters are grouped into Subject Area within Application, to help you locate the parameters that are specific to your Application.

Figure 81

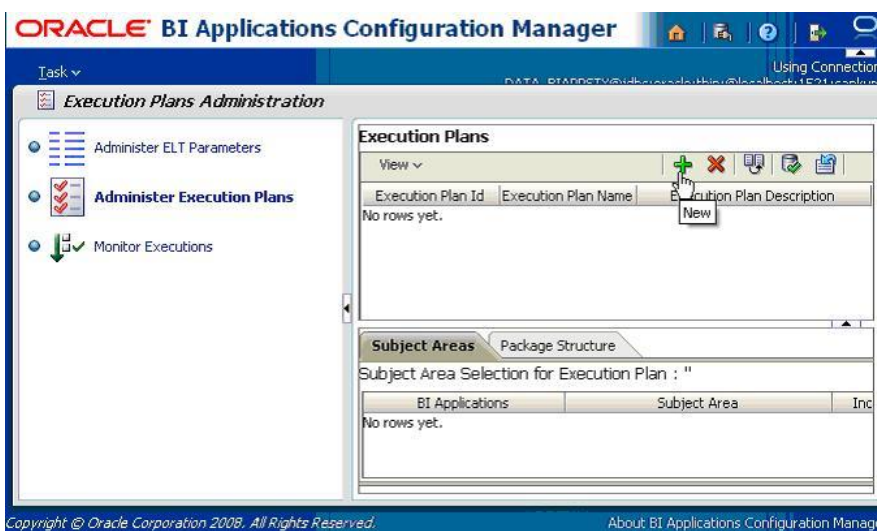


How to Create an Execution Plan In Oracle BI Applications Configuration Manager

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

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

Figure 82



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

Figure 83

Create New Execution Plan

Execution Plan Information:

Name :

Description :

Select Application or Subject Area to be included:

- ☐ Custom Applications
 - Custom Subject Area
- ☐ Financial Analytics
 - Accounts Payable
 - Accounts Receivable
 - Budget
 - Cost of Goods Sold
 - Customer Expense
 - General Ledger
 - Product Expense
 - Revenue
- ☐ Human Resources Analytics
 - HR Operations
 - Payroll
 - Workforce Events
- ☐ Order Management Analytics
 - Backlogs
 - Booking Line
 - Customer Status History
 - Cycle Lines

Synchronize package structure ☐

OK Cancel

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

Table 28. Create New Connection page fields

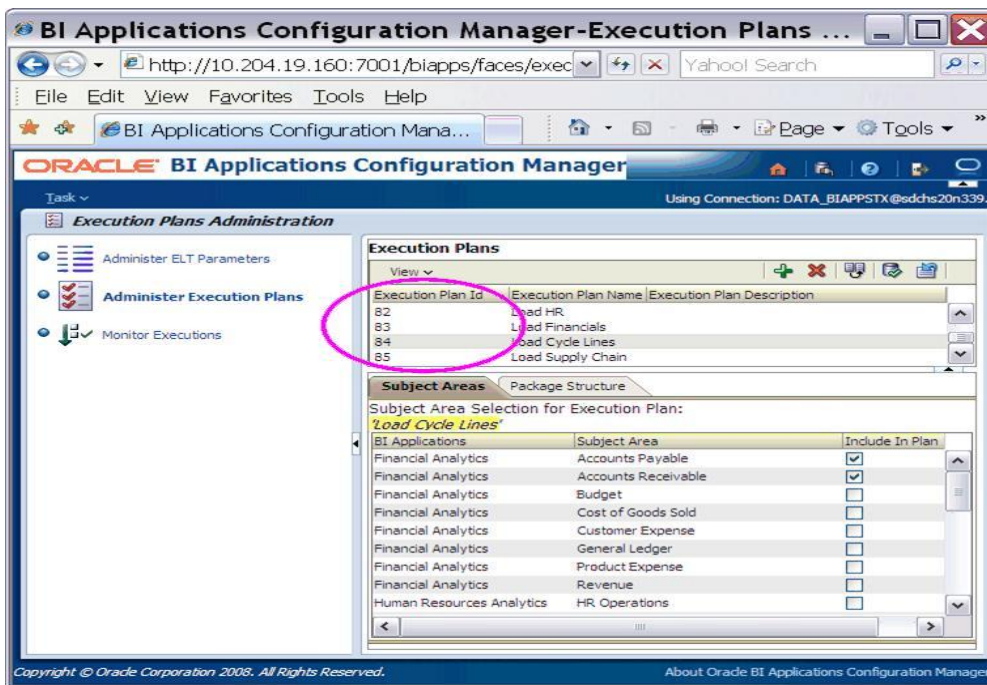
Field	Description
Description	Enter a short name to description of the Execution Plan. This description will be displayed in Oracle BI Applications Configuration Manager and ODI Designer.
Select Application or Subject Area to be included	Select the analytic application from which you want to load data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

	that appears on the Package Structure tab in order to get correct package structure. The synchronization process may take up to several minutes depending on the network traffic between the Oracle Business Analytics Warehouse database and the Oracle BI Applications Configuration Manager installation.

5. Save the details.

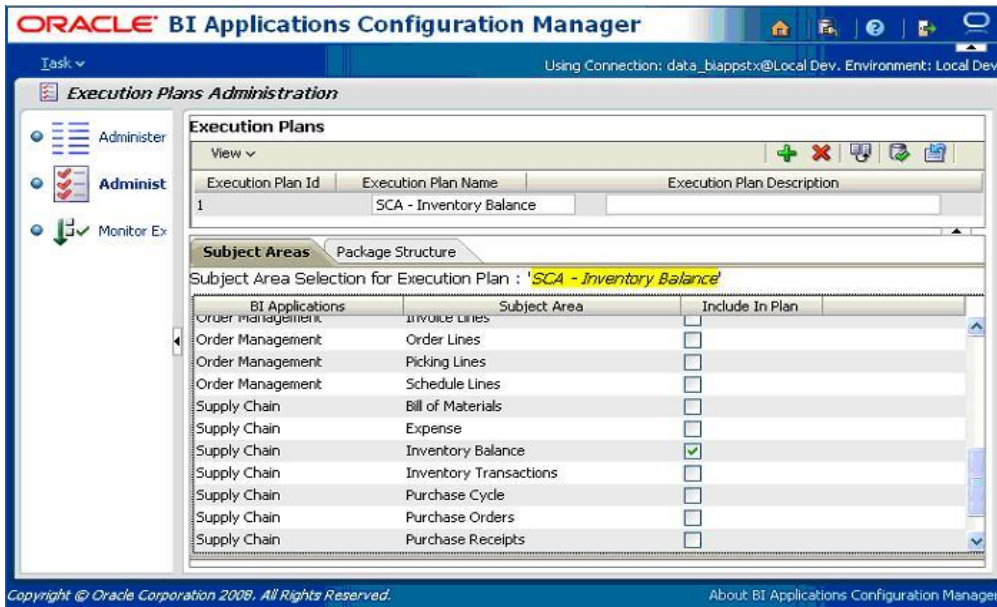
Note: The OK button is only active if you have included at least one Subject Area in the Execution Plan. When you save the details, Oracle BI Applications Configuration Manager automatically assigns a unique Execution Plan ID to the Execution Plan. You can see the Execution Plan ID value displayed in the Execution Plan Id column on the Execution Plan list (see screen shot below).

Figure 84



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

Figure 85

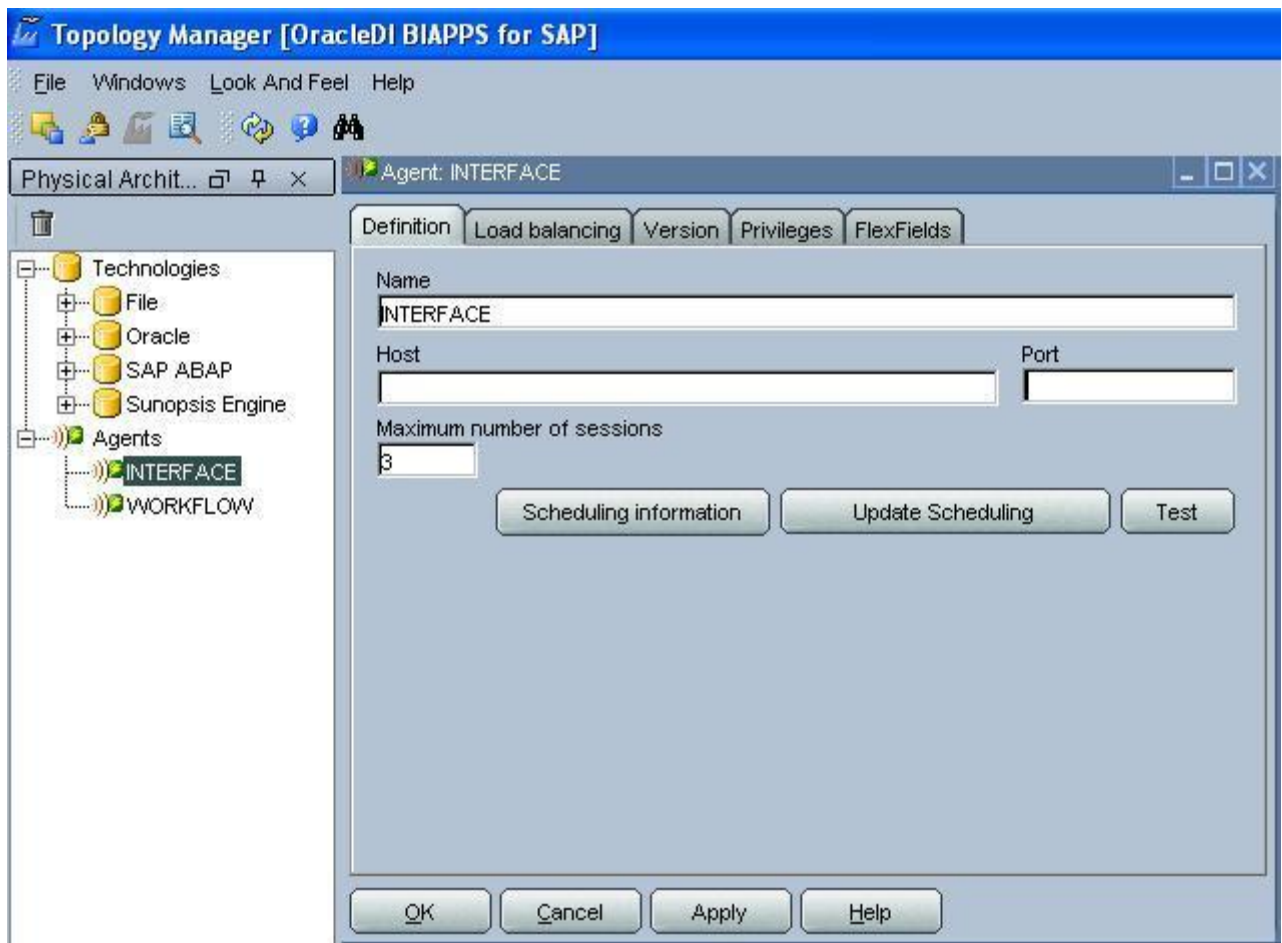


How to Upload SAP ABAP programs (PART 2 of 2)

This section describes how to upload the ABAP Programs generated by the interfaces. The upload of abap programs has to be done sequentially but execution can be parallel.

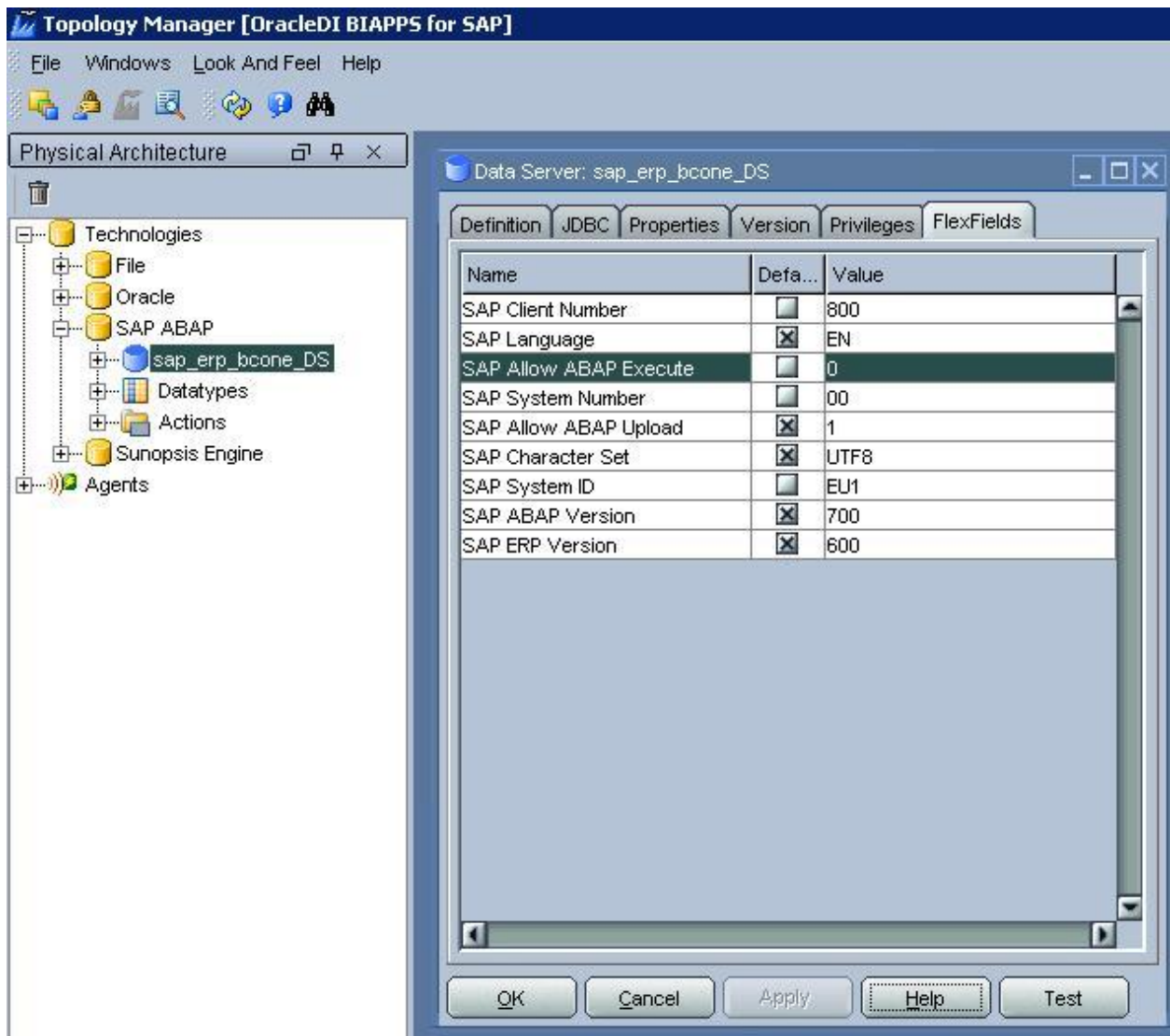
1. Go to Topology-> Physical Architecture. Expand agents and change the Maximum number of sessions on INTERFACE agent to 1 and click apply.

Figure 86



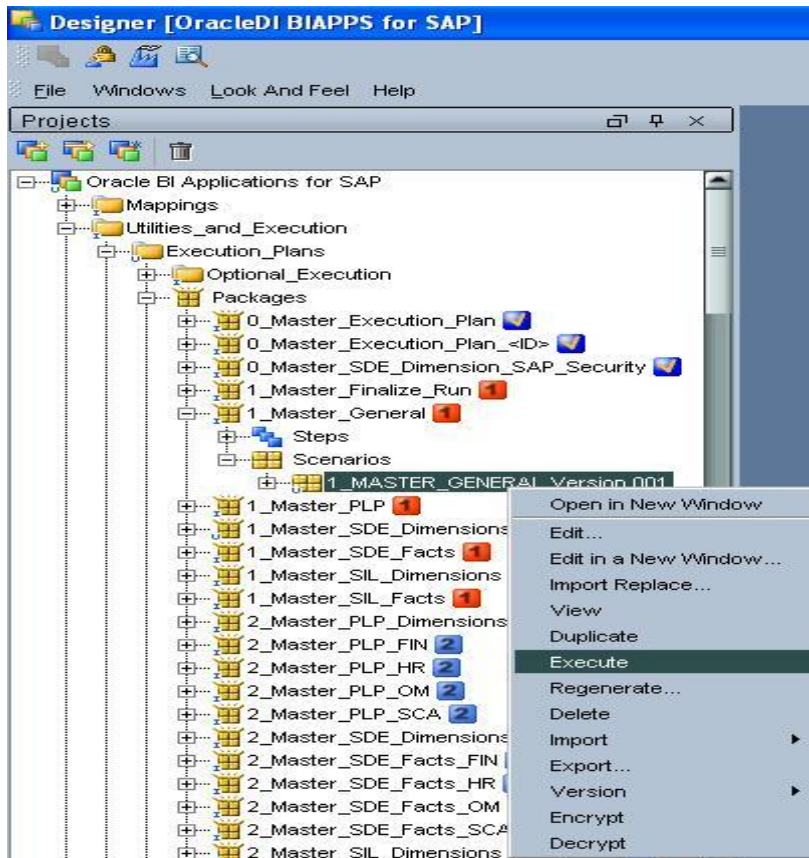
2. Go to Topology-> Physical Architecture .Expand Technologies and double click on the SAP data server . Go to flexfields and change the value of SAP Allow ABAP execute to 0.

Figure 87.



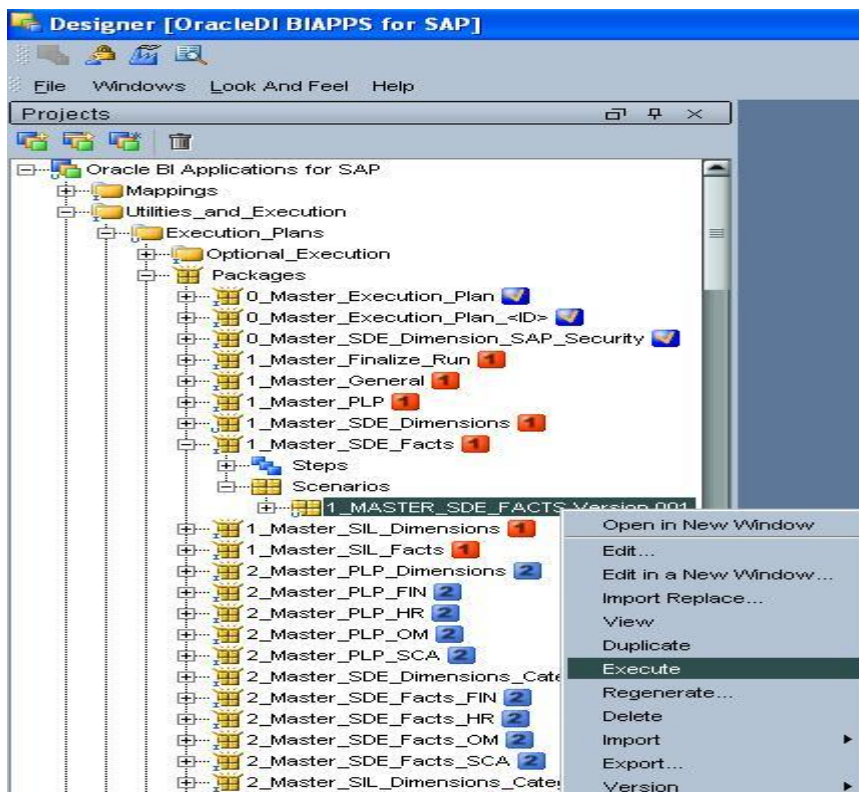
3.Go to Designer->Oracle BI Applications for SAP->Utilities_and_Execution-> Execution_Plans->Packages->1_MASTER_GENERAL package and execute it.

Figure 88



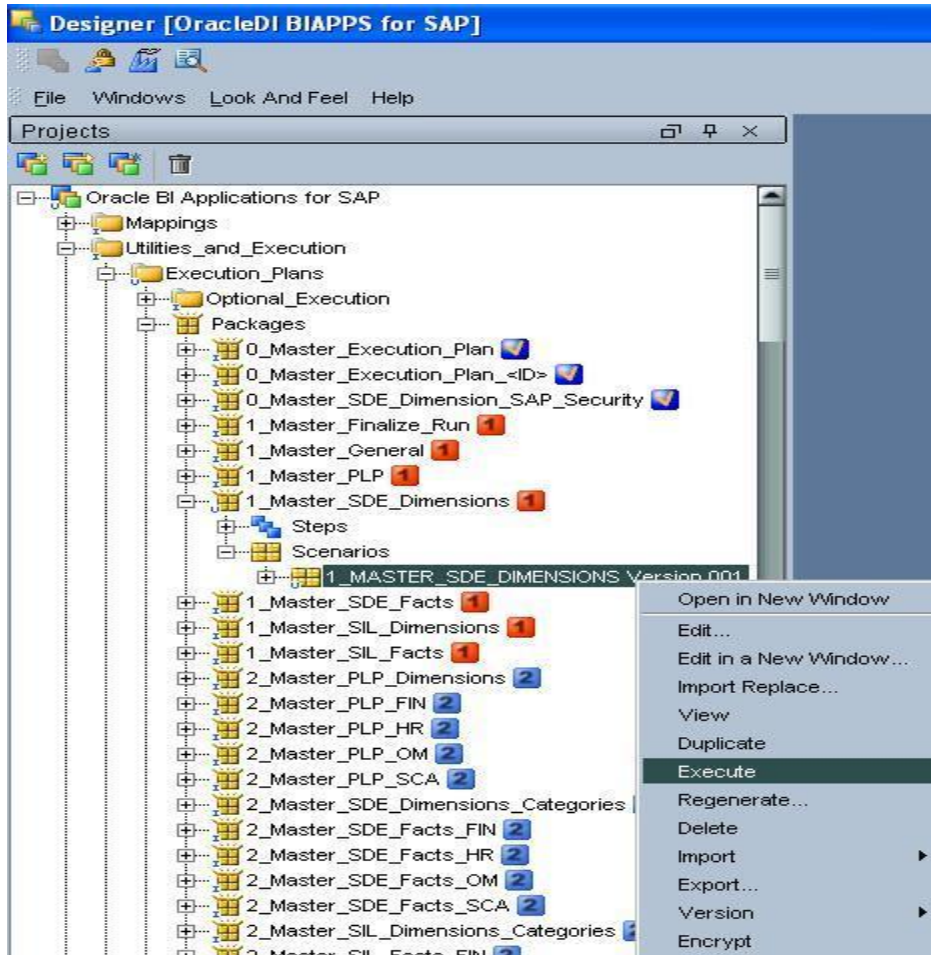
4. Once the above completes successfully in the operator, execute the 1_MASTER_SDE_FACTS package.

Figure 89



5. Once the above completes successfully in the operator, execute the 1_MASTER_SDE_DIMENSION package.

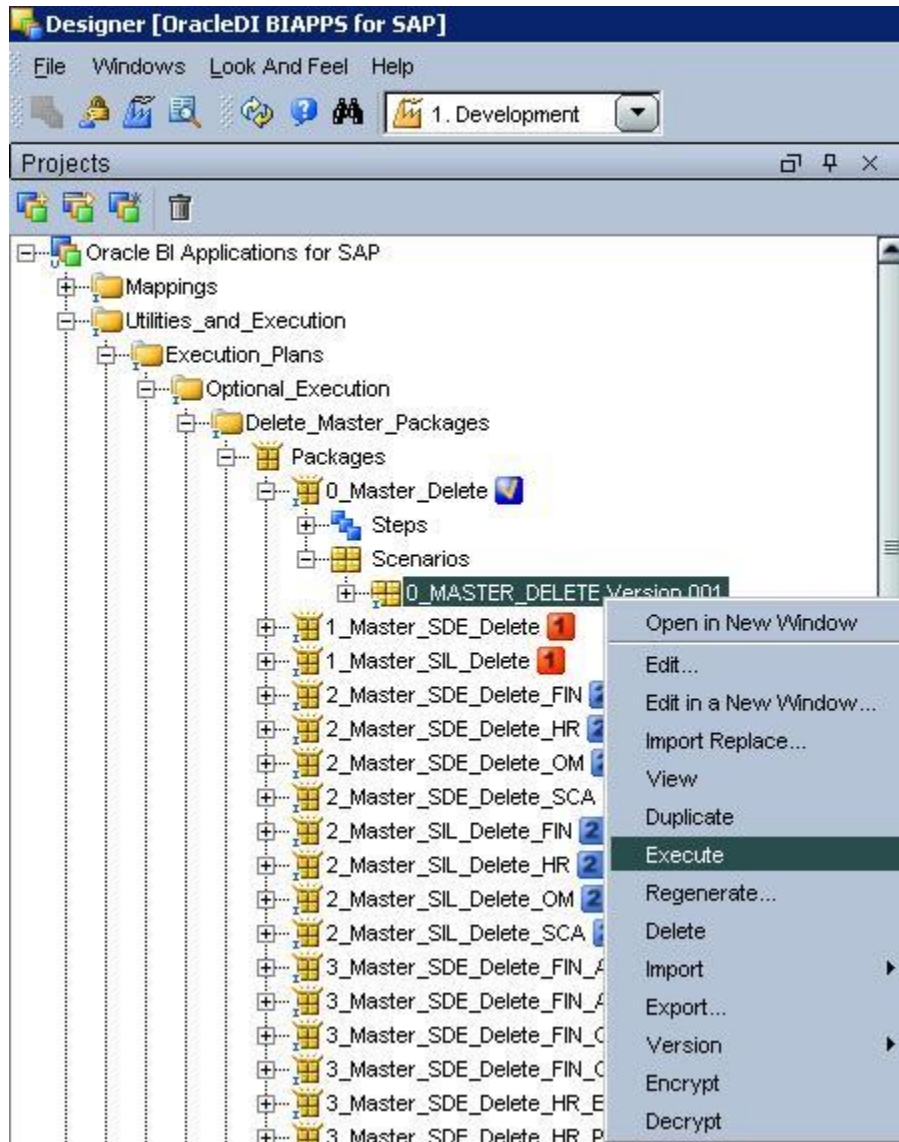
Figure 90



6. The above steps will install the SAP programs that are required for the packages for full load selected in "Execution Plan" in config manager.

7. Execute the 0_MASTER_DELETE this package is required for hard deletes in source system

Figure 91.



8. Once the upload ABAP programmes for full load is complete . Go to the OCM Administer ELT Parameters and change the ETL_PROC_WID to “2” and then save

9. Repeat steps 3,4,5& 7. This will install the SAP programs that are required for the packages for incremental load selected in “Execution Plan” in config manager And optional execution for hard deletes in source system

Once the above steps are completed successfully, go to topology change the SAP Allow ABAP execute to its default value 1.

Also change the SAP Allow ABAP upload to 0 now.

Change the maximum number of sessions on INTERFACE agents back to 3.

Change the ETL_PROC_WID to “ 1” in OCM Administer ELT Parameters

Delete all entries from C_LOAD_DATES table in your data schema(DATA_BIAPPS) manually.

Truncate W_ETL_RUN_S table from data schema(DATA_BIAPPS) manually.

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

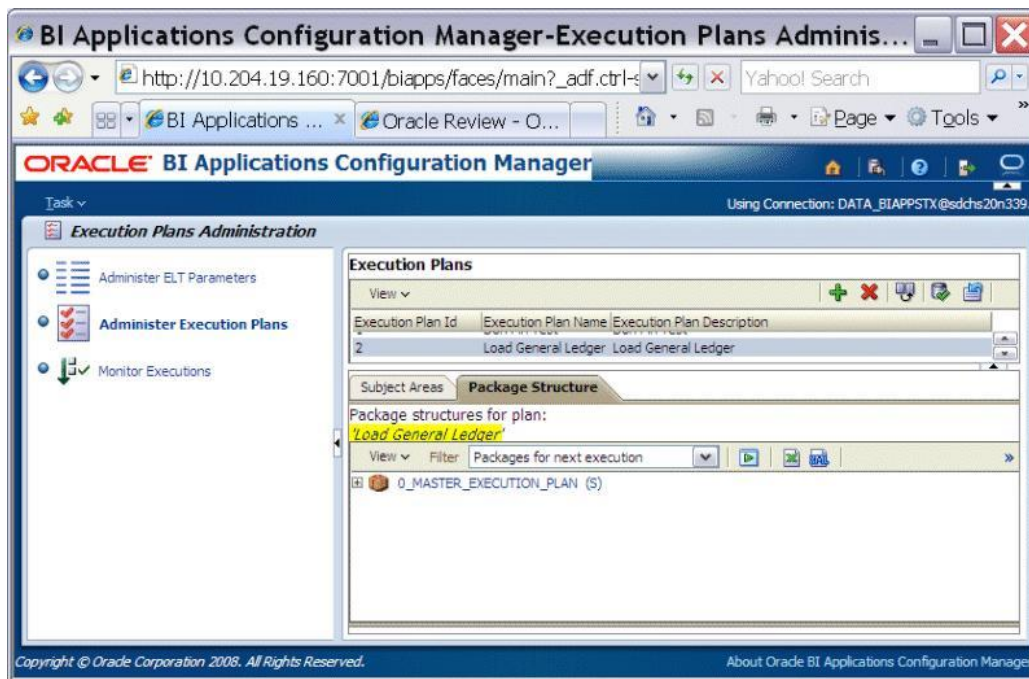
You run an Execution Plan to load data into the Oracle Business Analytics Warehouse for the Applications and Subject Areas specified by the Execution Plan.

Note: For the data to be loaded in the Warehouse, please ensure that uploading of all the required ABAP Programmes as per the Subject Areas selected in create Execution plan is completed. In case any change is done in Execution Plan then the ABAP Programs have to be uploaded as per the section "How to Upload SAP ABAP Programs".

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

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

Figure 92.



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

Figure 93.

The screenshot shows a Windows-style dialog box titled "Execute Package Structure". It contains three main sections:

- ODI Workflow Agent:**
 - * Host:
 - * Port:
- ODI Context:**
 - * Context Code:
- ODI Authentication:**
 - * User:
 - * Password:

At the bottom right, there are "OK" and "Cancel" buttons.

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

Table 29. *Execute Package Structure dialog fields*

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

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

Notes

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

Figure 94.



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

How to perform E-LT Using ODI Designer

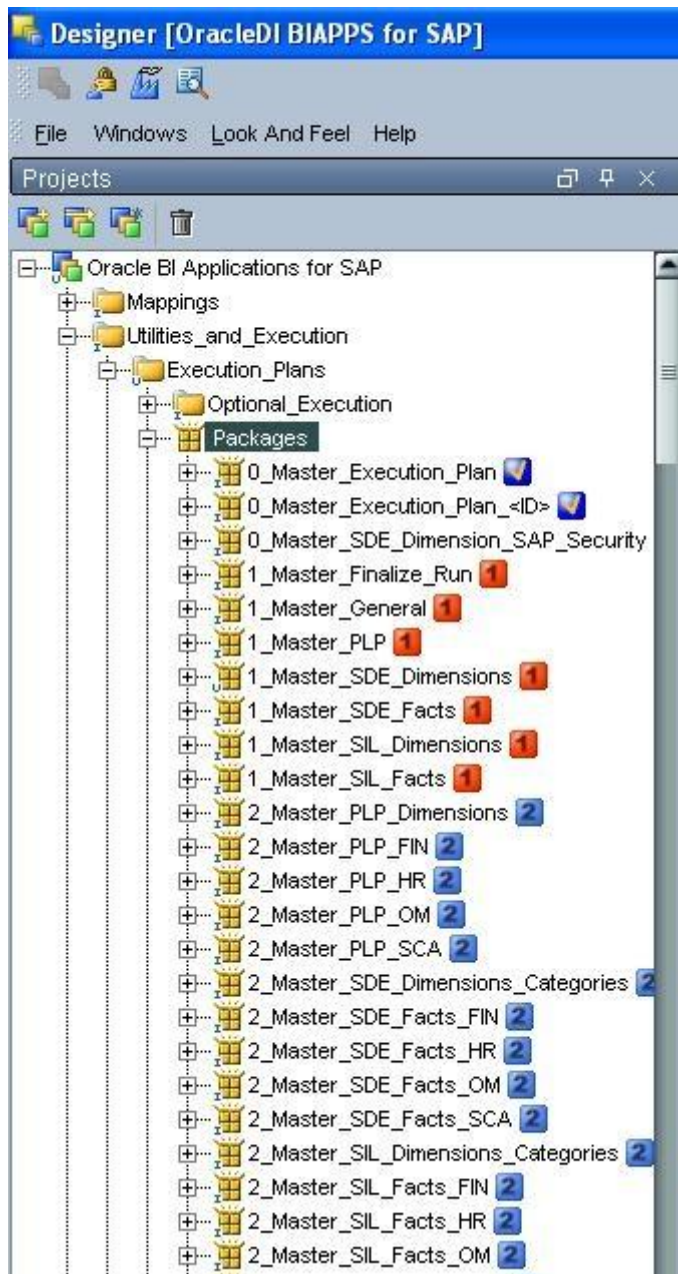
You can use ODI Designer to run an Execution Plan once or schedule it to perform more than once.

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

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

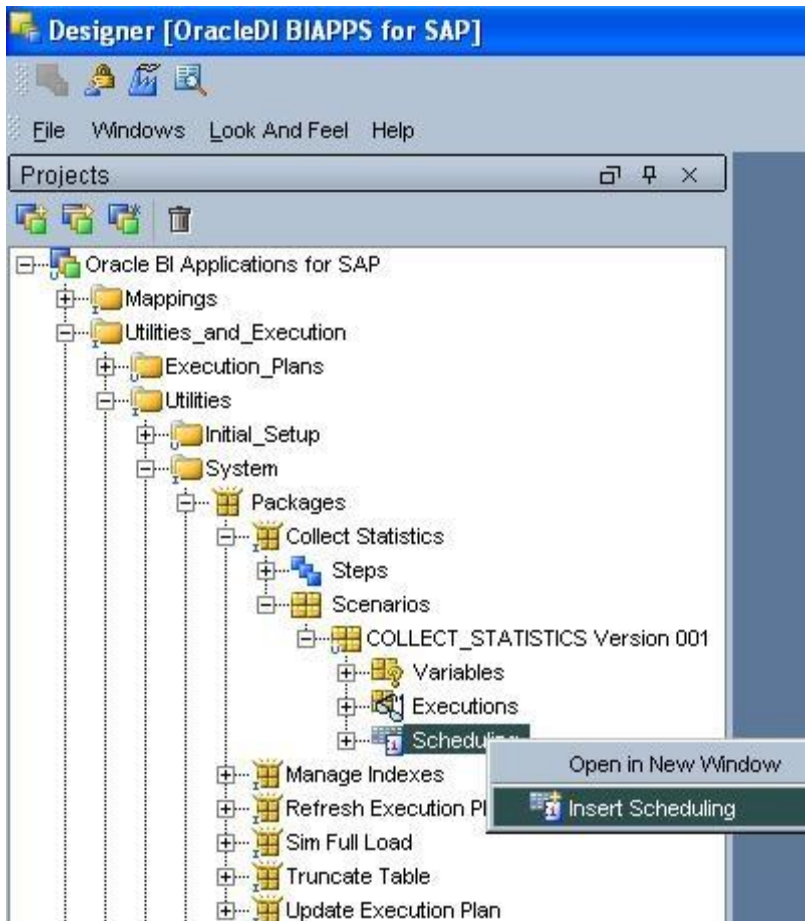
1. In ODI Designer, display the Projects view.
2. Expand the Oracle BI Applications for SAPproject.
3. Select Utilities_and_Execution, then Execution Plans, then Packages.

Figure 95



4. Right-click on 0_Master_Execution_Plan_<ID> and choose Duplicate. ODI Designer creates a copy of the package '0_Master_Execution_Plan_<ID>' named 'Copy of 0_Master_Execution_Plan_<ID>'. You can optionally rename this copy with a suitable name, for example, 'Execution Plan for HR'.
5. Double-click on the copy of the package to display the Package: <Name> dialog for this package.
6. Display the Diagram tab.
7. Select the 'Update Execution Plan' procedure at the top of the Diagram pane, then display the Options tab on the Properties pane below.

Figure 96.



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

Loading Source Data Using an Execution Plan

8. On the Options tab, use the Value field to set the value of the OBI_EXECUTION_PLAN property. You set the value of the OBI_EXECUTION_PLAN field to the same as the unique Execution Plan Id value assigned to the Execution Plan in Oracle BI Applications Configuration Manager.

For example, if an Execution Plan that you create in Oracle BI Configuration manager is assigned the ID '470', to run that Execution Plan you set the value of the OBI_EXECUTION_PLAN field to '470'. For more information about Execution Plan IDs, see "How to Create an Execution Plan In Oracle BI Applications Configuration Manager").

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

Notes

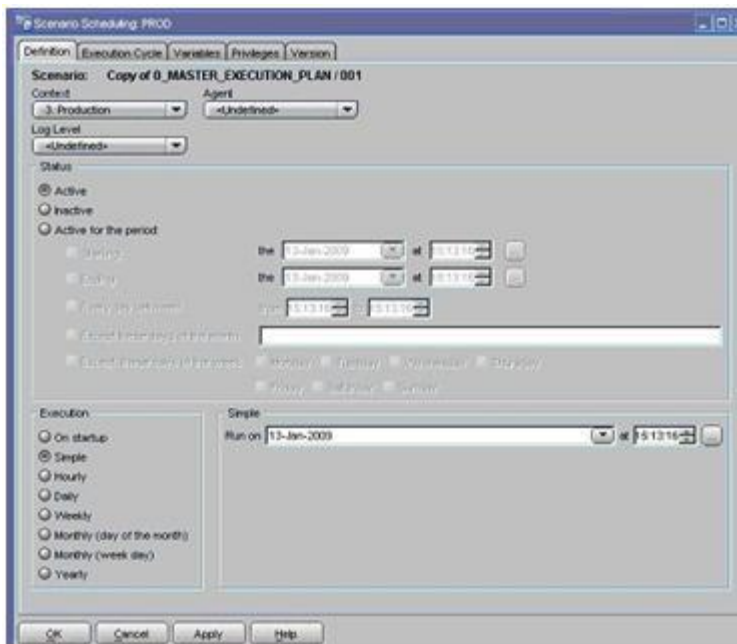
- To run multiple execution plans, copy the 0_Master_Execution_Plan_1 package, edit the package details, and set the OBI_EXECUTION_PLAN setting to the same as the unique Execution Plan Id value assigned to the Execution Plan in Oracle BI Applications Configuration Manager.

10. Do one of the following:

- To perform a single full-load E-LT process:

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

Figure 97.



- b. Expand the Scenario for that package.
- c. Right-click on the Scheduling node and select Insert Scheduling to display the Scenario Scheduling: <Name> dialog.
- d. Use the Scenario Scheduling: <Name> dialog to specify the scheduling details.
- e. Save the details.
- f. When the process starts, use ODI Operator to monitor the process. For more detailed information about using the ODI to perform E-LT, refer to the Oracle Data Integrator documentation.

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

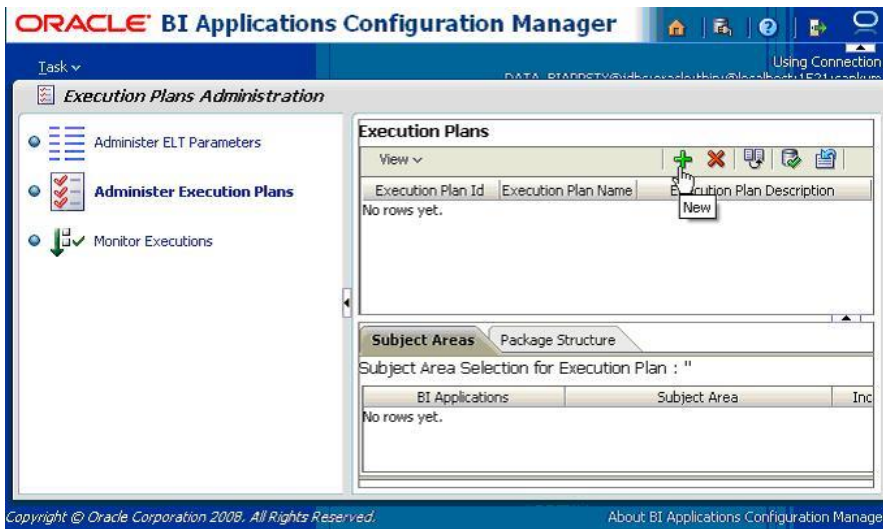
Example of Running a Full Load E-LT in Oracle BI Applications Configuration Manager

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

In this example, you have installed Oracle BI Applications with an SAP Application data source, and you want to load OLTP data for the subject area General Ledger, as follows:

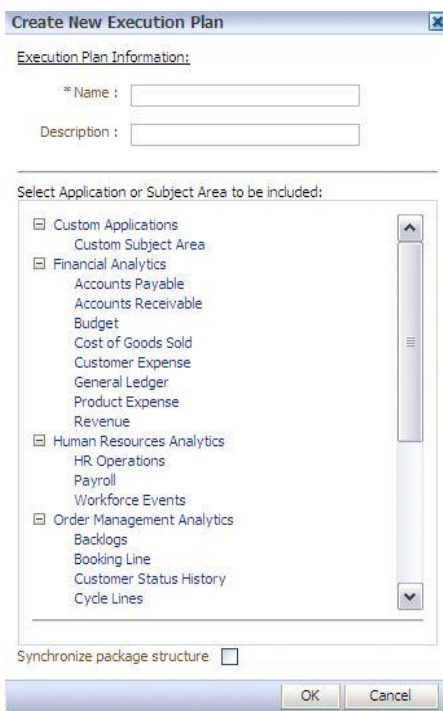
1. Start Oracle BI Applications Configuration Manager (or more information, see "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the Administer Execution Plans link.

Figure 98.



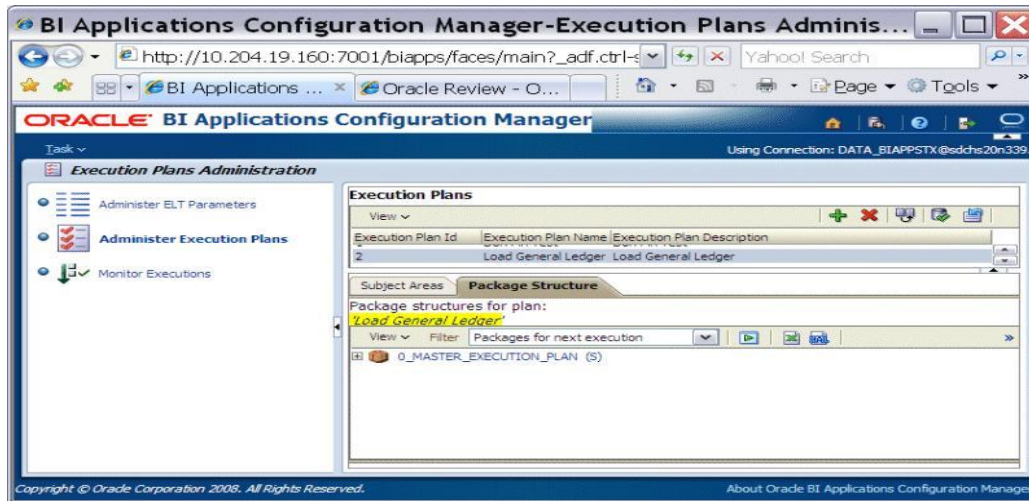
3. Click the New button (+) to display the Create New Execution Plan page.

Figure 99.



4. Enter 'Financials - General Ledger' in the Name field.
5. In the Select Application or Subject Area to be included box, select General Ledger.
6. Click OK.
7. Select the Administer Execution Plans link.
8. In the Execution Plans list, select the 'Financials - General Ledger' Execution Plan.
9. Display the Package Structure tab.
10. Upload ABAP Program as described in the section "How to Upload SAP ABAP Programs".

Figure 100.



11. Click the Execute button to display the Execute Package Structure dialog.

Figure 101.



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

Figure 102.

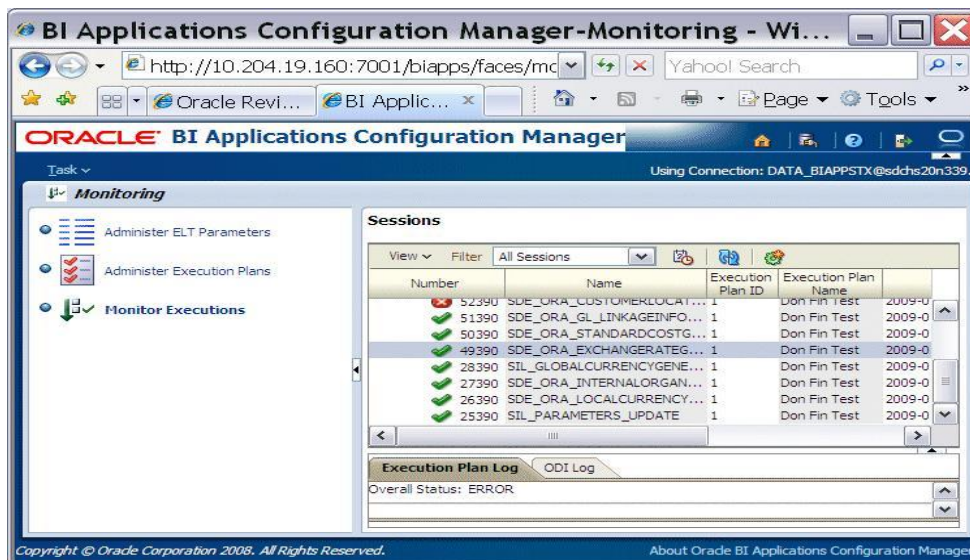


Table 30. *Execute Package Structure dialog fields*

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

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

Managing Execution Plans

The Execution Plans Administration page enables you to create and maintain your Execution Plans for E-LT runs. To navigate to the Execution Plans Administration page, either click on the Administer Execution Plans link from the left Navigation pane or click on the link from the Home page or choose the Administer Execution Plans menu item from the application menu.

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

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

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

For an example of creating and running an Execution Plan, see "Example of Running A Full Load E-LT in Oracle BI Applications Configuration Manager". For more information about re-starting an Execution Plan (for example, after diagnosing and fixing an E-LT error), see "How to Restart Execution Plans After Error Correction".

Editing and Deleting Execution Plans

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

Monitoring E-LT Processes

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

- How to Restart Execution Plans After Error Correction

How to Monitor E-LT Processes

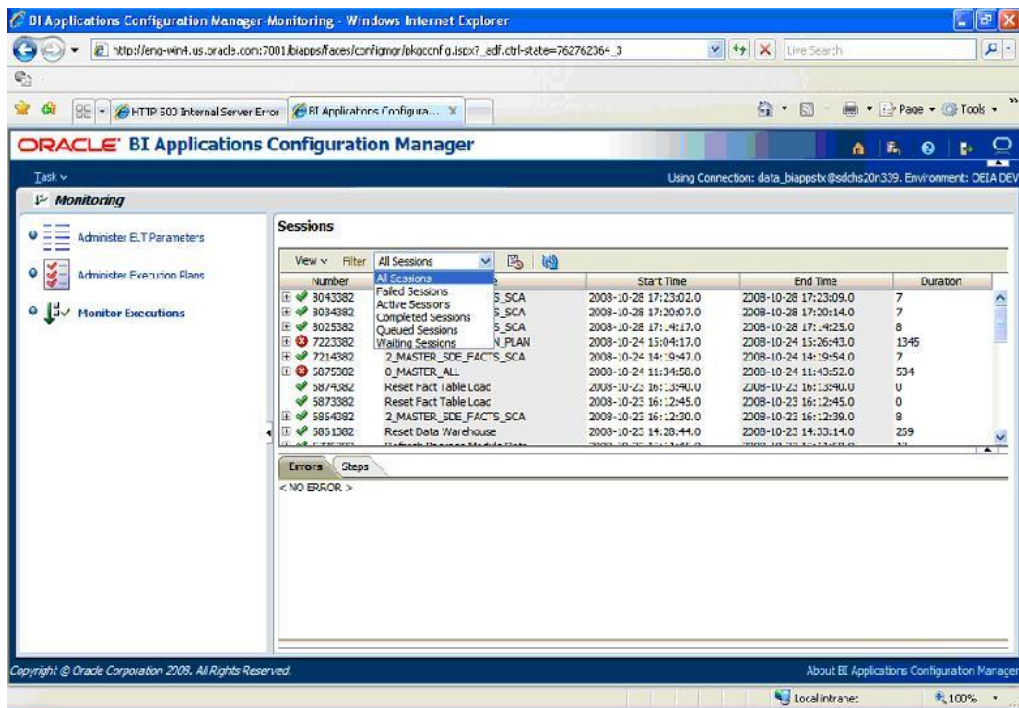
- You can use Oracle BI Applications
- How to Restart Execution Plans After Error Correction

How to Monitor E-LT Processes

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

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

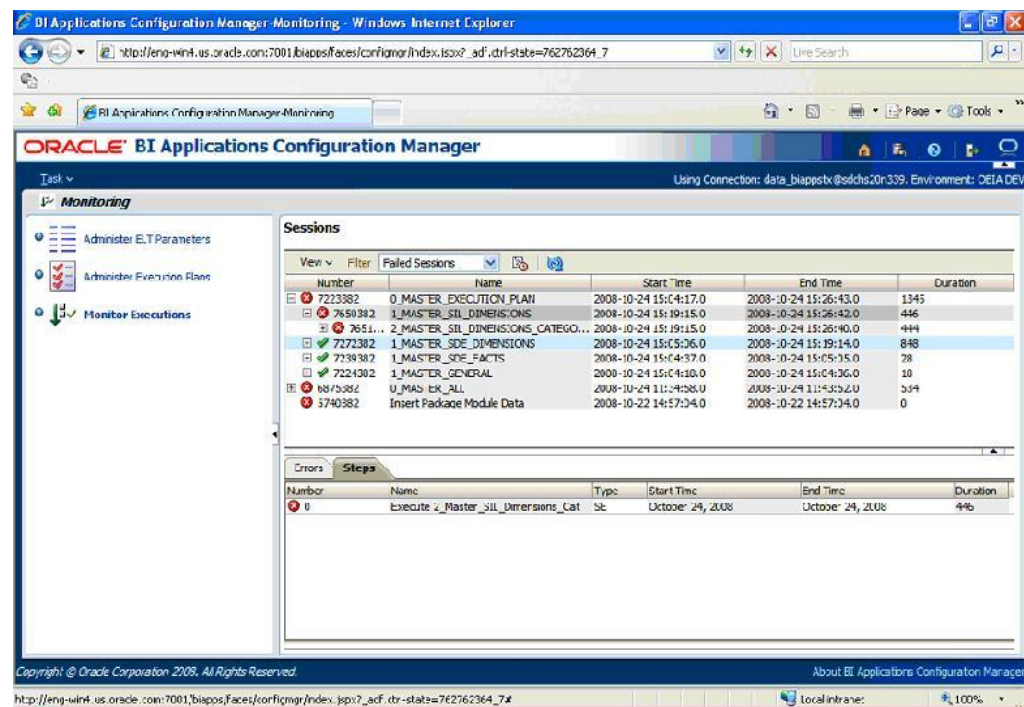
Figure 103.



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

[illegible]

Figure 105.



How to Restart Execution Plans After Error Correction

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

To restart an Execution Plan after error correction:

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

Miscellaneous and Supporting Tasks

This topic contains the following topics:

- How to verify installation and Set up
- Using ODI Designer to Manage Your Environment
- Using Oracle BI Applications Configuration Manager
- About Resetting the Default Passwords, Schema Names, and ODI Connection Details
- How to configure and activate Automated Database Statistics Collection
- How to configure and activate Automated Delete Handling
- About Data Warehouse Loads
- How to reset the Oracle Business Analytics Warehouse for Full Load
- How to add partitions to Data Warehouse tables
- About deploying ODI across multiple environments
- How to run an ODI Agent as a Unix background process
- How to uninstall ODI Agent Windows Services
- How to Resolve Conflicts in ODI Agent Port Numbers
- How to activate or deactivate Flow Control in ODI
- List of Log Files

How to verify installation and Set up

To verify and Oracle Business Intelligence Applications environment, do the following:.

1. Log into ODI Designer or ODI Topology Manager as user SUPERVISOR.
2. Use ODI Designer to connect to the Work Repository.
3. Use ODI Topology Manager to check the Oracle Data Server Connections.

4. Use ODI Topology Manager to check the Agents.
5. Use ODI Topology Designer to check that the Model objects are available.

Using ODI Designer to Manage Your Environment

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

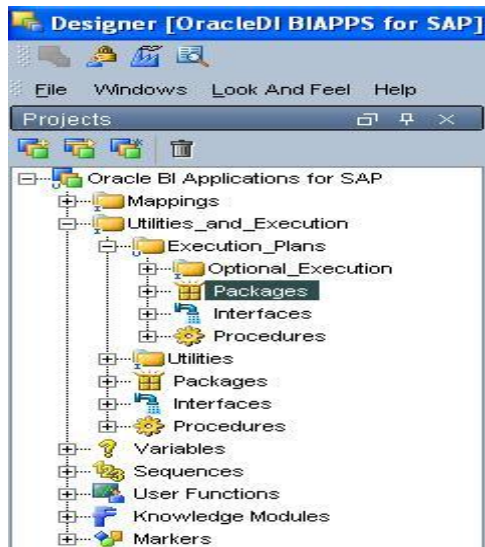
Note: To manage Oracle Business Intelligence Applications environment in ODI, you must log in as user 'SUPERVISOR'.

Using the Projects view

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

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

Figure 106. The main Oracle BI Applications project in ODI Designer



- a. The main Oracle BI Applications for SAP project (for more information, described in the chapter titled "About the Oracle BI Applications for SAP project").
- b. The Mappings folder (for more information, see "About the Mappings folder").
- c. The Execution Plans folder (for more information, see "About the Execution Plans folder").
- d. The Utilities folder (for more information, see "About the Utilities folder").

About the Oracle BI Applications project

The project contains all of the ODI project components required by Oracle Business Intelligence Applications.

Figure 107. The Project Folder



About the Mappings folder

The Mappings folder contains the Packages and Interfaces used by Oracle Business Intelligence Applications to perform E-LT. For example, the SDE_SAPECC6_Adaptor folder contains a sub-folder for every Application and Subject Area combination containing Packages and Interfaces for that Subject Area.

Figure 108



The Mappings folder contains the following main sub-folders:

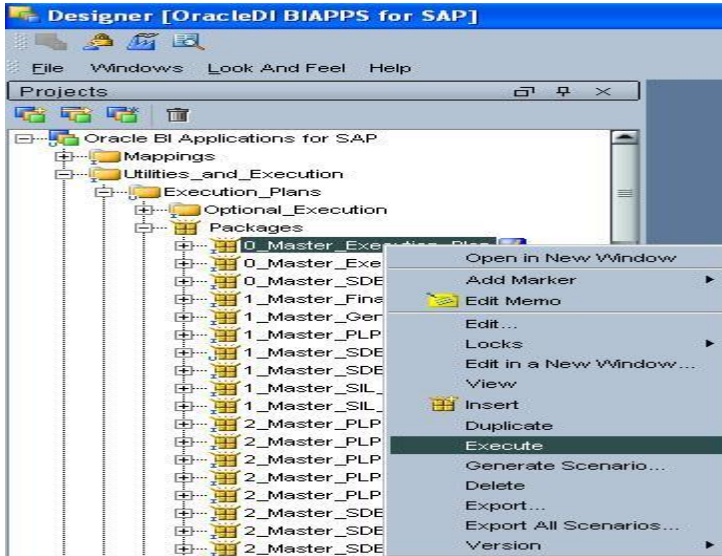
- PLP - Contains the Post Load Processing (PLP) Mappings.
- SDE_Universal_Adaptor - Contains the Source-Dependent Extract (SDE) mappings for Universal Adaptor.
- SILOS - Contains the Source Independent Load (SIL) mappings.
- SDE_SAPECC6_Adaptor - Contains the Source-Dependent Extract (SDE) mappings for SAP ERP ECC6.0.

- SDE_SAP46C_Adaptor - Contains the Source-Dependent Extract (SDE) mappings for SAP ERP 4.6c

About the Execution Plans folder

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

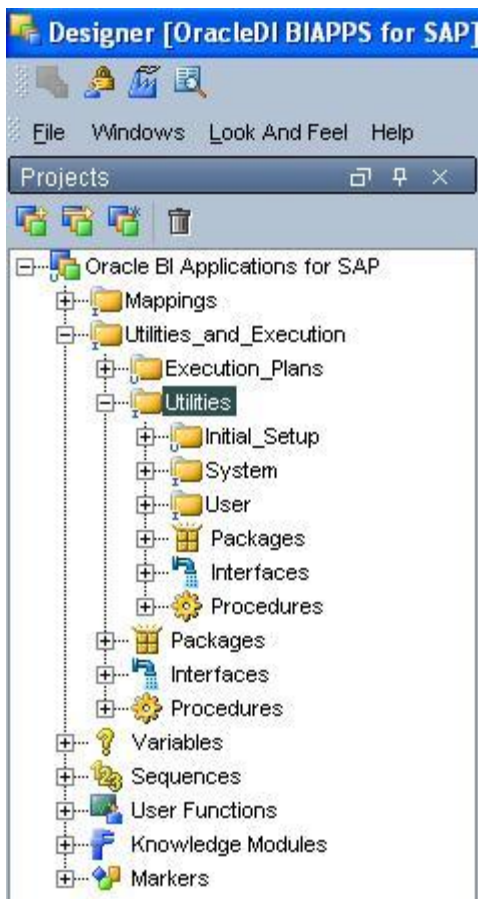
Figure 109



About the Utilities folder

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

Figure 110



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

Oracle Business Intelligence Applications installs with the following models:

- Flat File Lookups - Contains definitions (metadata) pertaining to the lookup flat file data stores. These objects are mapped to the files in the location `$ODI_HOME/oracledi/biapps_odi/odifiles/odidatafiles/lkpfiles`.
 - Oracle BI Applications for SAP- Contains definitions (metadata) of all the Oracle Business Analytics Warehouse objects.
 - Source Files - Contains definitions (metadata) of all the source flat-file data stores. These objects are mapped to the files in the location `$ODI_HOME/oracledi/biapps_odi/odifiles/odidatafiles/srcfiles`.
- SAP ECC 6.0 - Contains definitions (metadata) of all the SAP ERP 6.0 data source objects.
- SAP 4.6C - Contains definitions (metadata) of all the SAP ERP 4.6c data source objects.
- SAP Lookup Files - Contains definitions (metadata) pertaining to the lookup flat file data store for SAP
- SAP Src Files - Contains definitions (metadata) of all the SAP source flat-file data stores.

Chapter 5: Configuring Common Areas and Dimensions

This chapter contains configuration steps for Oracle Business Intelligence Applications that you need to follow for any applications you deploy (for example, Oracle Financial Analytics, Oracle Human Resources Analytics, Oracle Procurement and Spend Analytics), and contains the following topics:

- Steps Required Before a Full Load

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

Steps Required Before a Full Load

This section contains configuration steps required before a full data load that apply to Oracle Business Intelligence Applications deployed with an SAP R/3 release 4.6c and ECC 6.0 source system, and contains the following topics:

- How to Configure Initial Extract Date
- Configuring Global Currencies
- How to Configure Fiscal Calendar
- Configuring the General Ledger Account Hierarchies
- Configuring Product Hierarchy, Master Inventory Organization, and Group Accounts
- Configuration Steps for Controlling Your Data Set

How to Configure Initial Extract Date

Initial Extract Date is required when you extract data for a full load. It reduces the volume of data in the initial load. The specified initial extract date will be used as a filter on the creation date of OLTP data in the selected full extract mapping. When you set the Initial Extract Date parameter, make sure that you set it to the beginning of an accounting period, and not a date in the middle of an accounting period. For example, if you decide to extract data from June 2005, and the June 2005 accounting period starts from 5th June, set the date to 5th June, 2005.

To configure the initial extract date in your warehouse:

1. Start Oracle BI Applications Configuration Manager (for more information, see "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the Administer ELT Parameters link.
3. Display the Global tab.
4. Locate the following parameter and use the Parameter Value field to set the value:
 - INITIAL_EXTRACT_DATE
5. Save your changes.

For more information about specifying parameter values, see "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

Configuring Global Currencies

Currency conversions are required because your business might have transactions involving multiple currencies. To create a meaningful report, you have to use a common currency. The Oracle Business Analytics Warehouse stores amounts in the following currencies:

- Document currency. The currency of the transaction. For example, if you purchase a chair from a supplier in Mexico, the document currency is probably the Mexican Peso. Or, if you made a business trip to the UK and filed an expense report for meal expenses in the UK, the document currency of the expense report will be in GBP.
- Local currency. The currency defined in the company code is Known as local currency in SAP.
- Global currencies. Leading ledger currency (with reference to 4.6 C there is only one ledger which is a leading ledger) is Global Currency in SAP used for reporting consolidated financial statements.

To configure Global Currencies, follow the steps in "How to Configure Global Currencies "

The load mapping also loads the exchange rates required to convert the document amount into each of the three global currencies. In the target table, there will be two amount columns, and three exchange rate columns. Typically, the source system provides the document currency amount, which is the default currency handling setup for the Oracle Business Analytics Warehouse. If the source system provides only the document currency amount, the Source Adapter performs lookups to identify the local currency codes based on the source system. Based on the source system the appropriate currencies are assigned. After the lookups occur, the extract mapping provides the load mapping with the document currency amount and the document and local currency codes. The load mapping will then use the provided local currency codes and perform currency conversion to derive the local amount. The load mapping will also fetch the global currencies that are specified by parameters in Oracle BI Applications Configuration Manager and look up the corresponding exchange rates to each of the three global currencies.

To configure Exchange Rate Types, follow the steps in How to Configure Exchange Rate Types How to Configure Exchange Rate Types.

How to configure Global Currencies

To configure global currencies:

1. Start Oracle BI Applications Configuration Manager (for more information, see "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the Administer ELT Parameters link.
3. Display the Common tab.
4. Locate the following parameters and use the Parameter Value field to set the value:
 - GLOBAL1_CURR_CODE
 - GLOBAL2_CURR_CODE
 - GLOBAL3_CURR_CODE

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

Check the table entry T000 in SAP for field "MWAER", same currency key value should be setup as Global Currencies in warehouse for reporting during installation. If the value is blank then SAP system has not been configured for Global Currency.

Make sure all 3 GLOBAL_CURR_CODE in the table C_PARAM_VALUE have the same value as "MWAER" field's value in SAP retrieved from table T000.

5. Save your changes.

For more information about specifying parameter values, see "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

How to Configure Exchange Rate Types

When Oracle Business Intelligence Applications converts your transaction records' amount from document currency to global currencies, it also requires the exchange rate types to use to perform the conversion. For each of the global currencies, Oracle Business Intelligence Applications also allows you to specify the exchange rate type to use to perform the conversion. Oracle Business Intelligence Applications also provides three global exchange rate types for you to configure.

Oracle Business Intelligence Applications also converts your transaction records' amount from document currency to local currency. Local currencies are the base currencies in which your accounting entries and accounting reports are recorded. In order to perform this conversion, Oracle BI Application also allows you to configure the rate type that you want to use when converting the document currency to the local currency.

There should be no conversion from document currency and local currency in warehouse as SAP has a separate field to store local currency amount which are available in all facts.

By not converting using exchange rate table, we will not have discrepancy between SAP and warehouse due to records which are created using manually inputting exchange rate other than what we have value in exchange rate table.

To configure the exchange rate types:

1. Start Oracle BI Applications Configuration Manager (for more information, see "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the Administer ELT Parameters link.
3. Display the Common tab.
4. Locate the following parameters and use the Parameter Value field to set the value:
 - GLOBAL1_RATE_TYPE
 - GLOBAL2_RATE_TYPE
 - GLOBAL3_RATE_TYPE
 - DEFAULT_LOC_RATE_TYPE (the conversion rate type for document currency to local currency conversion).

Make sure that you spell the exchange rate type values as they are spelled in your source OLTP system. For example, Corporate.

How to Find Global Exchange Rate in SAP

Use transaction se11 and enter table T000 (Clients). Enter the client number you want to work with.

The field value T000.MWAER (Std currency) is the Global exchange rate type.

5. Save your changes

For more information about specifying parameter values, see "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

How to Configure Fiscal Calendar

- Overview of Calendars in Oracle BI Applications
- About Configuring Calendars
- About Configuring Enterprise Calendars
- Using SAP ERP Application to configure the Enterprise calendar
- About Configuring Warehouse Generated Fiscal Calendars
- Setting Up Fiscal Calendar By Fiscal Week
- Setting Up Fiscal Calendar By Fiscal Month
- How to Set Up The Fiscal Calendar
- Examples of Configuring the Universal Adapter
- Reloading the Time Dimension Tables After Your Data Warehouse is Populated

Overview of Calendars in Oracle BI Applications

Oracle Business Intelligence Applications Version 7.9.7.2 supports the following calendar formats:

- Enterprise (Global) - cross functional reporting calendar, which can be fiscal or gregorian.
- Fiscal - accounting or financial calendar.
- Gregorian - regular calendar that starts on January 1st and ends on December 31st.
- 13 Period - a calendar is which each year is comprised of 13 periods.
- 4-4-5 - each year is composed of twelve periods of either four weeks of 28 days or five weeks of 35 days.

About Calendar Tables:

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

Gregorian Calendar Tables

- W_WEEK_D
- W_MONTH_D
- W_QTR_D
- W_YEAR_D
- W_DAY_D

Fiscal Calendar Tables

- W_MCAL_WEEK_D
- W_MCAL_PERIOD_D
- W_MCAL_QTR_D
- W_MCAL_YEAR_D

Enterprise Calendar Tables

- W_ENT_WEEK_D

- W_ENT_PERIOD_D
- W_ENT_QTR_D
- W_ENT_YEAR_D

The following table shows Time Dimension Configuration and Context Tables.

Table 31: Time Dimension Configuration and Context Tables

Configuration Table	Context Table
W_MCAL_CONFIG_G	W_MCAL_CONTEXT_G

About Calendar Categories Calendars are categorized into two types:

- OLTP sourced (also known as Source Calendars)
 - OLTP sourced calendars are calendars that are defined in ERP sources and brought into the warehouse via ETL maps.
- Warehouse generated (also known as Generated Calendars)
 - Generated calendars are fiscal calendars generated in the warehouse based on configuration files.

About Configuring Calendars

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

Prerequisites to populating the MCAL tables

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

There are two parameters \$\$START_DATE and \$\$END_DATE for the task SIL_DayDimension that need to be setup to load the calendar data in W_DAY_D. The SIL mappings use standard time functions to create records for each calendar day falling within the boundary defined by these two parameters. Once the records are created in W_DAY_D, the aggregate calendar tables are loaded by their respective SIL mapping. Then the fiscal calendar tables (known as MCAL tables) are populated.

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

About Configuring Enterprise Calendars

An Enterprise calendar (or reporting calendar) enables cross subject area analysis. Enterprise calendar tables have W_ENT prefix.

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

- \$\$GBL_CALENDAR_ID
- \$\$GBL_DATSOURCE_NUM_ID

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

- Using SAP ERP Application to configure the Enterprise calendar

Using SAP ERP Application to configure the Enterprise calendar

Fiscal Calendars from SAP will be loaded from two tables, T009 & T009B. This data is then loaded in the following Oracle Warehouse tables - W_MCAL_CAL_D, W_MCAL_CONTEXT_G & W_MCAL_PERIOD_DS

A csv file, file_mcal_qtr.csv is created from SAP to load fiscal quarter definition. The csv file has the following columns in it.

- Fiscal Year Variant: Identification ID of Accounting Calendar Year.
- Year: Quarter definition per fiscal year for Non-Calendar Year dependent Fiscal Year Variant. Other cases, it contains zeros i.e applicable for all years.
- Qtr: Quarter Identification i.e. first quarter or second etc.
- Qtr Start Period: Classification of Accounting periods into quarter and specify Qtr Start Period.
- Qtr end period: Specify Qtr End Period.
- Client Id: Source System Identification Number.

Steps to be followed for preparing the csv file, file_mcal_qtr.csv :

Purpose: The file helps to define Quarter definition for the fiscal variant. The quarter start period and end period is defined.

- FISCAL YEAR VARIANT: The fiscal year variant is used to define the fiscal year. In the definition, you allocate your posting periods to the calendar year.
- YEAR: The fiscal year variant in SAP can be Year dependent (Calendar Year) or the fiscal year can start from different month of the year. Allocation of posting periods to calendar days must be made individually for each year.
- QTR: Definition value of the Quarter.
- QTR START PERIOD: The Start month of the Quarter.
- QTR END PERIOD: The end month of the Quarter.
- CLIENT ID: A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

Steps to Prepare CSV file.

Use transaction code Se16, and enter table 'T009', field PERIV stores value for Fiscal Variant. Check value for XKALE, if value = X then maintain "0" for the field Year in Excel. This denotes that the fiscal year variant is calendar Year thus the 1 day or month of the calendar will be always applicable. If value is <> X then check value XJABH the value shall be X. Enter table 'T009B'. This table have Year (T009B. BDATJ) defined for the fiscal variant and the 1 month (T009B.BUMON) of the year is equivalent to the period (T009B.POPER) of the fiscal year.

Based on above information the file can be defined.

Figure 111.

Fiscal Year Variant	Year	Qtr	Qtr Start Period	Qtr end period	Client Id
NR	0	1	1	3	800
NR	0	2	4	6	800
NR	0	3	7	9	800
NR	0	4	10	12	800
NV	2007	1	1	3	800
NV	2007	2	4	6	800
NV	2008	1	1	3	800
NV	2008	2	4	6	800
NV	2008	3	7	9	800
NV	2008	4	10	12	800

About Configuring Warehouse Generated Fiscal Calendars

Oracle Business Intelligence Applications Version 7.9.7.2 supports the following types of generated calendars:

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

About the Calendar Context table (W_MCAL_CONTEXT_G)

This table is used by Oracle Financial Analytics and Oracle Project Analytics facts to lookup the calendar ID for a given ledger or OU (Operating Unit). This needs to be populated for the fact tables to get loaded correctly (the default Execution Plan in Config Manager does this).

Notes on Configuring Calendars

When you set up calendars, note the following:

- The W_MCAL_CONFIG_G table controls how generated calendars are created.
- If generating the 4-4-5 or 13 period calendars, W_MCAL_CONFIG_G needs to have at least one row for the 4-4-5 period or 13 period.
- W_MCAL_WEEK_D will be populated only for the generated calendars (i.e. 13 period or 4-4-5 type of calendars), and hence the W_DAY_D week Enterprise columns will be null for non-generated calendars (known as OLTP sourced fiscal calendars). W_ENT_WEEK_D will not be populated if a non-generated calendar is chosen as the Enterprise Calendar.
- For 13 period calendars, there is no concept of Quarter, therefore all Quarter columns in W_MCAL_WEEK_D, W_MCAL_PERIOD_D, W_MCAL_YEAR_D will be null. W_ENT_QTR_D will not be populated if a 13 period calendar is chosen as the Enterprise Calendar.
- The following table outlines columns in the W_MCAL_CONFIG_G table, which is loaded from the file_mcal_config_g.csv

Table 32: Configuring Calendars

Column Name	Column Description
CALENDAR_ID	The ID of the calendar that is being configured. This is the primary key for this table

CALENDAR_NAME	The name of the calendar that is being configured
CALENDAR_CLASS	Automatically generated
PERIOD_TYPE	The type of the calendar period that is being configured e.g. 4-4-5
CAL_ST_DT	The date from which the calendar generation begins.
CAL_END_DT	The date at which the calendar generation ends.
CAL_OFFSET	The offset which identifies the start date of the calendar
WEEK_ALLOCATION_RULE	This parameter determines how weeks are allocated in the calendar that is being configured
OTHER Standard Columns	W_INSERT_DT, W_UPDATE_DT, TENANT_ID, X_CUSTOM etc.

- 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'

Setting Up Fiscal Calendar By Fiscal Week

In this option you provide data for the Fiscal Year, Fiscal Month, Fiscal Week and Start Date of Fiscal Week. The Fiscal Month information is derived using the 4-4-5 rule for grouping weeks into months. The Fiscal Week End Date is derived based on the start date of the next week that is supplied in the data file. Fiscal Months are grouped into sets of 4 months each to determine the Fiscal Quarter.

Setting Up Fiscal Calendar By Fiscal Month

In this option you can provide data at the level of Fiscal Month. The SIL_DayDimension_FiscalMonth_Extract task divides the Fiscal Month into Fiscal Weeks of seven days

each. If the number of days in the Fiscal Month is not in multiples of seven, the last week will have less number of days.

How to Set Up The Fiscal Calendar

To set up Fiscal Calendar by Fiscal Week:

1. Open the file `fiscal_week.csv` using a text editor in the `$ODI_HOME\biapps_odi\odifiles\odidatafiles\srcfiles` directory.
2. Enter the Fiscal Year, Fiscal Month, Fiscal Week and the Start Date of Fiscal Week in the format `YYYYMMDD`. Records must be inserted in ascending order of Fiscal Year, Fiscal Month, Fiscal Week.
3. Save the `fiscal_week.csv` file.
4. In Oracle BI Applications Configuration manager, click Parameters, display the Global tab, and do the following:
 - Set the value of `LOAD_FISCAL_CAL_BY_WEEK` to Y.
 - Set the value of `LOAD_FISCAL_CAL_BY_MONTH` to N.

To set up Fiscal Calendar by Fiscal Month:

5. Open the file `fiscal_month.csv` using a text editor in the `$ODI_HOME\biapps_odi\odifiles\odidatafiles\srcfiles` directory.
6. Enter the Fiscal Year and Fiscal Month and the Start Date of Fiscal Month in the format `YYYYMMDD`. Records must be inserted in ascending order of Fiscal Year and Fiscal Month.
7. Save the `fiscal_month.csv` file.
8. In Oracle BI Applications Configuration manager, select the Administer ELT
9. Parameters link, display the Global tab, and do the following:
 - Set the value of `LOAD_FISCAL_CAL_BY_WEEK` to N.
 - Set the value of `LOAD_FISCAL_CAL_BY_MONTH` to Y.

Examples of Configuring the Universal Adapter

The Universal Adapter is provided to allow data from SAP ERP Application to be brought into the Multiple Calendar tables. The data from these sources needs to be brought in using the following via CSV files or SAP tables in some cases

`file_mcal_qtr_SAPECC6.0.csv` - loads `W_MCAL_PERIOD_DS` (Staging Table).

`file_mcal_config_g.csv` - loads `W_MCAL_CONFIG_G`.

Steps to be followed for preparing the csv file, `file_mcal_qtr.csv` :

Purpose: The file helps to define Quarter definition for the fiscal variant. The quarter start period and end period is defined.

FISCAL YEAR VARIANT: The fiscal year variant is used to define the fiscal year. In the definition, you allocate your posting periods to the calendar year.

YEAR: The fiscal year variant in SAP can be Year dependent (Calendar Year) or the fiscal year can start from different month of the year. Allocation of posting periods to calendar days must be made individually for each year.

QTR: Definition value of the Quarter.

QTR START PERIOD: The Start month of the Quarter.

QTR END PERIOD: The end month of the Quarter.

CLIENT ID: A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

Steps to Prepare CSV file.

Use transaction code Se16, and enter table 'T009', field PERIV stores value for Fiscal Variant. Check value for XKALE, if value = X then maintain "0" for the field Year in Excel. This denotes that the fiscal year variant is calendar Year thus the 1 day or month of the calendar will be always applicable. If value is <> X then check value XJABH the value shall be X. Enter table 'T009B'. This table have Year (T009B. BDATJ) defined for the fiscal variant and the 1 month (T009B.BUMON) of the year is equivalent to the period (T009B.POPER) of the fiscal year.

Based on above information the file can be defined.

Figure 112

Fiscal Year Variant	Year	Qtr	Qtr Start Period	Qtr end period	Client Id
NR	0	1	1	3	800
NR	0	2	4	6	800
NR	0	3	7	9	800
NR	0	4	10	12	800
NV	2007	1	1	3	800
NV	2007	2	4	6	800
NV	2008	1	1	3	800
NV	2008	2	4	6	800
NV	2008	3	7	9	800
NV	2008	4	10	12	800

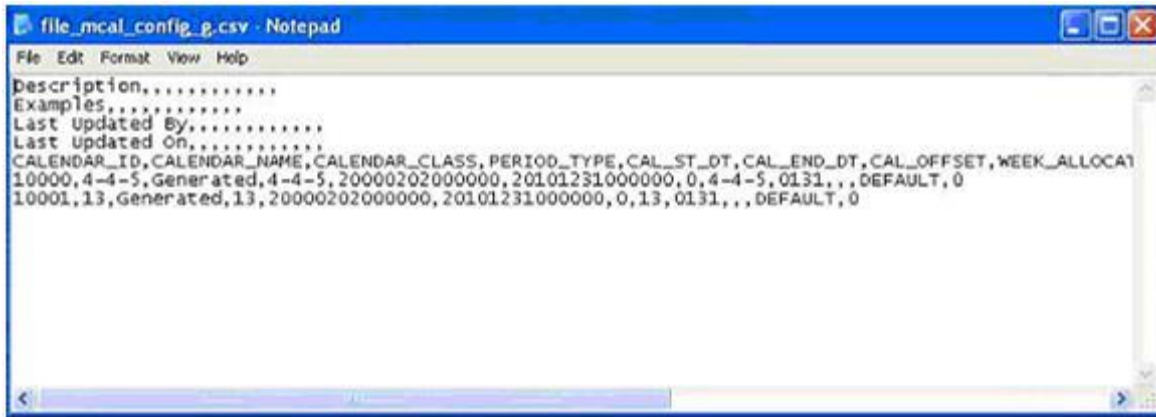
Table 31. Tables in base table W_DAY_D that are used to store calendar information

Regular calendar tables in W_DAY_D	Fiscal calendar tables in 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_Day Dimension that need to be setup to load the calendar data in W_DAY_D. The SIL mappings use standard time functions to create records for each calendar day falling within the boundary defined by these two parameters. Once the records are created in W_DAY_D, the aggregate calendar tables are loaded by their respective SIL mapping.

Example of file_mcal_config_g.csv

Figure 113



Notes:

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

Reloading the Time Dimension Tables After Your Data Warehouse is Populated

The data in the time dimension is loaded once during the initial full load. Subsequently, the SIL_*_UpdateFlag mappings run every day to update the domain value codes, which indicate whether a day, week, month, quarter, or year is 'Current', 'Next' or 'Previous' as of the current day. The SIL_Fiscal_UpdateFlag mappings also update the flags that indicate whether a fiscal week, month, quarter or year is 'Current', 'Previous' or 'Next' with respect to the system date. You might want to extend the range of data that you have in your time dimension some time after the warehouse is in production. In order to achieve this, please follow the steps below that will start a full load ELT run of the W_DAY_D and all of the aggregate time dimension tables.

To set up the load strategy of the time dimension table:

1. Edit the C_LOAD_DATES table in the data warehouse schema.
2. Refresh the dates (ETL_LOAD_DATE and LAST_MAX_DATE) for all the rows in the table where TARGET_TABLE_NAME is in one of the following:
 - W_DAY_D
 - W_WEEK_D
 - W_QTR_D
 - W_MONTH_D
 - W_YEAR_D

- W_FSCL_WEEK_D
 - W_FSCL_QTR_D
 - W_FSCL_MONTH_D
 - W_FSCL_YEAR_D
3. In Oracle BI Applications Configuration Manager, select the Administer ELT Parameters link, display the Common tab, and set the START_DATE and END_DATE parameters to specify the new date range.
 4. Save your changes

For more information about specifying parameter values, see [How to Set E-LT Parameters In Oracle BI Applications Configuration Manager](#)

Notes

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

If there is a week (starting on a Sunday and ending on a Saturday) that falls across two Calendar years, the week is counted in both years. For example the week that starts on 12/30/2007 will be counted in both 2007 and 2008. In 2007 the week Start Date will 12/30/2007 and the End Date will be 12/31/2007. In 2008 this will be the first week with Start Date as 01/01/2008 and End Date as 01/05/2008.

- W_DAY_D stores 31 records for each month irrespective of whether the month actually has 31 days or not. If the month actually has less number of days, there will be records with Null values in the Calendar Date and Day Date columns. These extra records are loaded for the calculation of Period Ago metrics in the RPD, and will not affect the ELT or reporting.
- There are some attributes on the W_DAY_D table that are not mapped in the physical layer of the RPD. Therefore, before creating any new attribute in the RPD, check whether the attribute is already available in the physical layer and if it can be mapped directly.

If your fiscal calendar contains more than 12 months, the extra months will be assigned a value of 0 for the Fiscal Quarter. The same holds for the Fiscal Trimester and Fiscal Half values.

Configuring the General Ledger Account Hierarchies

This section is not applicable to SAP

These configuration steps are required if you are deploying Oracle Financial Analytics, Oracle Procurement and Spend Analytics, or Oracle Supply Chain and Order Management Analytics. This section contains the following topics:

- [Configuring General Ledger Account Hierarchies Using General Ledger](#)
- [Configuring General Ledger Account Hierarchies Using Financial Statement](#)

Overview to General Ledger Account Hierarchies

There are two ways to set up hierarchies in Oracle Financial Analytics:

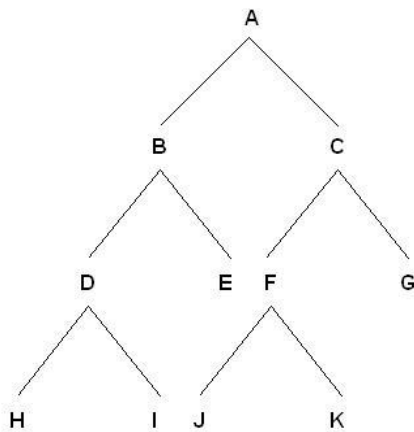
- [Using General Ledger Accounting Flexfield Value Sets Definitions](#) (for more information, see [Configuring General Ledger Account Hierarchies Using General Ledger](#)).
- [Using the Financial Statement Generator \(FSG\) Report Definition](#) (for more information, see [Configuring General Ledger Account Hierarchies Using Financial Statement](#))

Whichever method you choose to set up General Ledger Account hierarchies, you store the hierarchy information in the W_HIERARCHY_D table. As an example, the hierarchy for US Acct might have the following structure:

- Node A has child nodes B and C.
- Node B has child nodes D and E.
- Node C has child nodes F and G.
- Node D has child nodes H and I.
- Node F has child nodes J and K.

The figure below shows the example hierarchy for US Acct.

Figure 114. Example hierarchy US Acct



The table below shows how the hierarchy US Acct would be stored in the W_HIERARCHY_D table.

Table 32. Example hierarchy US Acct stored in W_HIERARCHY_D

HIER_KEY	HIER_ NAME	HIER1_ CODE	HIER2_ CODE	HIER3_ CODE	HIER4_ CODE	HIER5_ CODE	6 – 19	HIER20_ CODE
1	US Acct	A	B	D	H	H	H	H
2	US Acct	A	B	D	I	I	I	I
3	US Acct	A	B	E	E	E	E	E
4	US Acct	A	C	F	J	J	J	J
5	US Acct	A	C	F	K	K	K	K
6	US Acct	A	C	G	G	G	G	G

Configuring General Ledger Account Hierarchies Using General Ledger

Accounting Flexfield value sets definitions These configuration steps are required if you are deploying Oracle Financial Analytics,

Oracle Procurement and Spend Analytics, or Oracle Supply Chain and Order Management Analytics. This topic contains the following topics:

- Overview

- About Configuring the ELT Process for GL Accounting flexfields
- How To Set Up Hierarchies With General Ledger Accounting flexfields

Overview

Oracle EBS supports up to 30 segments in which to store accounting flexfields. Flexfields are flexible enough to support complex data configurations, for example:

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

An Example Data Configuration for a Chart of Accounts

A single company might have a US Chart of Account and an APAC Chart of Account with the following data configuration:

Table 33. Example Chart of Accounts

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

The example shows that in Chart Of Account 4256, Company is stored in the segment3 column in the Oracle EBS table GL_CODE_COMBINATIONS_ALL. In Chart Of Account COA4257, Company is stored in segment1 column in GL_CODE_COMBINATIONS_ALL table. The objective of this configuration file is to make sure that when segment information is extracted into the warehouse table W_GL_ACCOUNT_D, segments with the same nature from different chart of accounts restored in the same column in W_GL_ACCOUNT_D. For example, we can store Company segments from COA 4256 and 4257 in segment1 column in W_GL_ACCOUNT_D; and Cost Center segments from COA 4256 and 4257 in segment2 column in W_GL_ACCOUNT_D, and so forth.

About Configuring the ELT Process for GL Accounting flex fields

Configuration for SAP adaptor:

Before you run the ELT process for General Ledger Accounts, you need to specify the segments that you want to aggregate in W_GL_BALANCE_A. To specify the segments that you want to aggregate, use the following ELT configuration file:

\$ODI_HOME\biapps_odi\odifiles\odidatafiles\srcfiles\file_glacct_segment_config_sap.csv

Figure 115. Screen Shot of file_glacct_segment_config_sap.csv File Opened in Excel

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	CHART OF ACCOUNTS_ID	SEG1	SEG1_VALUESETID	SEG2	SEG2_VALUESETID	SEG3	SEG3_VALUESETID	SEG4	SEG4_VALUESETID	SEG5	SEG5_VALUESETID	SEG6	SEG6_VALUESETID	SEG7
2	AGGREGATION	Y		Y		Y		Y		Y		Y		

SAP data can be aggregated based on 6 segments, namely GL Account, Cost Center, Profit Center, Business Area, Segment for Segmental Reporting and Order. These 6 segments are populated in ACCOUNT_SEG1_CODE to ACCOUNT_SEG6_CODE of W_GL_ACCOUNT_D respectively. In the configuration file file_glacct_segment_config_sap.csv, you need to identify the segments from W_GL_ACCOUNT_D that need to be aggregated for W_GL_BALANCE_A and mark those segments as 'Y'.

Following section helps to understand the configuration of aggregation of segments in GL Balance:

In the given example shown in figure 7, If you want to aggregate the balance data on GL Account & Profit Center segments, the file_glacct_segment_config_sap.csv needs to be configured and SEG1 and SEG3 which represents the GL Account & Profit Center respectively in W_GL_ACCOUNT_D should be marked as 'Y' for AGGREGATION and rest are to be left blank.

Figure 116. Screen Shot of file_glacct_segment_config_sap.csv File Opened in Excel

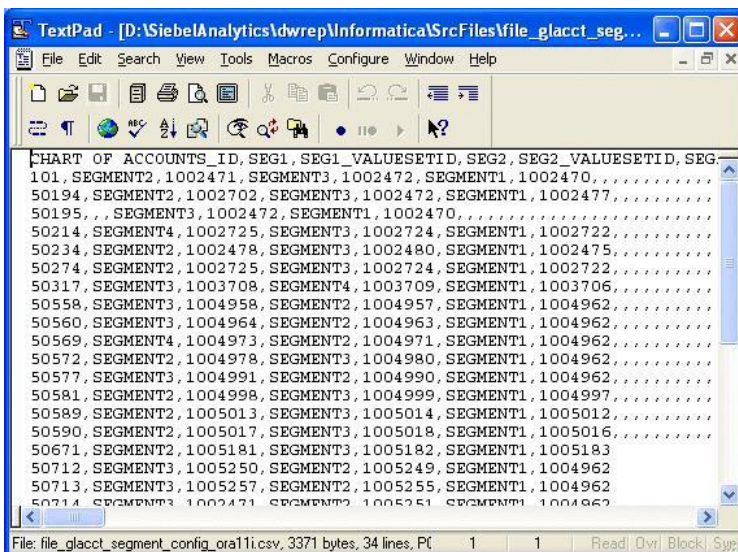
	A	B	C	D	E	F	G	H	I
1	CHART OF ACCOUNTS_ID	SEG1	SEG1_VALUESETID	SEG2	SEG2_VALUESETID	SEG3	SEG3_VALUESETID	SEG4	SEG4_VALUESETID
2	AGGREGATION	Y				Y			

Configuration for EBS:

Before you run the ELT process for General Ledger Accounts, you need to specify the segments that you want to analyze. To specify the segments that you want to analyze, use the following ELT configuration file:

<ODI_HOME>\biapps_odi\odifiles\odidatafiles\srcfiles\file_glacct_segment_config_ora.csv

Figure 117. Screen Shot of file_glacct_segment_config_ora.csv File Opened in a Text Editor



In the file_glacct_segment_config_ora.csv file, you need to specify the segments of the same type in the same column. For example, you might store all Cost Center segments from all chart of accounts in one

column, and all Company segments from all chart of accounts in another column. For example, you might want to do the following:

- Analyze GL account hierarchies using only Company, Cost Center, Natural Account, and LOB. You are not interested in using Geography for hierarchy analysis.
- Store all Cost Center segments from all COAs in ACCOUNT_SEG2_CODE column in W_GL_ACCOUNT_D.
- Store all Natural Account segments from all COAs in ACCOUNT_SEG3_CODE column in W_GL_ACCOUNT_D.
- Store all LOB segments from all COAs in ACCOUNT_SEG4_CODE column in W_GL_ACCOUNT_D.
- In W_GL_BALANCE_A (where you store GL account balances at aggregated level), you want to store GL account balances at Company and Cost Center level instead of at GL Code Combination level.

The screenshot below shows how the file_glacct_segment_config_ora.csv would be configured to implement the business requirements specified above.

Figure 118. 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	1002470	SEGMENT1	1002478
3	4257	SEGMENT1	1002702	SEGMENT2	1002472	SEGMENT3	1002477	SEGMENT4	1002479
4	AGGREGATION	Y	Y						

How To Set Up Hierarchies With General Ledger Accounting flexfields

This section is not Applicable for SAP

To set up hierarchies with General Ledger Accounting flexfields:

1. Use the <ODI_HOME>\biapps_odi\odifiles\odidatafiles\srcfiles\file_glacct_segment_configur_ora.csv file to specify the segments that you want to analyze.
2. Do the following:
 - a. Start Oracle BI Applications Configuration Manager (for more information, see "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
 - b. Select the Administer ELT Parameters link.
 - c. Display the Global tab.
 - d. Set the value of the LOAD_VALUESET_HIERARCHY parameter to 'Y'.
 - e. Set the value of the LOAD_FSG_HIERARCHY parameter is set to 'N'.
 - f. Save your changes

For more information about specifying parameter values, see "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

3. Run the ELT process for General Ledger Accounts.

Steps Required Before A Full Load

4. Using the Oracle BI Administration Tool, in the RPD Physical Layer, create additional aliases or change the names of the existing alias against the table W_HIERARCHY_D.

For example, if the existing aliases are numbered 1 to 3, name the new alias

Dim_W_HIERARCHY_D_ValueSetHierarchy4, and rename the existing aliases as described in the table below:

Table 34. Alias Names

Old alias name	New alias name
Dim_W_HIERARCHY_D_ValueSetHierarchy1	Dim_Company_Hierarchy_D
Dim_W_HIERARCHY_D_ValueSetHierarchy2	Dim_CostCenter_Hierarchy_D
Dim_W_HIERARCHY_D_ValueSetHierarchy3	Dim_NaturalAccount_Hierarchy_D

5. Create a new alias against W_HIERARCHY_D and name the new alias Dim_LOB_Hierarchy_D.
6. Using the Oracle BI Administration Tool, in the Physical Layer, create joins in the physical layer from the new aliases that you created in the previous step, as follows:
 - Company Hierarchy will join to the segment column in W_GL_ACCOUNT_D that stores the Company segment.
 - Dim_Company_Hierarchy_D.HIER20_CODE = W_GL_ACCOUNT_D.ACCOUNT_SEG1_CODE
 - Dim_Company_Hierarchy_D.HIER_CODE = W_GL_ACCOUNT_D.ACCOUNT_SEG1_ATTRIB
 - Cost Center Hierarchy will join to the segment column in W_GL_ACCOUNT_D that stores the Cost Center segment.
 - Dim_CostCenter_Hierarchy_D.HIER20_CODE = W_GL_ACCOUNT_D.ACCOUNT_SEG2_CODE
 - Dim_CostCenter_Hierarchy_D.HIER_CODE = W_GL_ACCOUNT_D.ACCOUNT_SEG2_ATTRIB
 - Natural Account Hierarchy will join to the segment column in W_GL_ACCOUNT_D, that stores the Natural Account segment.
 - Dim_NaturalAccount_Hierarchy_D.HIER20_CODE = W_GL_ACCOUNT_D.ACCOUNT_SEG3_CODE
 - Dim_NaturalAccount_Hierarchy_D.HIER_CODE = W_GL_ACCOUNT_D.ACCOUNT_SEG3_ATTRIB
 - LOB Hierarchy will join to the segment column in W_GL_ACCOUNT_D that stores the LOB segment.
 - Dim_LOB_Hierarchy_D.HIER20_CODE = W_GL_ACCOUNT_D.ACCOUNT_SEG4_CODE
 - Dim_LOB_Hierarchy_D.HIER_CODE = W_GL_ACCOUNT_D.ACCOUNT_SEG4_ATTRIB

Note: Hierarchies are linked to HIER20_CODE, because it is leaf node of the hierarchy

- Join the aliases of W_HIERARCHY_D to the corresponding columns in the table W_GL_BALANCE_A.
 - Dim_Company_Hierarchy_D.HIER20_CODE = Fact_Agg_W_GL_BALANCE_A.ACCOUNT_SEG1_CODE AND
 - Dim_Company_Hierarchy_D.HIER_CODE = Fact_Agg_W_GL_BALANCE_A.ACCOUNT_SEG1_ATTRIB

Note: W_GL_BALANCE_A has only 6 segment columns. So, if you have more than 6 hierarchies, join only the first 6 to W_GL_BALANCE_A but join all hierarchies to W_GL_ACCOUNT_D in the previous step.

7. A HIER_CODE filter should be specified in the Business Model Layer to restrain the output of logical table to be one Hierarchy only. You must set the HIER_CODE to be filtered by the Valueset IDs that are applicable for that particular hierarchy. The list of the valueset ids would be same as the valueset ids you configured in the csv file in the first set.

To specify a filter in the Business Model Layer to restrain the output of logical table, do the following:

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

```
"Oracle Data Warehouse"."Catalog"."dbo"." Dim_W_HIERARCHY_D_ValueSetHierarchy1"."HIER_CODE"
IN (<comma seperated valuesetids>)
```

8. A HIER_CODE filter should be specified in the Business Model Layer to restrain the output of the logical table to be one Hierarchy only. To specify a filter in the Business Model Layer to restrain the output of the logical table, do the following:

- a. Right click logical table Dim_W_Hierarchy_D_ValueSet1.
- b. Choose Properties.
- c. Choose the Source tab.
- d. Select 'Dim_W_Hierarchy_D_ValueSet1'.
- e. Click Edit.
- f. Choose the Content tab.
- g. Insert the following code into the 'use WHERE clause. . .' text box:

```
"Oracle Data
Warehouse"."Catalog"."dbo"."Dim_W_Hierarchy_D_ValueSet1"."HIER_CODE" =
<Value Set Hierarchy Id>Where <Value Set Hierarchy ID> is the Value Set
Hierarchy ID of the segment for which you are creating a hierarchy, for
example, 1001.
```

9. Using the Oracle BI Administration Tool, in the Business Model Layer of the Oracle BI Analytics Warehouse, create additional hierarchies using the dimensions that you created in the previous step. For more information about example hierarchies that are installed with Oracle Business Intelligence Applications, refer to the following hierarchies in the Oracle Business Analytics Warehouse:

- GL ValueSetHierarchy1
- GL ValueSetHierarchy2
- GL ValueSetHierarchy3

10. In the Business Model Layer, look for all the logical fact table that has logical join to the logical hierarchy table Dim - GL ValueSetHierarchy1. You will need to create a similar logical table between the new logical hierarchy dimensions that you created and these logical facts, as follows:

- a. Under each of the logical fact table, open the logical table sources and go to the content tab. Under aggregation content, select the Show unmapped checkbox. 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.

11. Using the Oracle BI Administration Tool, in the Presentation Layer of the Oracle BI Analytics Warehouse, drag the new hierarchies into the presentation folder.

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

Configuring General Ledger Account Hierarchies Using Financial Statement

Generator (FSG) Report Definition

These configuration steps are required if you are deploying Oracle Financial Analytics, Oracle Procurement and Spend Analytics, or Oracle Supply Chain and Order Management Analytics. This topic contains the following topics:

- Overview
- About Configuring the ELT Process for Oracle Financial Statement Generator Report
- How To Set Up Hierarchies With Financial Statement Generator Report Definition

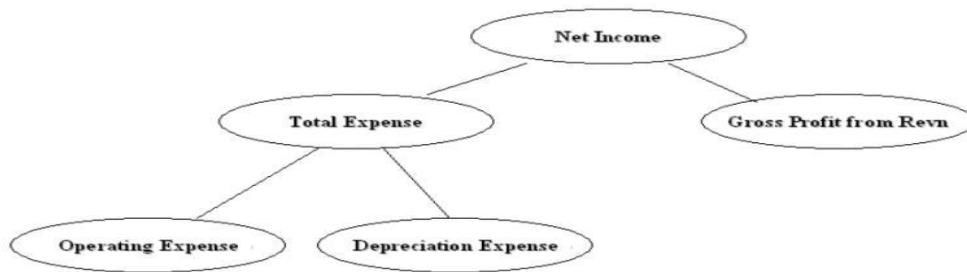
Overview

If you need to define GL account hierarchies based on multiple segments within a chart of accounts, you can use the Financial Statement Generator report definition UI in Oracle EBS to define them. You should first use the Oracle EBS Financial Statement Generator (FSG) form to define a row set or a column set, then Oracle Business Intelligence Applications will extract the row set or column set definition and convert them into hierarchies. Oracle Financial Statement Generator hierarchies are extracted from following EBS source tables:

- **RG_REPORT_AXIS_CONTENTS** - This table defines the relationship between the FSG report axis and GL code combinations. The GL code combinations with segment values within the value range defined for that axis are categorized as children of that axis.
- **RG_REPORT_AXIS_SETS** - This table stores the information for each of the row set or column set you defined. There is one record in this table for each row or column set you defined. Each row includes an axis set identifier, a row or column set name, and a structure identifier to assign a specific chart of accounts to the row set or column set.
- **RG_REPORT_CALCULATIONS** - This table stores formulas for calculating each row or column in the row or column set. An example of a row calculation might be to sum up the amount from the previous five rows. An example of a columns calculation might be to calculate column five by subtracting column four from column three. For example, in Income Statement, 'Net Income' is the calculation result of 'Gross Profit from Revenue' minus 'Total Expense'. When converting to hierarchy, Net Income becomes the parent of 'Gross Profit from Revenue' and 'Total Expense'. Therefore, hierarchy can be derived based on the information in RG_REPORT_CALCULATION.

The diagram below shows an example hierarchy, with the top level Net Income node having two child nodes, Total Expense, and Gross Profit from Revn, and the Total Expense node having two child nodes, Operating Expense, and Depreciation Expense. The diagram below shows how an income state is derived from a hierarchy.

Figure 119



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

Table 35. *Example flattened hierarchy stored in W_HIERARCHY_D*

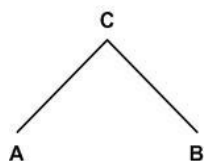
HIER Name	HIER1	HIER2	HIER3	HIER4	HIER20
Income Statement	Net Income	Gross Profit...	Gross Profit...	Gross Profit...	Gross Profit...
Income Statement	Net Income	Total Expenses	Operating Expenses	Operating Expenses	Operating Expenses
Income Statement	Net Income	Total Expenses	Depreciation Expense	Depreciation Expense	Depreciation Expense

Fact tables join to the W_HIERARCHY_D table via the General Ledger Account dimension table (W_GL_ACCOUNT_D).

The W_GL_ACCOUNT_D table contains six fields (HIER1_WID, HIER2_WID, HIER3_WID, ..., HIER6_WID), which are foreign keys to the W_HIERARCHY_D.row_wid. Therefore, each General Ledger Code combination can participate in up to six different hierarchies. You can decide which of the six hierarchies to drill on based on the column you use to join to W_HIERARCHY_D. For example, if you want to drill using the third hierarchy, you use W_GL_ACCOUNT_D.hier3_wid = W_HIERARCHY_D.row_wid.

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

Figure 120



About Configuring the ELT Process for Oracle Financial Statement Generator Report

This section is not Applicable for SAP

Before you run the ELT process for General Ledger Accounts, you need to specify the hierarchies that you want to reference. To specify the FSG hierarchies that you want to reference, use the following ELT configuration file:

<ODI_HOME>\biapps_odi\odifiles\odidatafiles\srcfiles\file_gl_hierarchy_assignment_ora.csv

Figure 121. Screen Shot of file_gl_hierarchy_assignment_ora.csv File Opened in a Spreadsheet Editor

	A	B	C	D	E	F
	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; Axis_set_id is a column from RG_REPORT_AXIS_SETS table. It is the unique ID of a row set or column set) you want to store in the GL account dimension table for the code combinations that belong to that chart of accounts.

The DATASOURCE_NUM_ID field specifies the data source to which the configurations apply. If you have multiple source systems, there might be a chart of accounts across the multiple source systems with the same ID. Therefore, you need to use the DATASOURCE_NUM_ID value to distinguish between them.

For example, we have an Income Statement FSG report and a Balance Sheet FSG report which we want to derive both of their hierarchy structures into OLAP. Oracle Business Intelligence Applications assumes that both reports are derived from same set of GL accounts with CHART_OF_ACCOUNTS=101. The axis_set_id of Income Statement is 1001, and for Balance Sheet, it is 1003. The DATASOURCE_NUM_ID for this application is 2. In addition, for those GL ACCOUNTS that belong to the two reports (by which we used to calculate the reports), we want to associate their HIER1 column (in GL_ACCOUNT_D) with Income Statement hierarchy structure, HIER3 column with Balance Sheet hierarchy structure.

In this case, we would add one row into file_gl_hierarchy_assignment_ora.csv with the following fields set as below:

CHART OF ACCOUNTS - 101

HIER1_AXIS_SET_ID - 1001

HIER3_AXIS_SET_ID - 1003

DATASOURCE_NUM_ID - 2

(Leave the other row values blank.)

This row indicates that for all of the GL Accounts with CHART_OF_ACCOUNTS=101 and DATASOURCE_NUM_ID=2, assigning hierarchies with axis_set_id=1001, null, 1003, null, null, null to HIER1~HIER6 columns respectively. Therefore, after extraction and loading, for those affected GL Account rows, HIER1 column will be the foreign key to Income Statement hierarchy's row ID in W_HIERARCHY_D and HIER3 column will be the foreign key to Balance Sheet hierarchy's row ID in W_HIERARCHY_D.

Note: Financial Analytics will not load those hierarchies with axis_set_id not been specified in file_gl_hierarchy_assignment_ora.csv.

How To Set Up Hierarchies With Financial Statement Generator Report Definition

To set up hierarchies with Financial Statement Generator Report Definition:

1. Use the <ODI_HOME>\biapps_odi\odifiles\odidatafiles\srcfiles\file_gl_hierarchy_assignment_ora.csv file to specify the hierarchies you want to reference for each CHART_OF_ACCOUNTS.
2. Do the following:
 - a. Start Oracle BI Applications Configuration Manager (for more information, see "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
 - b. Select the Administer ELT Parameters link.
 - c. Display the Global tab.
 - d. Set the value of the LOAD_VALUESET_HIERARCHY parameter to 'N'.
 - e. Set the value of the LOAD_FSG_HIERARCHY parameter is set to 'Y'.
 - f. Save your changes

For more information about specifying parameter values, see "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

3. Run the ELT process for General Ledger Accounts by doing one of the following:
 - If you have already created an Execution plan for General Ledger Accounts in Oracle BI Applications Configuration Manager, run this execution plan. For more information about how to create an Execution Plan, see "How to Create an Execution Plan In Oracle BI Applications Configuration Manager").
 - Use ODI Designer to perform an ELT process. for General Ledger Accounts. For more information about how to perform an ELT process in ODI Designer, see "How to perform E-LT Using ODI Designer").
4. Using the Oracle BI Administration Tool, in the RPD Physical Layer, create additional aliases or change the names of the existing alias against the table W_HIERARCHY_D.

For example, if you want to create an income statement hierarchy, create an additional alias Dim_IncomeStatement_FSGHierarchy_D against the table W_HIERARCHY_D.

5. Using the Oracle BI Administration Tool, in the Physical Layer of the Oracle BI Analytics Warehouse, create joins in the physical layer from the new aliases that you created in the previous step, as follows:

- a. Income Statement Hierarchy will join to one of the HIER1~6 columns that you have specified in file file_gl_hierarchy_assignment_ora.csv for Income Statement.
 - b. In this case, we join it to HIER1 column. Dim_W_GL_ACCOUNT_D.HIER1_WID = Dim_IncomeStatement_FSGHierarchy_D.ROW_WID
6. Using the Oracle BI Administration Tool, in the Business Model Layer of the Oracle BI Analytics Warehouse, create additional dimensions using the new alias. For the Income Statement hierarchy case, we create a new logical table Dim_IncomeStatement_FSGHierarchy_D, choose Dim_IncomeStatement_FSGHierarchy_D in physical layer as source. Mapping ROW_WID, HIER_CODE, and HIER1~HIER20 (both name and code) from physical table to logical key. Then, set HIER_CODE=1001 (this is the Axis_set_id of Income Statement hierarchy) in logical table to restrain the output of logical table to be Income Statement Hierarchy only (right click logical table Dim_IncomeStatement_FSGHierarchy_D - click properties - choose Source tab - select Dim_IncomeStatement_FSGHierarchy_D - click Edit button - choose Content tab - fill ("Oracle Data Warehouse"."Catalog"."dbo"."Dim_W_HIERARCHY_D_FSG1"."HIER_CODE" =1001) into 'use WHERE clause. . .' text box). For more information about this process, please refer to the following pre-installed example logical table Dim - FSG Hierarchy 1 in the Oracle Business Analytics Warehouse
7. Using the Oracle BI Administration Tool, in the Business Model Layer of the Oracle BI Analytics Warehouse, create new dimension based on the logical table that you created in the previous step. Please refer to 'FSG Hierarchy 1' as an example.
8. In the Business Model Layer, look for all the logical fact table that has logical join to the logical hierarchy table Dim - FSG Hierarchy1. You will need to create a similar logical join between the new logical hierarchy dimensions that you created and these logical facts, as follows:
 - a. Under each of the logical fact table, open the logical table sources and go to the content tab. Under aggregation content, select the Show unmapped checkbox. It will show all the hierarchies you created in the previous step. For each of these hierarchies, select the logical level to 'Detail'.
 - b. In the business model diagram, create a new Complex Join between each of the new logical hierarchy tables and each of the logical facts. In the join, make sure the cardinality is (0,1) on the dimension side and N on the fact side.
9. Using the Oracle BI Administration Tool, in the Presentation Layer of the Oracle BI Analytics Warehouse, drag the new hierarchies into the presentation folder.

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

Configuring Product Hierarchy, Master Inventory Organization, and Group Accounts

This section is not Applicable for SAP

This topic explains how to configure Product Hierarchy, Master Inventory Organization, and Group Accounts, and contains the following topics.

- How to Map SAP General Ledger Natural Accounts to Group Account Numbers
- How to do Group Account correction for Oracle Application
- Configuration of Product Hierarchy (except for GL, HR modules)
- Configure the Master Inventory Organization in Product dimension Extract for Oracle 11i adapter (Except for GL & HR Modules)
- How to Map SAP General Ledger Natural Accounts to Group Account Numbers

- How to do Group Account correction for Oracle Application

Configuration of Product Hierarchy (except for GL, HR modules)

This topic contains configuration points for product hierarchy in the Product dimension table and the Inventory Product dimension table.

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

As configured out-of-the-box, the Oracle Business Analytics Warehouse extracts product categories where the CATEGORY_SET_ID is 2 or 27. However, it is likely that the categories you extract from the source system are different from these prepackaged category sets.

To configure your product category sets, do the following:

- Identify the category sets you want to report (for more information, see "How to identify Category Sets from EBS").
- Configure the parameters for product hierarchy (for more information, see "Configure the ODI Parameters for Product Hierarchy").
- Configure the hierarchy mapping with the segments (for more information, see "Configure the Hierarchy Mapping With the Segments").

How to identify Category Sets from EBS

These steps are part of the task in this topic: Configuration of Product Hierarchy (except for GL, HR modules).

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

1. Log in to the EBS instance.
2. Navigate to Setup > Items > Categories > Default Category Sets.
3. Look for functional Area Inventory and place the cursor in the Category Set field.
4. Choose Help, then Diagnostics, then Examine, and specify the apps user password.
5. Click the Field LOV button and select CATEGORY_SET_ID, and note down the value.
6. Repeat steps 3-5 for the Purchasing functional area.

Configure the ODI Parameters for Product Hierarchy

These steps are part of the task in this topic: Configuration of Product Hierarchy (except for GL, HR modules).

To configure ODI Parameters for product hierarchy:

1. Start Oracle BI Applications Configuration Manager (for more information, see "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the Administer ELT Parameters link.
3. Display the Common tab.
4. Locate the following parameters and use the Parameter Value field to set the value:
 - INVPROD_CAT_SET_ID

Use the CATEGORY_SET_ID value that you returned in step 5 in the topic: How to identify Category Sets from EBS.

- PROD_CAT_SET_ID

Use the CATEGORY_SET_ID value that you returned in step 6 in the topic How to identify Category Sets from EBS.

5. Save your changes.

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

Configure the Hierarchy Mapping With the Segments

These steps are part of the task in topic: Configuration of Product Hierarchy (except for GL, HR modules.

To configure the hierarchy mapping with the segments: In ODI Designer, open the Conversion folder.

1. Select the folder 'SDE_ORA11510_Adaptor', then the 'Packages' folder
2. In the Packages folder locate the package 'SDE_ORA_ProductDimension', and edit the interfaces 'Run PRODUCT_DS' and 'Run PRODUCT_DS_FULL', to edit the mapping of the SEGMENTS to the HIER columns.
3. In the Target Datastore object, scroll down to find the hierarchy code ports named PROD_HIER<X>_CODE. Hierarchy levels are named with the convention PROD_HIER<X>_CODE, where <X> denotes the level within the hierarchy.

For example, if you want to edit the first level of your hierarchy, you must edit the definition for PROD_HIER1_CODE port.

The first level is pre-configured as follows:

```
PROD_HIER1_CODE = IIF(  
ISNULL(SQ_W_ORA_PRODUCT_DS_TMP.PROD_HIER1_CODE)  
OR ISNULL(SQ_W_ORA_PRODUCT_DS_TMP.PROD_HIER2_CODE), NULL,  
SQ_W_ORA_PRODUCT_DS_TMP.PROD_HIER1_CODE || '~' ||  
SQ_W_ORA_PRODUCT_DS_TMP.PROD_HIER2_CODE )
```

The second level is pre-configured as follows:

```
PROD_HIER2_CODE = SQ_W_ORA_PRODUCT_DS_TMP.PROD_HIER1_CODE
```

4. Click the Expression field for the appropriate PROD_HIER<X>_CODE port, then in the Expression Editor, modify the expression.
5. Apply and save your changes.

Note: The concatenation of the Segments at level 1 is done to make the CODE column unique with the assumption of the Segment2 is a Dependent Segment on the parent Segment1.

Configure the Master Inventory Organization in Product dimension Extract for Oracle 11i adapter (Except for GL & HR Modules)

In Oracle 11i applications, the products are defined in a Master Organization and then copied into the other Inventory Organizations for transactions. The Product dimension Extract mapping

'SDE_ORA_ProductDimension_Derive' has been enabled for configuration of this Master Organization based on the configuration in the OLTP. As installed out-of-the-box, the organization ID (that is set by the \$\$MASTER_ORGparameter) is set to 204. This organization ID 204 needs to be changed based on the individual implementation of OLTP in your deployment.

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

To set the Master Inventory Organization in Product dimension Extract:

1. Start Oracle BI Applications Configuration Manager (for more information, see "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the Administer ELT Parameters link.
3. Display the Common tab.
4. Locate the following parameter and use the Parameter Value field to set the value:
 - MASTER_ORG

For example, MASTER_ORG=204.

5. Save your changes.

How to Map SAP General Ledger Natural Accounts to Group Account Numbers

Note: It is critical that the General Ledger Account Numbers are mapped to the Group Account Numbers (or domain values) as the metrics in the General Ledger reporting layer uses these values. For a list of domain values for General Ledger Account Numbers, see Oracle Business Analytics Warehouse Data Model Reference.

A CSV file needs to be prepared for mapping the SAP General Ledger Natural Accounts to Oracle Group Account Numbers. Steps to prepare the CSV file depends on whether the SAP system Group Chart of Account has been activated or not. Check the field entry for 'KKTPL' in table 'T004' in SAP table for Group Chart of Account activation. If the field has a value then that particular value is Group Chart of Account in SAP system.

Scenario 1: Group Chart of Account activated.

Map the GL account range based on Group Chart of Account definition to Oracle Group Account number.

Steps:

1. Use transaction code 'OCCP' in SAP.
2. Find out the Group Account number in SAP which can be mapped to Oracle Group Account number based on description.
3. Find out all the GL account numbers which are assigned to that particular Group Account Number.
4. Find the minimum and maximum value from Step 3.
5. Take minimum value as 'From Account' and maximum value as 'ToAccount'.

Scenario 2: Group Chart of Account not activated

Based on GL Account group range and GL account description decide the range of GL account which can be Group together to assign a predefined Oracle Group Account Number.

Refer to SAP table, T077S for GL Account Group Range and SKA1 for GL account description.

1. The csv file, domainValues_GL_GRP_ACCOUNT contains the following information:
 - a. Chart of Account : Specify GL account belongs to which chart of Account in SAP
 - b. Account Group: Specify account group for GL account in SAP
 - c. From Account: Starting range point of GL Account
 - d. To Account: End range point of GL Account
 - e. Name: Description for GL account range
 - f. Group Account Number: Account number for consolidation in Oracle warehouse
 - g. FIN_STMT_ITEM_CODE: classification of balance sheet and P&L items based on facts created in SAP.
 - h. Sample CSV file looks like as follows:

Figure 122

	A	B	C	D	E	F	G
1	Chart of Account	Account Group	From Account	To Account	Name	Group Account No	FIN_STMT_ITEM_CODE
2	MMC	AS	3400000	3499999	ASSOCIATE/SUBSIDIARY COMPANIES	OTHER CA	GL_OTHER
3	MMC	BK	3000000	3399999	BANK ACCOUNTS	CASH	GL_OTHER
4	MMC	BO	3700000	3999999	OTHER BALANCE SHEET ACCOUNTS	OTHER CA	GL_OTHER

Note: These files will be client specific as Account group, Chart of Account and GL Account range will be different based on their Chart of Account design.

How to do Group Account correction for Oracle Application

Note: Refer to the topic: How to Map Oracle General Ledger Natural Accounts to Group Account Numbers, for general concepts about Group Account Number and Financial Statement Item Code.

When a user mistakenly maps a GL natural account number to an incorrect Group Account Number, incorrect accounting entries might be inserted into the fact table. For example, natural account 1210 is classified as belonging to 'AR' Group Account Number when it should be classified as having 'AP' Group Account Number. When this happens, ODI will get all the journal lines charging to account 100 and try to reconcile these journal lines against sub-ledger accounting records in the AR fact table (W_AR_XACT_F). Since these journal lines did not come from AR, ODI will not be able to find the corresponding matching sub-ledger accounting records for these journal lines. In this case, ODI will insert 'Manual' records into the AR fact table because it thinks that these GL journal lines are 'Manual' journal entries created directly in GL system charging against the AR accounts. This entire process is call GL Reconciliation process.

In order to revert these 'Manual' entries in the AR fact, you will need to utilize the 'Group Account Number Cleanup' program provided in Oracle Business Intelligence Applications. This program will revert the 'Manual' entries in the fact table (in this case, AR fact table); and then try to do the GL reconciliation process again. This time, ODI will try to look for the corresponding matching sub-ledger accounting records in the AP fact (W_AP_XACT_F); provided that you've re-assign the natural account 1210to the 'AP' group account number in the file_group_acct_codes_ora.csv file.

To do Group Account correction:

1. Open the file file_group_acct_codes_ora.csv with a text editor in the <ODI_HOME>\biapps_odi\odifiles\odidatafiles\srcfiles directory.
2. Change the mapping of the GL Natural account to the group account in the CSVfile. For example, before the change, the CSV file might have the following values:

CHART OF ACCOUNTS ID = 101

FROM ACCT = 1110

TO ACCT = 1110

GROUP_ACCT_NUM = CASH

After the change, if the account '1210' originally belonged to the 'AP' Group Acct

Num and after correcting the GL Natural account to the group account, the CSV

file would have the following values:

CHART OF ACCOUNTS ID = 101

FROM ACCT = 1210

TO ACCT = 1210

GROUP_ACCT_UM = AR

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

4. Select the Administer ELT Parameters link.

5. Display the Global tab.

6. Locate the following parameters and use the Parameter Value field to set the value:

- LOAD_CALCULATE_GL_BALANCE

Set this value to 'N'.

Note: If 'LOAD_CALCULATE_GL_BALANCE' is set to 'Y', then the value of LOAD_EXTRACT_GL_BALANCE cannot be set to 'Y'.

7. Save your changes.

8. Run the corresponding Group Account Cleanup execution plan named ORA11510 Financials - Group Account Number Clean Up R12.

Configuration Steps for Controlling Your Data Set

This topic contains additional configuration steps that apply to Oracle Business Intelligence Applications, and contains the following topics:

- Configuring the Configuring the Make-Buy Indicator
- How to Configure Data Source Num IDs
- Configuring the Names of Country Region, State Region, State, or Region
- Configuring the Configuring the Make-Buy Indicator

How to Configure Data Source Num IDs

DATASOURCE_NUM_ID is a system column in the data warehouse that uniquely identifies a data source category and indicates which source systems the data comes from.

To configure a DATASOURCE_NUM_ID value:

1. Start Oracle BI Applications Configuration Manager (for more information, see "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the Administer ELT Parameters link.
3. Display the Global tab.
4. Locate the following parameter and use the Parameter Value field to set the value:
 - WH_DATASOURCE_NUM_ID
5. Save your changes.

For more information about specifying parameter values, see "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

Configuring the Names of Country Region, State Region, State, or Region

For Oracle 11i, you can reconfigure the names of the Country Region, State Region, State, and Region. This configuration information applies only to plant, storage, and supplier locations. By default, the Region Name column (EXT_REGION_NAME) is populated using the same code value as the Region Code column (EXT_REGION_CODE).

To configure the names of Country Region, State Region, State, or Region:

1. In ODI Designer, open the folder SDE_ORA11510_adapter.
2. Select the Interface you want to edit.

The following is a list shows Interfaces that use the COUNTRY_REGION, COUNTRY_NAME, STATE_NAME and STATE_REGION columns:

- SDE_ORA_SupplierDimension.SUPPLIER_DS
 - SDE_ORA_SupplierDimension_Full.SUPPLIER_DS_Full
 - SDE_ORA_BusinessLocationDimension_Plant.BUSN_LOCATION_DS
 - SDE_ORA_BusinessLocationDimension_Plant_Full.BUSN_LOCATION_DS_Full
 - SDE_ORA_BusinessLocationDimension_StorageLocation.BUSN_LOCATION_DS
 - SDE_ORA_BusinessLocationDimension_StorageLocation_Full.BUSN_LOCATION_DS_Full
3. Make sure that the name columns do not map to any source columns, as follows:
 - a. Double-click the Interface to display the Interface: <Name> dialog.
 - b. Display the Diagram tab.
 - c. In the Target Datastore area, select the column name.
 - d. In the Mapping: <Name> pane below, display the Implementation tab and delete any text in this tab. Deleting the text in the Implementation tab enables the lookup to resolve the Names from the W_CODE_D table for the corresponding CODE.
 4. Save the changes.
 5. Regenerate the SCENARIOS for the Packages that were using the Interfaces that you edited.

Configuring the Configuring the Make-Buy Indicator

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

Your organization may require different indicator codes. If so, you can modify the indicator logic by reconfiguring the condition in the Interface mplt_SA_ORA_ProductDimension. For example, you may want your indicator code to be 0 for make, and 1 for buy.

To configure the Make-Buy Indicator:

1. In ODI Designer, open the folder SDE_ORA11510_adapter.
2. Select the Interface you want to edit.

The following is a list shows Interfaces that use the Make_Buy_Ind column:

- SDE_ORA_ProductDimension_Derive.ORA_PRODUCT_DS_TMP
- SDE_ORA_ProductDimension_Derive_Full.ORA_PRODUCT_DS_TMP_FULL

3. Edit the Make_Buy_Ind columns, as follows:
 - a. Double-click the Interface to display the Interface: <Name> dialog.
 - b. Display the Diagram tab.
 - c. In the Target Datastore area, select the column name.
 - d. In the Mapping: <Name> pane below, display the Implementation tab and modify the text in this tab.
4. Save the changes.
5. Regenerate the SCENARIOS for the Packages that were using the Interfaces that you edited.

Chapter 6: Configuring Oracle Financial Analytics for SAP

This chapter describes how to configure Oracle Financial Analytics for SAP R/3 release 4.6c and ECC 6.0 sources to meet your business needs, and contains the following topics:

- Overview of Oracle Financial Analytics for SAP
- Configuration Required Before a Full Load for Financial Analytics for SAP

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

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 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 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.

Configuration Required Before a Full Load for Financial Analytics for SAP

This topic contains configuration steps that you need to perform on Oracle Financial Analytics before you do a full data load, and contains the following topics:

- About Configuring Domain Values and CSV Worksheet Files for Oracle FinancialAnalytics" How to configure file_xact_category_AP_SAP.csv and file_xact_category_AR_SAP.csv for Oracle General Ledger and Profitability Analytics
- How to Specify the Ledger or Set of Books for which GL Data is Extracted"

The filter is not configurable for SAP extract. The warehouse GL data will have Leading & all Non Leading ledger data from SAP. For SAP 4.6C there is no concept of Non leading ledger.

About Configuring Domain Values and CSV Worksheet Files for Oracle FinancialAnalytics

Domain Values and CSV Worksheet Files for Oracle Financial Analytics and SAP ADAPTOR.

Figure 123

Worksheet File Name	Description	Package Name
domainvalues_GL_GRP_ACCOUNT_SAPECC6.0.csv	GL Number Range Mapping for Group Account Number	SDE_SAP_GLAAccountDimension and SDE_SAP_GLAAccountDimension_Commititm
domainvalues_GL_GRP_ACCOUNT_SAP4.6C.csv	GL Number Range Mapping for Group Account Number	SDE_SAP46C_GLAAccountDimension
domainvalues_STATUS_CATEGORY_CODE_D_SAP.csv	Status Of Document	SDE_SAP_CodeDimension_SubStatus and SDE_SAP_CodeDimension_Status
domainvalues_STATUS_D_ACCT_DOC_SAP.csv	Status Of Document	SDE_SAP_StatusDimension_AccountingDocument and SDE_SAP46C_StatusDimension_AccountingDocument
domainvalues_TransactionCategory_CODE_D_SAP.csv	Document Types	SDE_SAP_CodeDimension_TransactionCategory
domainvalues_XACT_TYPE_D_SAPECC6.0.csv	Various Transaction Type	SDE_SAP_TransactionTypeDimension_AcctDoc
domainvalues_XACT_TYPE_D_SAP4.6C.csv	Various Transaction Type	SDE_SAP46C_TransactionTypeDimension_AcctDoc
domainvalues_TAX_CODES_SAPECC6.0.csv	Tax Code Mapping	SDE_SAP46C_GLAAccountDimension and SDE_SAP_GLAAccountDimension
domainvalues_UOM_Conversion_SAPECC6.0.csv	Conversion of UOM	SDE_SAP_GLRevenueFact and SDE_SAP_GLCOGSFact
domainvalues_UOM_Conversion_SAP4.6C.csv	Conversion of UOM	SDE_SAP46C_GLCOGSFact and SDE_SAP46C_GLRevenueFact
		SDE_SAP46C_APTransactionFact_Open and SDE_SAP46C_ARTransactionFact_Cleared and SDE_SAP46C_APTransactionFact_Cleared and SDE_SAP46C_ARTransactionFact_Open and SDE_SAP_ARTransactionFact_Cleared and SDE_SAP_APTransactionFact_Open and SDE_SAP_APTransactionFact_Cleared and SDE_SAP_ARTransactionFact_Open
file_AWKEY_SAP.csv	Awkey Splitting Logic	
file_sap_group_acct_names_SAP.csv	Oracle Group Account Nuber Description	SDE_SAP_CodeDimension_Group_Acct_Names
file_xact_category_AP_SAP.csv	Posting Key Mapping for AP	SDE_SAP46C_APTransactionFact_Open and SDE_SAP46C_APTransactionFact_Cleared and SDE_SAP_APTransactionFact_Cleared and SDE_SAP_APTransactionFact_Open
file_xact_category_AR_SAP.csv	Posting Key Mapping for AR	SDE_SAP_ARTransactionFact_Cleared and SDE_SAP_ARTransactionFact_Open and SDE_SAP46C_ARTransactionFact_Cleared and SDE_SAP46C_ARTransactionFact_Open
file_glacct_segment_config_sap.csv	Aggregation of GL at Segment Level	SDE_SAP_Stage_GLAAccount_SegmentConfig_Extract
file_mcal_qtr_SAPECC6.0.csv	Fiscal Quarter Definition	SDE_SAP_Stage_McalPeriodLookup
file_mcal_qtr_SAP4.6C.csv	Fiscal Quarter Definition	SDE_SAP46C_Stage_McalPeriodLookup
domainvalues_EMP_D_ETHNICGRP_SAP.csv	Groups Employees by Ethnic Origin	SDE_SAP_EmployeeDimension and SDE_SAP46C_EmployeeDimension

DomainValues_XACT_TYPE_D_SAPECC6.0.csv & DomainValues_XACT_TYPE_D_SAP4.6c.csv

The purpose of this file is to categories the source into various activities to be grouped together.

W_XACT_Type_D defines various transaction types as they are defined in domain and client value fields.

The file has following fields:

- **CLIENT_ID:** A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.
- **XACT_CODE:** This identifies the source. From data warehouse point of view, sources have been defined into five categories : Receivables, Payables, Others, Revenue and COGS where Receivables means Customer, Payables means Vendors, Others include all business activities including receivables and payables, Revenue means income that a company receives from its normal business activities, usually from the sale of goods and services to customers and COGS means the inventory costs of those goods a business has sold during a particular period.
- **XACT_CAT_CODE:** It denotes that it is an accounting document. It is represented as 'ACC_DOC'
- **XACT_TYPE_CODE:** This is a combination of posting key, source, DR / Cr indicator, Special GL indicator and "A" which represents "Application". Use transaction code SE16 to get the values from the table. Posting key, Source, DR/CR indicator are available in table 'TBSL', TBSL.BSCHL, TBSL.SHKZG and TBSL.KOART respectively and Special GL indicator is available in table TSBLT, TSBLT.UMSKZ by applying filter "EN" on TBSL.SPRAS. The values corresponding to posting key has to be maintained. If No value is present corresponding to a posting key "" has to be maintained. Another line with same entries needs to be maintained with "A" at the last. This represents the Application transactions and is passed as hard coded value in mapping.

E.g.

- In case of receivables, 01~Customer~S~~. Here 01 is the posting key of Customers. S denotes a debit entry and there is no special GL indicator and A for Application is not maintained.
- In case of receivables, 01~Customer~S~A. Here 01 is the posting key of Customers. S denotes a debit entry and there is no special GL indicator and A for Application is maintained.
- In case of receivables, 09~Customer~S~1~. Here 09 is the posting key used for the purpose of special GL transaction of Customers. S denotes a debit entry and 1 is a special GL indicator which represents Down Payment and A for Application is not maintained.
- In case of receivables, 09~Customer~S~1~A. Here 09 is the posting key used for the purpose of special GL transaction of Customers. S denotes a debit entry and 1 is a special GL indicator which represents Down Payment and A for Application is maintained.

This applies in the case of receivables and payables.

- **XACT_TYPE_NAME :** It contains the description of the Xact Type code included above.
- **W_XACT_TYPE_CODE:** On the basis of business transaction, a category needs to be assigned to each Xact Type Code. Three Xact type code are used. "Original", "Payment" and "Application". "Original" is used for Invoices, "Payment" is used whenever there is stand alone payment usually applicable in Open line items for Customer & Vendors and "Application" in case of Payment or credit memo which is referred to an invoice in case of open line items and in case of cleared line items against invoice.
- **W_XACT_SUBTYPE_CODE:** A further sub type categorization of Xact type code which details the transaction. E.g. For Application, it provide more detail information as whether it is cash application or Credit Memo application.

Sample CSV looks as follows:

Figure 124

	A	B	C	D	E	F	G
1	CLIENT_ID	XACT_CODE	XACT_CAT_CODE	XACT_TYPE_CODE	XACT_TYPE_NAME	W_XACT_TYPE_CODE	W_XACT_SUBTYPE_CODE
2	800	RECEIVABLES	ACC_DOC	01~Customer~S~	Customer Invoice	ORIGINAL	INVOICE
3	800	RECEIVABLES	ACC_DOC	02~Customer~S~	Cust : Credit Memo Reversal	ORIGINAL	INVOICE
4	800	RECEIVABLES	ACC_DOC	03~Customer~S~	Bank Charges	ORIGINAL	OTHER
5	800	RECEIVABLES	ACC_DOC	04~Customer~S~	Customer : Other Receivables	ORIGINAL	OTHER

This CSV file contains information for Receivables, Payables and Others (General Ledger, Revenue, COGS, Assets, Tax, and Non-leading ledger information)

The structure of XACT_TYPE_CODE_D_SAPCC6.0.csv and XACT_TYPE_CODE_D_SAP4.6c.csv is 'Posting Key - "Customer" - Debit / Credit indicator' ~ Special GL Indicator ~ "A" for Application

Where S: Debit and H: Credit

SAP 4.6C File can be prepared in similar way.

file_AWKEY_SAP.CSV

Purpose of this CSV file is to derive the "Purchase Invoice Number" or "Sales Invoice Number".

This CSV file will be valid across all SAP Clients and Supported Version.

This CSV files contains following information:

AWKEY SPLIT LOGIC AND ITS DERIVATION

For all the invoices the reference number (AWKEY) is split based on the reference transaction code based on the below logic.

For Reference transaction codes from Finance Module the last four characters are Financial Year, The Previous four characters are company code & the rest is Invoice No (Finance Document No).

For Reference transaction codes from Material Management Module the last four characters are Financial Year & the rest is the Invoice No.

- For Reference transaction codes of SD module, only Invoice No is available in the AWKEY field.

Table 36. Example CSV File

Module	Document No	Company Code	Financial Year
FI	Y	Y	Y
MM	Y		Y
SD	Y		

Sample CSV looks as follows:

Figure 125

AWTYP	DOC_NO	C_CODE	F_YEAR
BKPF	1	4	4
RMRP	1	0	4
VBRK	1	0	0

File_FIN_STMT_ITEM_Name

This file has been prepared based on the Financial Statement Item code available in SAP and scope of adaptor for Finance module.

It contains Financial Statement Item code which determines the nature of accounts.

The following nature of accounts is maintained in SAP Financial module which is covered within the scope of the Oracle Financial Analytics adapter for SAP:

1. Revenue : Sales Revenue
2. COGS : Cost of Goods Sold
3. AP : Accounts Payable
4. AR : Accounts Receivable
5. Tax : Tax Payables
6. Others :Other Transaction
7. Assets : Assets

Sample CSV looks as follows:

Figure 126

	A	B	C
1	GROUP_ACCOUNT_NUM	GROUP_ACCOUNT_NAME	LANGUAGE
2	ACC DEPCN	ACC DEPCN	E
3	ACC LIAB	ACCRUED LIABILITIES	E
4	AP	ACCOUNTS PAYABLES	E
5	AR	ACCOUNTS RECEIVABLE	E
6	CASH	CASH	E
7	CMMN STOCK	CMMN STOCK	E
8	COGS	COST OF GOODS SOLD	E

Similarly file is prepared for 4.6c.

How to configure file_xact_category_AP_SAP.csv and file_xact_category_AR_SAP.csv for Oracle General Ledger and Profitability Analytics

File_Xact_Category_AP_SAP

1. Purpose : The objective of this file is to provide the posting keys used in various business transactions in SAP. This particular file has been prepared to include all the posting keys used for recordings business transactions with Vendor in SAP.
2. The file is divided into three parts
 - a. XACT_CODE : This represents the source. This CSV is for Vendors domainValues_XACT_TYPE_D_SAPECC6.0.csv
 - b. Posting Key : The structure of posting key is 'Posting Key ~ "VENDOR" ~ DR / CR indicator / ~ Special GL Indicator ~ Sub code of Special GL Indicator'
 - In SAP, there are separate posting keys defined for business transactions. Posting key 21 to 39 belongs to Vendor transaction
 - 'VENDOR' : This is hard coded
 - DR/ CR Indicator : 'S' represents for Debit and 'H' represents Credit

- Special GL transaction e.g. advance paid, Guarantee provided

Sample CSV looks as follows:

Figure 127

A	B	C
XACT_CODE	POSTING_KEY	CATEGORY
PAYABLES	21~Vendor~S~	PAYMENT
PAYABLES	22~Vendor~S~	PAYMENT
PAYABLES	24~Vendor~S~	PAYMENT
PAYABLES	25~Vendor~S~	PAYMENT
PAYABLES	26~Vendor~S~	PAYMENT

3. Business activities have been divided into two categories :
 - a. Invoice : All activities which results in increasing the liabilities
 - b. Payment : All activities which results in reduction of liability

File_Xact_Category_AR_SAP

1. Purpose : The objective of this file is to provide the posting keys used in various business transactions in SAP. This particular file has been prepared to include all the posting keys used for recordings business transactions with Customer in SAP.
2. The file is divided into three parts
 - a. XACT_CODE : This represents the source. This CSV is for Customers
 - b. Posting Key : The structure of posting key is 'Posting Key ~ "CUSTOMER" ~ DR / CR indicator ~ Special GL Indicator ~ Sub code of Special GL Indicator
 - In SAP, there are separate posting keys defined for business transactions. Posting key 11 to 19 belongs to Customer transaction
 - 'CUSTOMER' : This is hard coded
 - DR/ CR Indicator : 'S' represents for Debit and 'H' represents Credit
 - Special GL transaction e.g. advance paid, Guarantee provided
 - c. Sample CSV looks as follows:

Figure 128

XACT_CODE	XACT_TYPE_CODE	CATEGORY
RECEIVABLES	01~Customer~S~	INVOICE
RECEIVABLES	02~Customer~S~	INVOICE
RECEIVABLES	03~Customer~S~	INVOICE
RECEIVABLES	04~Customer~S~	INVOICE
RECEIVABLES	05~Customer~S~	INVOICE

3. Business activities have been divided into two categories :
 - a. Invoice : All activities which results into increasing the Assets / receivables
 - b. Payment : All activities which results into reduction of assets / Receivables

DomainValues_STATUS_D_ACCT_DOC_SAPECC6.0

This document is prepared to identify the status of a particular business activity. In SAP FI, there are below scenarios available:

1. Open and Cleared: In case of sub ledgers e.g. Customer / Vendors transaction documents have two conditions open and cleared. When any invoice is created, it is in Open status. Once the invoice is paid off, its status changes to cleared.

Add this value in column STATUS_CODE.

2. Posted : All other business transaction are called posted. Once a business transaction is entered into the system, it is in Posted status i.e. the existence of business document affects the ledger records of the business.

In SAP, there is no concept of 'Unposted'. A document created in FI having impact on total balance can always be in Posted status.

Within Posted status, it can be further sub divided as 'Open Posted' and 'Clear Posted'.

Add this value in column STATUS_CAT.

Sample CSV looks as follows:

Figure 129

STATUS_TYPE	STATUS_CODE	STATUS_NAME	STATUS_CAT	STATUS_CAT_DESC	SUBSTATUS_CODE	SUBSTATUS_NAME
ACCT_DOC	OPEN	OPEN DOCUMENT	POSTED	POSTED TO GL	OPEN~POSTED	OPEN DOCUMENT
ACCT_DOC	CLEARED	CLEARED DOCUMENT	POSTED	POSTED TO GL	CLEARED~POSTED	CLEARED DOCUMENT
ACCT_DOC	POSTED	POSTED DOCUMENT	POSTED	POSTED TO GL	POSTED	POSTED DOCUMENT

DomainValues_TAX_CODES_SAPECC6.0

This CSV files is used to identify different tax codes used in SAP for posting in a general ledger account.

In SAP FI the following SAP standard Tax Codes are available with description.

Figure 130 Tax Codes

Tax Code	Tax Code Description
-	Only input tax allowed
+	Only output tax allowed
*	All tax types allowed
<	Input Tax Account
>	Output Tax Account

The file contains the following information

- Tax Code - Type of tax to identify input or output tax
- Tax Code Description - description of each tax code.

Steps to Prepare CSV file.

Use transaction code FSS0, Enter a company code and the select GL Account. Point the cursor in Tax Category field and then press F4 key. All tax codes & their respective description will be displayed. We can then create the file based on the values displayed.

DomainValues_Transactioncategory_CODE_D_SAP

Purpose: This file is prepared to identify the document types and their status of a particular business transaction.

- XACT_CODE: This identifies the source. From data warehouse point of view, sources have been defined into five categories : Receivables, Payables, Others, Revenue and COGS where Receivables means Customer, Payables means Vendors, Others include all business activities including receivables and payables, Revenue means income that a company receives from its normal business activities, usually from the sale of goods and services to customers and COGS means the inventory costs of those goods a business has sold during a particular period.

- XACT_CAT_CODE: It denotes that it is an accounting document. It is represented as 'ACCT_DOC'
- XACT_TYPE_CODE: This represents the Document Type. Use transaction code SE16 to get the values from the table. Document Types are available in table 'T003', T003.BLART.
- XACT_TYPE_NAME: This represents the description. Use transaction code SE16 to get the values from the table. Document Types are available in table 'T003', T003.XMREF2.
- W_XACT_TYPE_CODE: The Values are defined as Original, Payment & Difference.
- W_XACT_SUBTYPE_CODE: The values are based on functionalities performed by the Document type like Return etc.

Sample CSV looks as follows:

Figure 131

	A	B	C	D	E	F
1	XACT_CODE	XACT_CAT_CODE	XACT_TYPE_CODE	XACT_TYPE_NAME	W_XACT_TYPE_CODE	W_XACT_SUBTYPE_CODE
2	OTHERS	ACCT_DOC	AA	Asset posting	ORIGINAL	OTHERS
3	OTHERS	ACCT_DOC	AB	Accounting document	ORIGINAL	OTHERS
4	OTHERS	ACCT_DOC	AE	Accounting document	ORIGINAL	OTHERS
5	OTHERS	ACCT_DOC	AF	Dep. postings	ORIGINAL	OTHERS
6	OTHERS	ACCT_DOC	AN	Net asset posting	ORIGINAL	OTHERS
7	RECEIVABLES	ACCT_DOC	DA	Customer document	ORIGINAL	OTHERS

file_glacct_segment_config-sap.csv

Purpose: The file is used in PLP for aggregation.

CHART_OF_ACCOUNTS_ID: It's used for aggregation in PLP thus aggregation is defined.

SEG1: "Y" represents that the segment is candidate for aggregation.

Figure 132

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
1	CHART OF ACCOUNTS_ID	SEG1	SEG1_VALUESETID	SEG2	SEG2_VALUESETID	SEG3	SEG3_VALUESETID	SEG4	SEG4_VALUESETID	SEG5	SEG5_VALUESETID	SEG6	SEG6_VALUESETID	SEG7	SEG7
2	AGGREGATION	Y		Y		Y		Y		Y		Y			
3															

file_sap_group_acct_names_SAP

Purpose: The file helps in identifying the Groups accounts in OBIA for balance sheet.

- CHART OF ACCOUNT: This is a list of all G/L accounts used by one or several company codes.
- ACCOUNT GROUP: In order to organize and manage a large number of G/L Accounts better, they are arranged in account group. The accounts of an account group normally have similar business functions. You could, for example have an account group for cash accounts, one for Expenses accounts, one for revenue accounts, and one for other balance sheet accounts etc.
- FROM ACCOUNT: The starting Number Range for G/L Accounts. Generally sufficient number range is provide to accommodate
 - Future G/L accounts, but if in case of different number ranges or number range being shared between Account groups the entries for each has to be maintained separately.
- TO ACCOUNT: The End of number range.
- NAME: The G/L Account Name

- **GROUP ACCOUNT NO:** This is same as G/L Account Name.
- **FIN_STMT_ITEM_CODE:** This identifies the source. From data warehouse point of view, sources have been defined into five categories : Receivables, Payables, Others, Revenue and COGS where Receivables means Customer, Payables means Vendors, Others include all business activities including receivables and payables, Revenue means income that a company receives from its normal business activities, usually from the sale of goods and services to customers and COGS means the inventory costs of those goods a business has sold during a particular period.

Steps to Prepare CSV file.

Use transaction code Se16, and enter table 'T077S', filed KTOPL stores value for Chart of accounts, KTOKS stores the value for Account group, VONNR & BISNR stores value for from account group & to account group and TXT30 store the value Group Account No.

Figure 133

	A	B	C	D	E	F	G
1	Chart of Account	Account Group	From Account	To Account	Name	Group Account No	FIN_STMT_ITEM_CODE
2	INT	AS	1000	1000	OTHER ASSET	OTHER ASSET	GL_OTHER
3	INT	AS	1010	1010	ACC DEPCN	ACC DEPCN	GL_OTHER
4	INT	AS	1050	1050	OTHER ASSET	OTHER ASSET	GL_OTHER
5	INT	AS	2000	2000	OTHER ASSET	OTHER ASSET	GL_OTHER
6	INT	AS	2010	2010	ACC DEPCN	ACC DEPCN	GL_OTHER
7	INT	AS	2050	2050	OTHER ASSET	OTHER ASSET	GL_OTHER
8	INT	ANL	2100	2110	OTHER ASSET	OTHER ASSET	GL_OTHER
9	INT	ANL	5000	5001	CASH	CASH	GL_OTHER

domain_values_UOM_Conversion_SAPECC6.0

Purpose: In SAP In the UNIT and commercial unit value varies in database. This file helps in representing the correct Unit based on Unit stored in SAP database.

- **CLIENT:** A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.
- **INT.MEAS.UNIT:** Internal, language-independent, measurement unit format. This format is used for internal processing.
 - All programs and tables use this representation internally. The internal format is automatically replaced by the external commercial (three-character) or technical (six-character) format, before output to the screen.
- **COMMERCIAL:** Commercial measurement unit format. The internal measurement units are automatically converted to this format for screen output. This format depends on the logon language.

Steps to Prepare CSV file.

Use transaction code Se16, and enter table 'T006A', field MSEHI stores value for Int.meas.unit, MSEH3 Commercial. Apply filter on Language.

Figure 134

	A	B	C
1	Client	Int. meas. unit	Commercial
2	800	%	%
3	800	%%	10
4	800	%O	%O
5	800	/MI	/MI
6	800	1	1
7	800	2	2
8	800	3	/NL
9	800	4	/PL

file_mcal_qtr_SAPECC6.0

Purpose: The file helps to define Quarter definition for the fiscal variant. The quarter start period and end period is defined.

- FISCAL YEAR VARIANT: The fiscal year variant is used to define the fiscal year. In the definition, you allocate your posting periods to the calendar year.
- YEAR: The fiscal year variant in SAP can be Year dependent (Calendar Year) or the fiscal year can start from different month of the year. Allocation of posting periods to calendar days must be made individually for each year.
- QTR: Definition value of the Quarter.
- QTR START PERIOD: The Start month of the Quarter.
- QTR END PERIOD: The end month of the Quarter.
- CLIENT ID: A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

Steps to Prepare CSV file.

Use transaction code Se16, and enter table 'T009', field PERIV stores value for Fiscal Variant. Check value for XKALE, if value = X then maintain "0" for the field Year in Excel. This denotes that the fiscal year variant is calendar Year thus the 1 day or month of the calendar will be always applicable. If value is <> X then check value XJABH the value shall be X. Enter table 'T009B'. This table have Year (T009B. BDATJ) defined for the fiscal variant and the 1 month (T009B.BUMON) of the year is equivalent to the period (T009B.POPER) of the fiscal year.

Based on above information the file can be defined.

Figure 135

	A	B	C	D	E	F
1	Fiscal Year Variant	Year	Qtr	Qtr Start Period	Qtr end period	Client Id
2	K4	0	1	1	3	800
3	K4	0	2	4	6	800
4	K4	0	3	7	9	800
5	K4	0	4	10	16	800
6	K0	0	1	1	3	800
7	K0	0	2	4	6	800
8	K0	0	3	7	9	800
9	K0	0	4	10	12	800

domainValues_EMP_D_ETHNICGRP_SAP

Purpose: Allows you to group employees by ethnic origin. It is used in countries which require this type of reporting by law.

- SAP_ETHNIC_GRP_DESC: Ethnic Origin.
- W_ETHNIC_GRP_CODE: Ethnic code defined in SAP.
- W_ETHNIC_GRP_DESC: Describes the Ethnic Origin.

Steps to Prepare CSV file.

Use transaction code Se16, and enter table 'T505S', field LTEXT stores value for Ethnic Description and RACKY stores the value of Ethnic code.

Figure 136

	A	B	C
1	SAP_ETHNIC_GRP_DESC	W_ETHNIC_GRP_CODE	W_ETHNIC_GRP_DESC
2	White		1 White (Not Hispanic or Latino)
3	Black		2 Black or African American (Not Hispanic or Latino)
4	Asian		3 Asian (Not Hispanic or Latino)
5	Asian (Not Hispanic or Latino)		3 Asian (Not Hispanic or Latino)
6	American Indian or Alaskan Native		4 American Indian or Alaskan Native (Not Hispanic or Latino)
7	Native Hawaiian or Other Pacific Islander		5 Native Hawaiian or Other Pacific Islander (Not Hispanic or Latino)
8	Hispanic/Latino		6 Hispanic or Latino
9	Not Hispanic/Latino		7 Two or more Races (Not Hispanic or Latino)
10			999 Not Specified

How to Specify the Ledger or Set of Books for which GL Data is Extracted

The filter is not configurable for SAP extract. The warehouse GL data will have Leading & all Non Leading ledger data from SAP. For SAP 4.6C there is no concept of Non leading ledger.

Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle Financial Analytics, and contains the following topics:

- How to Configure AP Balance ID for Payables Analytics
- How to Configure AR Balance ID for Oracle Receivables Analytics and Oracle General Ledger and Profitability Analytics
- How to Configure General Ledger and Profitability Analytics
- How to include UnApplied Payments in AR Aging Calculations

- How to Configure the AP/AR Aging Tables
- How to Configure How GL Balances Is Populated
- How to Setup Drill Down in Oracle BI Answers from General Ledger to Sub-ledger
- How to Configure Data Security for SAP R/3

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:

MANDT||~||BUKRS||~||KOKRS||~||HKONT||~||KOSTL||~||PRCTR||~||GSBER||~||SEGMENT
||~||AUFNR||~||RLDNR||~||WAERS||~||WAERS

Client ~ Company Code ~ Controlling Area ~ GL Account No ~ Cost Center ~ Profit Center ~ Business Area
~ Segement ~ Order Number ~ Ledger ID ~ Local Currency Key ~ Currency Key

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

To modify the Accounts Payable Balance ID:

Note: To modify the Accounts Payable Balance ID, you must modify the following packages:

- SDE_SAP_APTransactionFact_Open
 - SDE_SAP_APTransactionFact_Cleared
1. In ODI Designer, open the appropriate SAP Applications folder (SDE_SAPECC6_Adaptor).
 2. Open the folder SDE_SAP_APTransactionFact_Cleared
 3. Open the appropriate Package SDE_SAP_APTransactionFact_Cleared, and display the Diagram tab.
 4. Edit the Interface in the flow (for example, SDE_SAP_APTransactionFact_Cleared.W_SAP_AP_XACT_FS_TMP_CLEARED_ORIGINAL & SDE_SAP_APTransactionFact_Cleared.W_SAP_AP_XACT_FS_TMP_CLEARED_OFFSET).
 5. Display the Diagram tab.

On the right of the pane, you will see the 'Target Datastore' panel with the column definition of the target table (e.g. W_SAP_AP_XACT_FS_TMP).

6. Select BALANCE_ID, and change the expression in the text editor in the lower pane.
7. Save your changes.
8. The above changes to the interface will be applicable for incremental load as the same interface called in both Full & incremental load.
9. Re-generate the scenario of this package (expand the scenario folder under this package, right click the scenario and choose regenerate, keep all the parameters as default).
10. Repeat steps 1 to 8 for each of the packages listed above.

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 Accounts Payable (AR) Balance ID is maintained at the following granularity:

MANDT||'|BUKRS||'|KOKRS||'|HKONT||'|KOSTL||'|PRCTR||'|GSBER||'|SEGMENT
||'|AUFNR||'|RLDNR||'|WAERS||'|WAERS

Client ~ Company Code ~ Controlling Area ~ GL Account No ~ Cost Center ~ Profit Center ~ Business Area
~ Segment ~ Order Number ~ Ledger ID ~ Local Currency Key ~ Currency Key

However, if you want to maintain your AR balance at a different grain, you can redefine the Balance ID value in the applicable packages.

To modify the Accounts Receivables Balance ID:

Note: To modify the Accounts Receivables Balance ID, you must modify the following packages:

SDE_SAP_ARTransactionFact_Open

- SDE_SAP_ARTransactionFact_Cleared
 - In ODI Designer, open the appropriate SAP Applications folder (SDE_SAPECC6_Adaptor).
1. Open the folder SDE_SAP_ARTransactionFact_Cleared
 2. Open the appropriate Package SDE_SAP_ARTransactionFact_Cleared, and display the Diagram tab.
 3. Edit the Interface in the flow (for example, SDE_SAP_ARTransactionFact_Cleared_Full.AR_XACT_FS_TMP_ORG & SDE_SAP_APTransactionFact_Cleared.SDE_SAP_ARTransactionFact_Cleared).
 4. Display the Diagram tab.
 5. On the right of the pane, you will see the 'Target Datastore' panel with the column definition of the target table (e.g. W_SAP_AR_XACT_FS_TMP).
 6. Select BALANCE_ID, and change the expression in the text editor in the lower pane.
 7. Save your changes.
 8. The above changes to the interface will be applicable for incremental load as the same interface called in both Full & incremental load.
 9. Re-generate the scenario of this package (expand the scenario folder under this package, right click the scenario and choose regenerate, keep all the parameters as default).

Repeat steps 1 to 9 for each of the packages listed above.

How to include UnApplied Payments in AR Aging Calculations

To include UnApplied Payments in Aging Calculations for AR Aging tables

1. In ODI Designer, open the package PLP_ARSnapshotInvoiceAging in the PLP folder.
2. Open the Interface prep SQ_IA_AR_XACTS.
3. Display the Diagram tab and locate the table W_XACT_TYPE_D.
4. Selects the filter on the W_XACT_SUBTYPE_CODE column.
5. Change the SQL to the following:

```
W_XACT_TYPE_D.W_XACT_SUBTYPE_CODE IN ('INVOICE','CR MEMO','DR MEMO',  
'PAYMENT')
```

6. Save the changes.
7. Regenerate the scenario of this package (expand the scenario folder under this package, right click the scenario and choose regenerate, keeping all the parameters as default). The next E-LT will populate the Aging tables using UnApplied payments in the calculations.

How to Configure the AP/AR Aging Tables

This topic explains how to control the lengths of the aging buckets in the AP and AR aging snapshot tables. These tables are:

- W_AP_AGING_INVOICE_A
- W_AR_AGING_INVOICE_A
- W_AP_AGING_SUPPLIER_A
- W_AR_AGING_CUSTOMER_A

In these four tables, outstanding AP/AR balance information is broken out into rows. Each row represents the outstanding balance information for a particular aging bucket. Four aging buckets are provided out-of-the-box, with the following durations:

- Bucket 1: 0 - 30 days
- Bucket 2: 31 - 60 days
- Bucket 3: 61 - 90 days
- Bucket 4: 90+ days

To configure the length of aging buckets:

1. Start Oracle BI Applications Configuration Manager (for more information, see "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the Administer ELT Parameters link.
3. Display the Application Specific tab, and select Financial Analytics from the Select BI Application field.
4. Locate the following parameters and use the Parameter Value field to set the following values:
 - BUCKET1_START and BUCKET1_END
 - BUCKET2_START and BUCKET2_END
 - BUCKET3_START and BUCKET3_END
 - BUCKET4_START and BUCKET4_END
5. Save your changes.

For more information about specifying parameter values, see

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

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

How to Configure How GL Balances Is Populated

To populate the GL balances (stored in the W_GL_BALANCE_F table), you can either extract them directly from ERP General Ledger, or calculate them based on the records in the W_GL_OTHER_F table (which stores all journal lines). To configure how GL balances is populated:

1. Start Oracle BI Applications Configuration Manager (for more information, see "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the Administer ELT Parameters link.
3. Display the Global tab.
4. Locate the following parameters and use the Parameter Value field to set the values:

To extract the GL balances directly from ERP General Ledger:

- Set 'LOAD_EXTRACT_GL_BALANCE' to Y.
- Set 'LOAD_CALCULATE_GL_BALANCE' to N.

To calculate GL balances based on the records in the W_GL_OTHER_F table:

- Set 'LOAD_EXTRACT_GL_BALANCE' to N.
- Set 'LOAD_CALCULATE_GL_BALANCE' to Y.

5. Save your changes.

For more information about specifying parameter values, see "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

How to Setup Drill Down in Oracle BI Answers from General Ledger to Sub-ledger

Oracle Business Intelligence Applications enables you to trace a GL Journal to the Sub-ledger transaction that created that journal. This ability (using drill down) is achieved through the Navigation feature in Oracle BI Answers. This feature is available for AP if the source is Oracle EBS 11i. To set up drill down in Oracle BI Answers from General Ledger to Sub-ledger:

1. Create your sub-ledger request from 'Financials - AP Transactions' or 'Financials - AR Transactions' catalog as applicable.
2. In your request, add a filter on the column 'GL Journal ID' under the 'Document Details' folder and the set the operator of the filter to 'Is Prompted'.
3. Build your GL Journal request from the 'Financials - GL Detail Transactions' catalog.
4. To your request, add the column 'GL Journal ID' under the 'Document Details' folder.
5. Navigate to the Column Properties of this column, and set the Value Interaction property in the Column Format tab to 'Navigate'.
6. Add a navigation target and set the target location to the sub-ledger request you created earlier.

How to Configure Data Security for SAP R/3

Please refer security guide for this section.

Chapter 7: Configuring Oracle Procurement and Spend Analytics for SAP

This chapter describes how to configure Oracle Procurement and Spend Analytics for particular sources to meet your business needs, and contains the following topics:

- Section 7.1, "Overview of Oracle Procurement and Spend Analytics"
- Section 7.2, "Configuration Required Before A Full Load for Oracle Procurement and Spend Analytics"

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

Overview of Oracle Procurement and Spend Analytics for SAP

Oracle Procurement and Spend Analytics comprises the following:

- Oracle Procurement and Spend Analytics
(for more information, see Section , "Oracle Procurement and Spend Analytics Module").
- Oracle Supplier Performance Analytics
(for more information, see Section , "Supplier Performance Analytics Module").

Oracle Procurement and Spend Analytics enable organizations to optimize their supply chain performance by integrating data from across the enterprise supply chain and enabling executives, managers, and frontline employees to make more informed and actionable decisions. Organizations using Oracle Procurement and Spend Analytics benefit from increased visibility into the complete Procurement and Spend process, including comprehensive supplier performance analysis and supplier payables analysis. Through complete end-to-end insight into the factors that impact Procurement and Spend performance, organizations can significantly reduce costs, enhance profitability, increase customer satisfaction, and gain competitive advantage. Oracle Procurement and Spend Analytics also integrate with the other applications in Oracle Business Intelligence Applications product line. They deliver this insight across the organization to increase the company's effectiveness in managing its customers, suppliers, and financial decisions.

Oracle Procurement and Spend Analytics Module

Provides complete visibility into direct and indirect spend across the enterprise, payment, and employee expenses. Example analyses are spend by Commodity & Supplier, by Purchase Org, Cost Center, and expense by Employee, Buyer, etc.

The Oracle Procurement and Spend Analytics application is comprised of these subject areas:

- Total Spend: This is a summary subject area that provides the ability to do comparative analysis and report on requested spend, committed spend and actual spend across suppliers, company, products, commodities and associated hierarchies for both direct and indirect spend (indirect spend being MRO and employee expenses) in detail to allow complete visibility of spending across your organization.
- Purchase Orders: This is a detailed subject area that provides the ability to report on committed spend, and Purchase orders of the suppliers of an organization across suppliers, company, products, commodities and associated hierarchies at purchase order line level.

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

Supplier Performance Analytics Module

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

The Supplier Performance Analytics application is comprised of these subject areas:

- **Supplier Performance.** The Suppliers functional area contains targeted reports and metrics that allow you to analyze the timeliness, reliability, cost, and quality of goods provided by your suppliers. It helps you to understand how well suppliers are contributing to success of your organization, and to evaluate the price, quality, and delivery timing in procuring materials
- **Supplier AP Transactions:** This is a summary subject area that provides the ability to analyze payment performance and payment due analysis of the suppliers of an organization across suppliers, company, location, products, commodities and associated hierarchies. In addition to monitoring supplier performance, it is important to monitor organization's performance of making on time payments. This will help the Organizations to maintain better relationships with their best suppliers.

Configuration Required Before A Full Load for Oracle Procurement and Spend Analytics

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

- Section "How to Configure the Parameter for Purchase Cycle Line"
- Section "Domain Values and CSV Worksheet Files for Oracle Procurement and Spend Analytics"
- Section "Configuration Steps for Controlling Your Data Set"

How to Configure the Parameter for Purchase Cycle Line

To load the purchase cycle line table (W_PURCH_CYCLE_LINE_F), the ELT tasks need to distinguish data that originates in Oracle 11i applications.

To configure the parameter for Purchase Cycle Line:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the Administer ELT Parameters link.
3. Display the Application Specific tab, and select Supply Chain Analytics from the Select BI Application drop down list.
4. Locate the following parameters and use the Parameter Value field to set the value:

- ORA_DATASOURCE_NUM_ID_LIST

Change the values of parameter ORA_DATASOURCE_NUM_ID_LIST from 4 (the default value) to the list of Data Source NUM IDs that you defined for your Oracle data sources.

- SAP_DATASOURCE_NUM_ID_LIST

Change the values of parameter SAP_DATASOURCE_NUM_ID_LIST from (50,51) (the default value) to the list of Data Source NUM IDs that you defined for your SAP data sources.

- SIL_PURCHASECYCLELINESFACT
- SIL_PURCHASECYCLELINESFACT_EXTRACT

5. Save your changes.

For more information about specifying parameter values, see Section 4.6.3.3.1, "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

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

Table 38. *Domain Values and CSV Worksheet Files for SAP Procurement and Spend Analytics*

Worksheet File Name	Description	Session
domainValues_Status_Purch_Approve_SAP_ECC6.0.csv	Lists the Purchasing Approval Status column and the corresponding domain values Within SAP R/3 release 4.6c and ECC 6.0 source system. For information about how to edit this file, see Section "To configure domainValues_Status_Purch_Approve_SAP.csv".	SDE_SAP_StatusDimension_PurchaseApprove

domainValues_Status_Purch_Cycle_SAP.csv	Lists the Purchasing Cycle Status column and the corresponding domain values Within SAP R/3 release 4.6c and ECC 6.0 source system. For information about how to edit this file, see Section , "To configure domainValues_Status_Purch_Cycle_SAP.csv".	SDE_SAP_StatusDimension_PurchaseCycle
domainValues_Xact_Types_PO_Line_Type_SAP.csv	Lists the Purchasing Line Type and the corresponding domain Values Within SAP R/3 release 4.6c and ECC 6.0 source system. For information about how to edit this file, see Section "To configure domainValues_Xact_Types_PO_Line_Type_SAP.csv".	SDE_SAP_TransactionTypeDimension_PO_Line_Type
domainValues_Xact_Types_Purch_Orders_SAP.csv	Lists the Purchase Order Transaction Type column and the corresponding domain values Within SAP R/3 release 4.6c and ECC 6.0 source system. For information about how to edit this file, see Section "To configure domainValues_Xact_Types_Purch_Orders_SAP.csv".	SDE_SAP_TransactionTypeDimension_PurchaseOrder
domainValues_Xact_Types_Purch_Receipt_SAP.csv	Lists the Purchase Receipt Transaction Type column and the corresponding domain Within SAP R/3 release 4.6c and ECC 6.0 source system. For information about how to edit this file, see Section "To configure domainValues_Xact_Types_Purch_Requisitions_SAP.csv".	SDE_SAP_TransactionTypeDimension_PurchRcpt
domainValues_Xact_Types_Purch_Requisitions_SAP.csv	Lists the Purchase Requisition Transaction Type column and the corresponding domain Within SAP R/3 release 4.6c and ECC 6.0 source system. For information about how to edit this file, see Section, "To configure domainValues_Xact_Types_Purch_Requisitions_SAP.csv".	SDE_SAP_TransactionTypeDimension_PurchaseRequest

domainValues_status_Fulfillment SAP.csv	Lists the Fulfillment status of Purchase Requisition column and the corresponding domain Within R/3 4.6 C and ERP Central Component (ECC) 6.0 source system. For information about how to edit this file, see Section "To configure domainValues_Xact_Types_Purch_Requisitions_SAP.csv"	SDE_SAP_StatusDimension_Fulfillment
domainValues_Status_AP_Invoice_Payment_SAP.csv	Lists the status of AP payment column and the corresponding domain Within R/3 4.6 C and ERP Central Component (ECC) 6.0 source system. For information about how to edit this file, see Section "To configure domainValues_Status_AP_Invoice_Payment_SAP.csv"	SDE_SAP_StatusDimension_AP_Payment
domainValues_Xact_Types_AgrLevType_SAP.csv	Lists the Agreement leverage type of purchase order column and the corresponding domain Within R/3 4.6 C and ERP Central Component (ECC) 6.0 source system. For information about how to edit this file, see Section "To configure domainValues_Xact_Types_AgrLevType_SAP.csv"	SDE_SAP_TransactionTypeDimension_AgrLevType
domainValues_Xact_Types_ConsignedType_SAP.csv	Lists the consignment type of purchase order column and the corresponding domain Within R/3 4.6 C and ERP Central Component (ECC) 6.0 source system. For information about how to edit this file, see Section "To configure domainValues_Xact_Types_ConsignedType_SAP.csv"	SDE_SAP_TransactionTypeDimension_Consigned
domainValues_Xact_Types_Shipment_Type_SAP.csv	Lists the shipment type of Purchase order column and the corresponding domain Within R/3 4.6 C and ERP Central Component (ECC) 6.0 source system. For information about how to edit this file, see Section "To configure domainValues_Xact_Types_Shipment_Type_SAP.csv"	SDE_SAP_TransactionTypeDimension_ShipmentTypeeld

To configure file_APIInvoice_Spend_Type_SAP_ECC6.0.csv	Lists the AP Invoice Spend Type for Accounts Payables Invoice column and the corresponding domain Within R/3 4.6 C and ERP Central Component (ECC) 6.0 source system. For information about how to edit this file, see Section "To configure file_APIInvoice_Spend_Type_SAP_ECC6.0.csv"	SDE_SAP_CodeDimension_APIInvoice_Spend_Type
To configure file_Movement_SAP_EC C6.0.csv	Lists the Movements in SAP for Material Movements column and the corresponding domain Within SAP R/3 release 4.6c and ECC 6.0 source system. For information about how to edit this file, see Section "To configure file_Movement_SAP_ECC6.0.csv"	SDE_SAP_ProductXactFact,SDE_SAP_StageMatRqstnRcptQtyExtract,SDE_SAP_PurchaseReceiptFact
To configure file_PO_REJECT_SAP.csv	Lists the Release Indicators in SAP for Purchase Order column and the corresponding domain Within SAP R/3 release 4.6c and ECC 6.0 source system. For information about how to edit this file, see Section "To configure file_PO_REJECT_SAP.csv"	SDE_SAP46C_PurchaseCostFact,SDE_SAP_StagePurchaseScheduleExtract,SDE_SAP_PurchaseCostFact,SDE_SAP_PurchaseOrderFact,SDE_SAP_StagePurchaseContractExtract
Code_D_Xact_Types_Shipment_Type_SAP.csv	Lists the shipment type of Purchase order column and the corresponding Master Value Within Oracle Business Analytics Warehouse. For information about how to edit this file, see Section "Code_D_Xact_Types_Shipment_Type_SAP.csv"	SDE_SAP_CodeDimension_Xact_Types_Shipment_Type
To configure Code_D_Xact_Types_Purch_Requisitions_SAP.csv	Lists the shipment type of Purchase order column and the corresponding Master Value Within Oracle Business Analytics Warehouse. For information about how to edit this file, see Section "Code_D_Xact_Types_Shipment_Type_SAP.csv"	SDE_SAP_CodeDimension_Xact_Types_Purch_Requisitions

To configure Code_D_Xact_Types_Purch_Receipt_SAP.csv	Lists the Purchase Receipt Transaction Type column and the corresponding Master Value Within Oracle Business Analytics Warehouse. For information about how to edit this file, see Section "To configure Code_D_Xact_Types_Purch_Receipt_SAP.csv".	SDE_SAP_CodeDimension_Xact_Types_Purch_Receipt
To configure Code_D_Xact_Types_Purch_Orders_SAP.csv	Lists the Purchase Order Transaction Type column and the corresponding Master Value Within Oracle Business Analytics Warehouse. For information about how to edit this file, see Section "To configure Code_D_Xact_Types_Purch_Orders_SAP.csv".	SDE_SAP_CodeDimension_Xact_Types_Purch_Orders
To configure Code_D_Xact_Types_PO_Line_Type_SAP.csv	Lists the Purchasing Line Type and the corresponding Master Value Within Oracle Business Analytics Warehouse. For information about how to edit this file, see Section "To configure Code_D_Xact_Types_PO_Line_Type_SAP.csv".	SDE_SAP_CodeDimension_Xact_Types_PO_Line_Type
To configure Code_D_Xact_Types_ConsignedType_SAP.csv	Lists the Consigned Type and the corresponding Master Value Within Oracle Business Analytics Warehouse. For information about how to edit this file, see Section "To configure Code_D_Xact_Types_ConsignedType_SAP.csv".	
To configure domainValues_Xact_Types_AgrLevType_SAP.csv	Lists the Agreement leverage type of purchase order column and the corresponding Master Value Within Oracle Business Analytics Warehouse. For information about how to edit this file, see Section "To configure domainValues_Xact_Types_AgrLevType_SAP.csv"	SDE_SAP_CodeDimension_Xact_Types_AgrLevType

To configure Code_D_Status_AP_Invoice_Payment_SAP.csv	Lists the status of AP payment column and the corresponding Master Value Within Oracle Business Analytics Warehouse. For information about how to edit this file, see Section "To configure Code_D_Status_AP_Invoice_Payment_SAP.csv"	SDE_SAP_CodeDimension_Status_AP_Invoice_Payment
To configure Code_D_APInvoice_Spend_Type_SAP.csv	Lists the AP Invoice Spend Type for Accounts Payables Invoice column and the corresponding corresponding Master Value Within Oracle Business Analytics Warehouse. For information about how to edit this file, see Section "To configure Code_D_APInvoice_Spend_Type_SAP.csv"	SDE_SAP_CodeDimension_APInvoice_Spend_Type

To configure domainValues_Status_Purch_Approve_SAP_ECC6.0.csv and To configure domainValues_Status_Purch_Approve_SAP_4.6c.csv

The file has Approve status for Purchase requisition, Purchase order and Invoice

In SAP the release strategy is configured for Purchase requisition and purchase order.

1. CLIENT:

A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

2. STATUS CODE :

Represents the values in SAP Source. It is defined as a combination of Client~ Purchasing Document Category ~ Release Indicator. For invoice this is defined as Client~Approved.

Purchasing Document Category: differentiates the data within SAP between the various Business Objects like Purchase Requisition (B), Purchase Order (F) etc. Use Se16n Transaction and enter table EKPO (Purchasing Document Item). Find field BSTYP(Doc Category) and double click on field name BSTYP. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain- BSTYP. You will be in Dictionary: Display Domain Screen. Move to Value range tab. All the possible Purchasing Document Categories are displayed. Select the Purchasing Document Category.

Release Indicator: Indicates the processing phase the requisition is currently going through (e.g. In process of release).

In Case of Purchase Requisition use SE16n transaction and enter table T161S (Release Indicator: Purchasing Document). Select all value for field FRGKE(Release Indicator). If the corresponding value in field FRANG or FRBST is "X" then the Release indicator donates that the Requisition is released. The

Purchase Requisition which is not part of the approval (release strategy) as assumed to be released to the vendor and is donated with an entry “SELF” in place of release indicator.

In Case of Purchase Order use SE16n transaction and enter table T16FB (Release Indicator: Purchasing Document). Select all value for field FRGKE(Release Indicator). If the corresponding value in field KZFRE is “X” then the Release indicator donates that the Order is released. The purchase order which are not part of the approval (release strategy) as assumed to be released to the vendor and are donated with an entry “SELF” in place of release indicator.

In Case of Invoice since there is no Release strategy the posted invoices are assumed to be approved as are defined as approved.

3. STATUS_NAME & STATUS_DESC.

In Case of Purchase requisition T161S.FKZTX values can be maintained in the file as STATUS NAME & STATUS DESC values can be maintained in the file as STATUS NAME & STATUS_DESC for the corresponding FRGKE(Release Indicator). In case of Purchase Order use SE16n transaction and enter table T16FE(Descriptions of Release Indicators: Purchasing Documents). Provide language filter in SPRAS = “EN” and execute. Select all value for field FRGET(Description) for the corresponding FRGKE(Release Indicator). In Case of Invoice it is defined as Approved.

4. W_STATUS_CODE & W_STATUS_DESC:

On the basis of Approval Status of the business object, a status code needs to be assigned to each STATUS CODE. Four W_STATUS_CODE are used. “APPROVED”, “PENDING APPROVAL”, “REJECTED”, “OTHERS”.

- c. “APPROVED” is used when the business object is released,
- d. “PENDING APPROVAL” is used when the business object is still under release,
- e. REJECTED” is used when the business object is rejected and
- f. OTHERS” is used for other than the three categories are not applicable.

Figure 137.

	A	B	C	D	E	F	G
1	CLIENT	W_STATUS_CLASS	STATUS_CODE	STATUS_NAME	STATUS_DESC	W_STATUS_CODE	W_STATUS_DESC
2	800	PURCH_APPROVE	800~B~1	Request for Quotaion	Request for Quotaion	APPROVED	APPROVED
3	800	PURCH_APPROVE	800~B~2	RFQ/Purchase Order	RFQ/Purchase Order	APPROVED	APPROVED
4	800	PURCH_APPROVE	800~B~3	RFQ/PO no Change of date	RFQ/PO no Change of date	APPROVED	APPROVED
5	800	PURCH_APPROVE	800~B~4	RFQ/PO no changes	RFQ/PO no changes	APPROVED	APPROVED
6	800	PURCH_APPROVE	800~B~A	Fixed RFQ	Fixed RFQ	APPROVED	APPROVED
7	800	PURCH_APPROVE	800~B~B	Fixed RFQ/purchase order	Fixed RFQ/purchase order	APPROVED	APPROVED
8	800	PURCH_APPROVE	800~B~S	Blocked	Blocked	PENDING APPROVAL	PENDING APPROVAL
9	800	PURCH_APPROVE	800~B~SELF	No Release Strategy	No Release Strategy	APPROVED	APPROVED
10	800	PURCH_APPROVE	800~F~B	Blocked - Changeable with value	Blocked - Changeable with value	PENDING APPROVAL	PENDING APPROVAL
11	800	PURCH_APPROVE	800~F~G	Released	Released	APPROVED	APPROVED

To configure domainValues_Status_Purch_Cycle_SAP_ECC6.0.csv and To configure domainValues_Status_Purch_Cycle_SAP_4.6c.csv

The file has Purchase Cycle for Purchase requisition, Purchase order

1. CLIENT:

A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

W_STATUS_CLASS :

it denotes the Purchase Cycle Class. It is represented as ' PURCH_CYCLE '

STATUS CODE :

Represents the values in SAP Source. It is defined as a combination of Client~ Purchasing Document Category ~ Cycle Status.

- a. Purchasing Document Category: differentiates the data within SAP between the various Business Objects like Purchase Requisition (B), Purchase Order (F), Scheduling Agreement (L) etc. Use Se16n Transaction and enter table EKPO (Purchasing Document Item). Find field BSTYP(Doc Category) and double click on field name BSTYP. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain- BSTYP. You will be in Dictionary: Display Domain Screen. Move to Value range tab. All the possible Purchasing Document Categories are displayed. Select the Purchasing Document Category.
 - i. Cycle status for Purchase Requisition
 1. “DELETE” – if the value for EBAN.LOEKZ (Deletion Ind.) = "X" then the requisition is marked for deletion in SAP and for these requisitions the cycle status is marked as “DELETE”. (EBAN.MANDT~ EBAN.BSTYP ~"DELETE")
 2. “CLOSE” – if the value for EBAN. EBAKZ (Closed) = "X" then the requisition is marked as Closed in SAP and for these requisitions the cycle status is marked as “CLOSE”. (EBAN.MANDT~EBAN.BSTYP~"CLOSE")
 3. “OPEN” – If the Value for BANPR (Proc. State) = '05' (Release Complete) Or BANPR (Proc. State) = '02' (Active) and EBAN.LOEKZ = " " (Not marked for deletion) and EBAN.EBAKZ = " " (Not Marked as close) then the requisition is considered to be “OPEN”
 4. (EBAN.BANPR = '05' Or EBAN.BANPR = '02') AND EBAN.LOEKZ = " " AND EBAN.EBAKZ = " " then EBAN.MANDT~EBAN.BSTYP~"OPEN")
 5. “PENDING APPROVAL” – If the Value EBAN.FRGKZ = 'S' (Release Indicator) and EBAN.LOEKZ = " " (Not marked for deletion) and EBAN.EBAKZ = " " (Not Marked as close) then the requisition is considered to be “PENDING APPROVAL”
 6. (EBAN.FRGKZ = 'S' AND EBAN.EBAKZ = " " then EBAN.MANDT~EBAN.BSTYP~"PENDING APPROVAL")
 - ii. To Get the value for BANPR (Release Indicator)
 - iii. Release Indicator: Indicates the processing phase the requisition is currently going through (e.g. In process of release). In Case of Purchase requisition the standard SAP values are maintained in the file. You can also use Se16n Transaction and enter table EBAN (Purchase Requisition). Find field BANPR and double click on field name BANPR (Proc. State). You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain- BANPR. You will be in Dictionary: Display Domain Screen. Move to Value range tab. All the possible values are displayed.
- b. Cycle status for Purchase Order
 - i. “DELETE” – if the value for EKPO.LOEKZ (Deletion Ind.) = "L" then the order is marked for deletion in SAP and for these orders the cycle status is marked as “DELETE”. (EKKO.MANDT~EKKO.BSTYP~EKPO.LOEKZ)

- ii. "BLOCK" – if the value for EKPO.LOEKZ (Deletion Ind.) = "S" then the order is marked for block in SAP and for these orders the cycle status is marked as "BLOCK".
(EKKO.MANDT~EKKO.BSTYP~EKPO.LOEKZ)
- iii. "DELIVERY COMPLETE" – if the value for EKPO.ELIKZ (Deliv. Compl.) = "X" then the order is marked as delivery completed in SAP and for these orders the cycle status is marked as "DELIVERY COMPLETE". (EKKO.MANDT~EKKO.BSTYP~EKPO.ELIKZ)
- iv. "OPEN" – If the Value for EKKO.FRGKE (Release Indicator) is null (Not maintained – Purchase order is not under release strategy and is considered as released) Or file_PO_REJECT_SAP.Value = N (Approved – See Configuration of file_PO_REJECT_SAP for more details) and EKPO.LOEKZ (Deletion Ind.) <> "L" (Not marked for deletion) and EKPO.LOEKZ (Deletion Ind.) <> "S" (Not Marked as close) and EKPO.ELIKZ (Deliv. Compl.) <> "X" (not marked as delivery completed) then the Order is considered to be "OPEN" (IF EKPO.LOEKZ <> 'L' or LOEKZ <> 'S' AND EKPO.ELIKZ <> 'X' AND (EKKO.FRGKE is null or file_PO_REJECT_SAP.Value = N) then EKKO.MANDT~EKKO.BSTYP~"OPEN")
- v. "PENDING APPROVAL" – If the Value for EKKO.FRGKE (Release Indicator) is not null (Is maintained – Purchase order is under release strategy) Or file_PO_REJECT_SAP.Value = Y (Blocked – See Configuration of file_PO_REJECT_SAP for more details) and EKPO.LOEKZ (Deletion Ind.) <> "L" (Not marked for deletion) and EKPO.LOEKZ (Deletion Ind.) <> "S" (Not Marked as close) and EKPO.ELIKZ (Deliv. Compl.) <> "X" (not marked as delivery completed) then the Order is considered to be "PENDING APPROVAL" (IF EKPO.LOEKZ <> 'L' or LOEKZ <> 'S' AND EKPO.ELIKZ <> 'X' AND (EKKO.FRGKE is not null or file_PO_REJECT_SAP.Value = Y) then EKKO.MANDT~EKKO.BSTYP~"PENDING APPROVAL")

STATUS_ NAME & STATUS_ DESC.

Status codes "DELETE", "CLOSE", "OPEN", "BLOCK" & "DELIVERY COMPLETE" are specified with their respective words where as "PENDING APPROVAL" is marked as "OTHERS"

W_STATUS_CODE & W_STATUS_DESC:

On the basis of Purchase Cycle of the business object, a status code needs to be assigned to each STATUS CODE. Five W_STATUS_CODE are used. "DELETE", "BLOCK", "CLOSE", "PENDING APPROVAL", "OPEN" & "DELIVERY COMPLETE" .

- a. "DELETE" is used when the business object is Deleted or Cancelled in SAP and it is considered as "CANCELLED" in the file,
- b. "BLOCK" is used when the business object (Purchase Order) is Blocked in SAP and it is considered as "ON HOLD" in the file,
- c. "CLOSE" is used when the business object (Purchase requisition) is marked for close in SAP and it is considered as "COMPLETED" in the file,
- d. "PENDING APPROVAL" is defined when the Purchase order is under release strategy, is blocked if under release, is not marked for deletion, is not blocked, is not marked for close or delivery complete then is considered as "OTHER" in the file,
- e. "OPEN" is defined when the Purchase order is not under release strategy, is Approved if under release, is not marked for deletion, is not blocked, is not marked for close or delivery complete then is considered as "OPEN" in the file,

- f. "DELIVERY COMPLETE" is used when the business object (purchase order) is marked for delivery completion in SAP and it is considered as "COMPLETED" in the file

Figure 138.

	A	B	C	D	E	F	G
1	CLIENT	W_STATUS_CLASS	STATUS_CODE	STATUS_NAME	STATUS_DESC	W_STATUS_CODE	W_STATUS_DESC
2	800	PURCH_CYCLE	800*B~DELETE	DELETED	DELETED	CANCELLED	CANCELLED
3	800	PURCH_CYCLE	800*B~CLOSE	CLOSE	CLOSED	COMPLETED	COMPLETED
4	800	PURCH_CYCLE	800*B~OPEN	OPEN	OPEN	OPEN	OPEN
5	800	PURCH_CYCLE	800*B~PENDING APPROVAL	OTHER	OTHER	OTHER	OTHER
6	800	PURCH_CYCLE	800*F~L	DELETED	DELETED	CANCELLED	CANCELLED
7	800	PURCH_CYCLE	800*F~S	BLOCK	BLOCK	ON HOLD	ON HOLD
8	800	PURCH_CYCLE	800*F~X	DELIVERY COMPLETE	DELIVERY COMPLETE	COMPLETED	COMPLETED
9	800	PURCH_CYCLE	800*F~OPEN	OPEN	OPEN	OPEN	OPEN
10	800	PURCH_CYCLE	800*F~PENDING APPROVAL	OTHER	OTHER	OTHER	OTHER
11							

To configure domainValues_Xact_Types_PO_Line_Type_SAP_ECC6.0.csv and To configure domainValues_Xact_Types_PO_Line_Type_SAP_4.6c.csv

The file has Line type defined for Purchase requisition, Purchase order

1. CLIENT:

A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

2. INTEGRATION_ID:

Stores the primary key or the unique identifier of a record as in the source table.

Represents the values in SAP Source. It is defined as a combination of Client~ Purchasing Document Category ~ Item Category.

- Purchasing Document Category:** differentiates the data within SAP between the various Business Objects like Purchase Requisition (B), Purchase Order (F) etc. Use Se16n Transaction and enter table EKPO (Purchasing Document Item). Find field BSTYP(Doc Category) and double click on field name BSTYP. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain- BSTYP. You will be in Dictionary: Display Domain Screen. Move to Value range tab. All the possible Purchasing Document Categories are displayed. Select the Purchasing Document Category.
- Item Category:** The standard SAP values are maintained in the file. Use SE16n transaction and enter table T163Y (Texts for Item Categories). Provide language filter in SPRAS = "EN" and execute. Select all value for field PSTYP (Item Category). You can pick the STATUS NAME & STATUS DESC from the value of PTEXT(Text for ItCat.)

3. W_XACT_CAT_CODE:

It denotes that the line type is from PROCUREMENT AND SPENT Subject area. It is represented as 'PROCUREMENT AND SPENT'

4. W_XACT_CODE:

It denotes that the line type is for Purchase Order. It is represented as 'PO_LINE_TYPE'

5. XACT_TYPE_CODE :

Represents the values in SAP Source. It is defined as a combination of Client~ Purchasing Document Category ~ Item Category.

- c. Purchasing Document Category: differentiates the data within SAP between the various Business Objects like Purchase Requisition (B), Purchase Order (F) etc. Use Se16n Transaction and enter table EKPO (Purchasing Document Item). Find field BSTYP(Doc Category) and double click on field name BSTYP. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain- BSTYP. You will be in Dictionary: Display Domain Screen. Move to Value range tab. All the possible Purchasing Document Categories are displayed. Select the Purchasing Document Category.
- d. Item Category: The standard SAP values are maintained in the file. Use SE16n transaction and enter table T163Y (Texts for Item Categories). Provide language filter in SPRAS = “EN” and execute. Select all value for field PSTYP (Item Category). You can pick the STATUS NAME & STATUS DESC from the value of PTEXT(Text for ItCat.)

6. XACT_TYPE_CODE_NAME.

The standard SAP values are maintained in the file. Use SE16n transaction and enter table T163Y (Texts for Item Categories). Provide language filter in SPRAS = “EN” and execute. Select all value for field PTEXT(Text for ItCat.) and maintain for the corresponding PSTYP (Item Category) values.

7. W_XACT_TYPE_CODE: Two W_XACT_TYPE_CODE are defined “QUANTITY” & “AMOUNT”.

- a. “QUANTITY” is used for item categories where the quantity has impact on unit price,
- b. “AMOUNT” is used for item categories where the procurement is based on amount like service, Limit etc.

8. W_XACT_TYPE_CODE1: Two W_XACT_TYPE_CODE1 are defined “GOODS” & “SERVICES”.

- a. “GOODS” is used for item categories which are tangible;
- b. “SERVICES” is used for item categories which are Non tangible

Figure 139.

	A	B	C	D	E	F	G	H
1	CLIENT	INTEGRATION_ID	W_XACT_CAT_CODE	W_XACT_CODE	XACT_TYPE_CODE	XACT_TYPE_CODE_NAME	W_XACT_TYPE_CODE	W_XACT_TYPE_CODE1
2	800	800*B*0	PROCUREMENT AND SPENT	PO_LINE_TYPE	800*B*0	Standard	QUANTITY	GOODS
3	800	800*B*2	PROCUREMENT AND SPENT	PO_LINE_TYPE	800*B*2	Consignment	AMOUNT	GOODS
4	800	800*B*3	PROCUREMENT AND SPENT	PO_LINE_TYPE	800*B*3	Subcontracting	QUANTITY	SERVICES
5	800	800*B*5	PROCUREMENT AND SPENT	PO_LINE_TYPE	800*B*5	Third-party	QUANTITY	GOODS
6	800	800*B*7	PROCUREMENT AND SPENT	PO_LINE_TYPE	800*B*7	Stock Transfer	QUANTITY	GOODS
7	800	800*B*9	PROCUREMENT AND SPENT	PO_LINE_TYPE	800*B*9	Service	AMOUNT	SERVICES
8	800	800*F*0	PROCUREMENT AND SPENT	PO_LINE_TYPE	800*F*0	Standard	QUANTITY	GOODS
9	800	800*F*1	PROCUREMENT AND SPENT	PO_LINE_TYPE	800*F*1	Limit	AMOUNT	GOODS
10	800	800*F*2	PROCUREMENT AND SPENT	PO_LINE_TYPE	800*F*2	Consignment	AMOUNT	GOODS
11	800	800*F*3	PROCUREMENT AND SPENT	PO_LINE_TYPE	800*F*3	Subcontracting	QUANTITY	SERVICES
12	800	800*F*5	PROCUREMENT AND SPENT	PO_LINE_TYPE	800*F*5	Third-party	QUANTITY	GOODS
13	800	800*F*9	PROCUREMENT AND SPENT	PO_LINE_TYPE	800*F*9	Service	AMOUNT	SERVICES

To configure domainValues_Xact_Types_Purch_Orders_SAP_ECC6.0.csv and To configure domainValues_Xact_Types_Purch_Orders_SAP_4.6c.csv

The file has Document Type defined for Purchase order

1. CLIENT: A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

2. INTEGRATION_ID: Stores the primary key or the unique identifier of a record as in the source table.

Represents the values in SAP Source. It is defined as a combination of Client~ Purchasing Document Category ~ Document Type.

a. Purchasing Document Category: differentiates the data within SAP between the various Business Objects like Purchase Requisition (B), Purchase Order (F) etc. Use Se16n Transaction and enter table EKPO (Purchasing Document Item). Find field BSTYP(Doc Category) and double click on field name BSTYP. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain- BSTYP. You will be in Dictionary: Display Domain Screen. Move to Value range tab. All the possible Purchasing Document Categories are displayed. Select the Purchasing Document Category.

Document Type: Use SE16n transaction and enter table T161 (Purchasing Document Types). Provide Purchasing Document Category BSTYP = “F” and execute. Select all value for field BSART (Document Type). You can pick XACT_TYPE_DESC from the corresponding field BATXT(Doc. Type Descript.)

3. XACT_CODE: It denotes that the document type is for Purchase Order. It is represented as 'PURCH_ORDERS'.

4. XACT_CAT_CODE: It denotes that the document type is from SUPPLY CHAIN analytics. It is represented as 'SUPPLY_CHAIN '

5. XACT_TYPE_CODE : Represents the values in SAP Source. It is defined as a combination of Client~ Purchasing Document Category ~ Document Type.

b. Purchasing Document Category: differentiates the data within SAP between the various Business Objects like Purchase Requisition (B), Purchase Order (F) etc. Use Se16n Transaction and enter table EKPO (Purchasing Document Item). Find field BSTYP(Doc Category) and double click on field name BSTYP. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain- BSTYP. You will be in Dictionary: Display Domain Screen. Move to Value range tab. All the possible Purchasing Document Categories are displayed. Select the Purchasing Document Category.

c. Document Type: Use SE16n transaction and enter table T161 (Purchasing Document Types). Provide Purchasing Document Category BSTYP = “F” and execute. Select all value for field BSART (Document Type). You can pick XACT_TYPE_DESC from the corresponding field BATXT(Doc. Type Descript.)

6. XACT_TYPE_DESC. Use SE16n transaction and enter table T161 (Purchasing Document Types). Provide Purchasing Document Category BSTYP = “F” and execute. Select all value for field BATXT(Doc. Type Descript.) for the corresponding field BSART (Document Type).

7. W_XACT_TYPE_CODE: Two W_XACT_TYPE_CODE are defined “ORDER” & “AGREEMENT”.

a. “ORDER” is used for Document types other than framework orders;

b. “AGREEMENT” is used for Document types maintained for framework orders (SAP Standard - FO).

Figure 140.

	A	B	C	D	E	F	G
1	CLIENT	INTEGRATION_ID	XACT_CODE	XACT_CAT_CODE	XACT_TYPE_CODE	XACT_TYPE_DESC	W_XACT_TYPE_CODE
34	800	800~F~ECDP	PURCH_ORDERS	SUPPLY_CHAIN	800~F~ECDP	Purchase order	ORDER
35	800	800~F~ECEC	PURCH_ORDERS	SUPPLY_CHAIN	800~F~ECEC	Purchase order	ORDER
36	800	800~F~EUB	PURCH_ORDERS	SUPPLY_CHAIN	800~F~EUB	Purchase order	ORDER
37	800	800~F~FO	PURCH_ORDERS	SUPPLY_CHAIN	800~F~FO	Purchase order	AGREEMENT
38	800	800~F~GNB	PURCH_ORDERS	SUPPLY_CHAIN	800~F~GNB	Purchase order	ORDER
39	800	800~F~GST	PURCH_ORDERS	SUPPLY_CHAIN	800~F~GST	Purchase order	ORDER
40	800	800~F~IN	PURCH_ORDERS	SUPPLY_CHAIN	800~F~IN	Purchase order	ORDER
41	800	800~F~NB	PURCH_ORDERS	SUPPLY_CHAIN	800~F~NB	Purchase order	ORDER
42	800	800~F~UB	PURCH_ORDERS	SUPPLY_CHAIN	800~F~UB	Purchase order	ORDER
43	800	800~F~ZGT1	PURCH_ORDERS	SUPPLY_CHAIN	800~F~ZGT1	Purchase order	ORDER
44	800	800~F~ZNB	PURCH_ORDERS	SUPPLY_CHAIN	800~F~ZNB	Purchase order	ORDER

To configure domainValues_Xact_Types_Purch_Receipt_SAP_ECC6.0.csv and To configure domainValues_Xact_Types_Purch_Receipt_SAP_4.6c.csv

The file has Document Type defined for Goods Receipt against purchase order

1. CLIENT: A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.
2. INTEGRATION_ID: Stores the primary key or the unique identifier of a record as in the source table.

Represents the values in SAP Source. It is defined as a combination of Client~ Transaction Event Type~ Document Type~ Movement Type.

Transaction Event Type : A key allowing the user to differentiate between transactions and events that occur in Inventory Management (for example, goods movements and physical inventory transactions). SAP Standard for Goods Receipt for Purchase Order is "WE".

Use Se16n Transaction and enter table T158V(Inv. Mngmt - General: Transaction/Event Typ). Execute and check for description in field LTEXT(Line Text) "Goods Receipt for Purchase Order" and select the corresponding value for VGART(Trans./Event Type).

- a. Document Type: Use Se16n Transaction and enter table T003 (Document Types). Execute and check for description in field LTEXT(Line Text) "Goods Receipt" and select the corresponding value for BLART (Document Type).
 - b. Movement Type: Specifies a key for the type of goods movement. Each goods movement (for example, purchase order to warehouse) is allocated to a movement type in the system. The Movement type for Purchase order is 101,102,103,104,105,106,122,123,411,412.
3. XACT_CODE: It denotes that the document type is for Goods Receipt for Purchase Order. It is represented as 'PURCH_RCPTS'.
 4. XACT_CAT_CODE: It denotes that the document type is from SUPPLY CHAIN analytics. It is represented as SUPPLY_CHAIN '
 5. XACT_TYPE_CODE : Represents the values in SAP Source. It is defined as a combination of Client~ Transaction Event Type~ Document Type~ Movement Type.

Transaction Event Type : A key allowing the user to differentiate between transactions and events that occur in Inventory Management (for example, goods movements and physical inventory transactions). SAP Standard for Goods Receipt for Purchase Order is “WE”.

Use Se16n Transaction and enter table T158V(Inv. Mngmt - General: Transaction/Event Typ). Execute and check for description in field LTEXT(Line Text) “Goods Receipt for Purchase Order” and select the corresponding value for VGART(Trans./Event Type).

- c. Document Type: Use Se16n Transaction and enter table T003 (Document Types). Execute and check for description in field LTEXT(Line Text) “Goods Receipt” and select the corresponding value for BLART (Document Type).
- d. Movement Type: Specifies a key for the type of goods movement. Each goods movement (for example, purchase order to warehouse) is allocated to a movement type in the system. The Movement type for Purchase order is 101,102,103,104,105,106,122,123,411,412.

6. XACT_TYPE_DESC.

The description is maintained as “Good Receipt” since these are good receipt against purchase order.

7. W_XACT_TYPE_CODE: One W_XACT_TYPE_CODE is defined “RECEIVE”.

“RECEIVE” is used for Transaction event keys, Document types which has receipt against purchase order.

8. W_XACT_TYPE_CODE1: Two W_XACT_TYPE_CODE1 are defined “RECEIVE”, “RETURN TO VENDOR” which differentiates between receipt and return.

- a. “RECEIVE” is receipt against purchase order.
- b. “RETURN TO VENDOR” return of the receipt material to vendor.

Figure 141.

	A	B	C	D	E	F	G	J
1	CLIENT	INTEGRATION_ID	XACT_CODE	XACT_CAT_CODE	XACT_TYPE_CODE	XACT_TYPE_DESC	W_XACT_TYPE_CODE	W_XACT_TYPE_CODE1
2	800	800~WE~WE~101	PURCH_RCPTS	SUPPLY_CHAIN	800~WE~WE~101	Good Receipt	RECEIVE	RECEIVE
3	800	800~WE~WE~102	PURCH_RCPTS	SUPPLY_CHAIN	800~WE~WE~102	Good Receipt	RECEIVE	RECEIVE
4	800	800~WE~WE~103	PURCH_RCPTS	SUPPLY_CHAIN	800~WE~WE~103	Good Receipt	RECEIVE	RECEIVE
5	800	800~WE~WE~104	PURCH_RCPTS	SUPPLY_CHAIN	800~WE~WE~104	Good Receipt	RECEIVE	RECEIVE
6	800	800~WE~WE~105	PURCH_RCPTS	SUPPLY_CHAIN	800~WE~WE~105	Good Receipt	RECEIVE	RECEIVE
7	800	800~WE~WE~106	PURCH_RCPTS	SUPPLY_CHAIN	800~WE~WE~106	Good Receipt	RECEIVE	RECEIVE
8	800	800~WE~WE~122	PURCH_RCPTS	SUPPLY_CHAIN	800~WE~WE~122	Good Receipt	RECEIVE	RETURN TO VENDOR
9	800	800~WE~WE~123	PURCH_RCPTS	SUPPLY_CHAIN	800~WE~WE~123	Good Receipt	RECEIVE	RETURN TO VENDOR
10	800	800~WE~WE~411	PURCH_RCPTS	SUPPLY_CHAIN	800~WE~WE~411	Good Receipt	RECEIVE	RECEIVE
11	800	800~WE~WE~412	PURCH_RCPTS	SUPPLY_CHAIN	800~WE~WE~412	Good Receipt	RECEIVE	RECEIVE

To configure domainValues_Xact_Types_Purch_Requisitions_SAP_ECC6.0.csv and To configure domainValues_Xact_Types_Purch_Requisitions_SAP_4.6c.csv

The file has Document Type defined for Purchase Requisition

1. CLIENT:

A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

2. INTEGRATION_ID:

Stores the primary key or the unique identifier of a record as in the source table.

Represents the values in SAP Source. It is defined as a combination of Client~ Purchasing Document Category ~ Document Type ~ Item Category. Item category helps to further differentiate the transaction type into internal & external

Purchasing Document Category: differentiates the data within SAP between the various Business Objects like Purchase Requisition (B), Purchase Order (F) etc. Use Se16n Transaction and enter table EKPO (Purchasing Document Item). Find field BSTYP(Doc Category) and double click on field name BSTYP. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain- BSTYP. You will be in Dictionary: Display Domain Screen. Move to Value range tab. All the possible Purchasing Document Categories are displayed. Select the Purchasing Document Category.

Document Type: Use SE16n transaction and enter table T161 (Purchasing Document Types). Provide Purchasing Document Category BSTYP = “B” and execute. Select all value for field BSART (Document Type). You can pick XACT_TYPE_DESC from the corresponding field BATXT(Doc. Type Descript.)

- a. Item Category: The standard SAP values are maintained in the file. Use SE16n transaction and enter table T163Y (Texts for Item Categories). Provide language filter in SPRAS = “EN” and execute. Select all value for field PSTYP (Item Category).

3. XACT_CODE:

It denotes that the document type is for Purchase Order. It is represented as 'PURCH_RQLNS'.

4. XACT_CAT_CODE: It denotes that the document type is from SUPPLY CHAIN analytics. It is represented as 'SUPPLY_CHAIN '

5. XACT_TYPE_CODE : Represents the values in SAP Source. It is defined as a combination of Client~ Purchasing Document Category ~ Document Type ~ Item Category. Item category helps to further differentiate the transaction type into internal & external

Purchasing Document Category: differentiates the data within SAP between the various Business Objects like Purchase Requisition (B), Purchase Order (F) etc. Use Se16n Transaction and enter table EKPO (Purchasing Document Item). Find field BSTYP(Doc Category) and double click on field name BSTYP. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain- BSTYP. You will be in Dictionary: Display Domain Screen. Move to Value range tab. All the possible Purchasing Document Categories are displayed. Select the Purchasing Document Category.

Document Type: Use SE16n transaction and enter table T161 (Purchasing Document Types). Provide Purchasing Document Category BSTYP = “B” and execute. Select all value for field BSART (Document Type). You can pick XACT_TYPE_DESC from the corresponding field BATXT(Doc. Type Descript.)

- b. Item Category: The standard SAP values are maintained in the file. Use SE16n transaction and enter table T163Y (Texts for Item Categories). Provide language filter in SPRAS = “EN” and execute. Select all value for field PSTYP (Item Category).

6. XACT_TYPE_DESC. Use SE16n transaction and enter table T161 (Purchasing Document Types). Provide Purchasing Document Category BSTYP = “B” and execute. Select all value for field BATXT(Doc. Type Descript.) for the corresponding field BSART (Document Type).
7. W_XACT_TYPE_CODE: One W_XACT_TYPE_CODE is defined “REQUISITION”.
8. “REQUISITION” is used for all Document types as the business object is Purchase requisition.
9. XACT_SUBTYPE_CODE & W_XACT_SUBTYPE_CODE : Two W_XACT_TYPE_CODE is defined “INTERNAL” & “EXTERNAL”.

- “INTERNAL” is used for item category U (Stock Transport Order) represented in data base as 7 (Seven).
- “EXTERNAL” is used for all other item category other than U (Stock Transport Order).

Figure 142.

	A	B	C	D	E	F	G	H	I
1	CLIENT	INTEGRATION_ID	XACT_CODE	XACT_CAT_CODE	XACT_TYPE_CODE	XACT_TYPE_DESC	W_XACT_TYPE_CODE	XACT_SUBTYPE_CODE	W_XACT_SUBTYPE_CODE
2	800	800*B*MV*0	PURCH_RQINS	SUPPLY_CHAIN	800*B*MV	Purchase requisition	REQUISITION	EXTERNAL	EXTERNAL
3	800	800*B*MV*2	PURCH_RQINS	SUPPLY_CHAIN	800*B*MV	Purchase requisition	REQUISITION	EXTERNAL	EXTERNAL
4	800	800*B*MV*3	PURCH_RQINS	SUPPLY_CHAIN	800*B*MV	Purchase requisition	REQUISITION	EXTERNAL	EXTERNAL
5	800	800*B*MV*5	PURCH_RQINS	SUPPLY_CHAIN	800*B*MV	Purchase requisition	REQUISITION	EXTERNAL	EXTERNAL
6	800	800*B*MV*7	PURCH_RQINS	SUPPLY_CHAIN	800*B*MV	Purchase requisition	REQUISITION	INTERNAL	INTERNAL
7	800	800*B*MV*9	PURCH_RQINS	SUPPLY_CHAIN	800*B*MV	Purchase requisition	REQUISITION	EXTERNAL	EXTERNAL
8	800	800*B*EC*0	PURCH_RQINS	SUPPLY_CHAIN	800*B*EC	Purchase requisition	REQUISITION	EXTERNAL	EXTERNAL

To configure domainValues_Status_Fulfillment_SAP.csv

The file has configuration steps for Fulfilment Status

CLIENT: A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

INTEGRATION_ID: Stores the primary key or the unique identifier of a record as in the source table.

W_STATUS_CLASS: It denotes that the Class for Requisition fulfilment status. It is represented as 'POR_REQUISITION_FULFILLMENT_STATUS'.

STATUS_CODE : Represents the values in SAP & Adaptor's, Defined values. It is defined as a combination of Client~Purchasing Category~SAP values~ Adaptor Value. SAP values are only considered where it is necessary to differentiate between values with similar behaviour but with different resultant , e.g. release indicators have similar behaviour but each indicator has different resultant

In Case of FULFILLMENT_STATUS dependent on Purchase Requisition approval, use SE16n transaction and enter table T161S (Release Indicator: Purchasing Document). Select all value for field FRGKE(Release Indicator). If the corresponding value in field FRANG or FRBST is “X” then the Release indicator donates that the Requisition is released. The Purchase Requisition which is not part of the approval (release strategy) as assumed to be released to the vendor and is donated with an entry “SELF” in place of release indicator.

In Case of FULFILLMENT_STATUS dependent on Purchase Requisition Processing status: use Se16n Transaction and enter table EBAN (Purchase Requisition). Find field STATU (Processing stat) and double click on field name STATU. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain- BANST. You will be in Dictionary: Display Domain Screen. Move to Value range tab. All the possible Purchasing Document Categories are displayed. Select the Purchase Requisition Processing status.

STATUS_NAME & STATUS_DESC: In Case of FULFILLMENT_STATUS dependent on Purchase Requisition approval T161S.FKZTX values can be maintained in the file as STATUS NAME & STATUS_DESC for the corresponding FRGKE(Release Indicator). In Case of FULFILLMENT_STATUS dependent on Purchase Requisition Processing status Short Descript values can be maintained in the file as STATUS NAME & STATUS_DESC for the corresponding STATU (Processing stat).

W_STATUS_CODE: Five W_STATUS_CODE is defined “NOT APPLICABLE”, “FULFILLED”, “UNFULFILLED”, “UNPROCESSED” and” UNAPPROVED”.

“NOT APPLICABLE” is used for extractions where the purchase requisition is deleted or is closed.

“FULFILLED” is used for extractions where the purchase requisition have been fulfilled.

“UNFULFILLED” is used for extractions where the purchase requisition has still receipts pending.

“UNPROCESSED” is used for extractions where the purchase requisitions are in Sourcing (RFQ), or are in approval process or are approved but purchase order is not created.

” UNAPPROVED” is used for extractions where the purchase requisition has been rejected.

W_STATUS_DESC: The same description is used as in what is defined in W_STATUS_CODE

W_SUBSTATUS_CODE : Seven W_SUBSTATUS_CODE is defined “NOT_APPLICABLE”, “FULFILLMENT_TRACKED”, “FULFILLMENT_NOT_TRACKED”, “PENDING_SOURCING”, “PENDING_PO_APPROVAL”, ” PENDING_BUYER_SUBMIT” and ” PENDING_BUYERS_WORKBENCH”

“NOT_APPLICABLE” is used for extractions where the purchase requisition is deleted or is closed or is rejected.

“FULFILLMENT_TRACKED” is used for extractions where the purchase requisition has a purchase order created and is approved and has receipts.

“FULFILLMENT_NOT_TRACKED” is used for extractions where the purchase requisition has a Service or Limit purchase order created.

“PENDING_SOURCING” is used for extractions where the purchase requisition has a Service or Limit purchase order created.

“PENDING_PO_APPROVAL” is used for extractions where the purchase order created against a purchase requisition is pending for approval.

” PENDING_BUYER_SUBMIT” is used for extractions where the purchase requisition is approved but purchase order is not created against this.

” PENDING_BUYERS_WORKBENCH” is used for extractions where the purchase requisition has assignment to vendor through source determination process.

W_SUBSTATUS_DESC: A suitable description as per the W_SUBSTATUS_CODE is provided.

Figure 143.

	A	B	C	D	E	F	G	H	I
1	CLIENT	W_STATUS_CLASS	STATUS_CODE	STATUS_NAME	STATUS_DESC	W_STATUS_CODE	W_STATUS_DESC	W_SUBSTATUS_CODE	W_SUBSTATUS_DESC
2		POR_REQUISITION_F 800 ULFILLMENT_STATUS	800*B~NOT_APPLICABLE	Purchase Requisition Closed	Purchase Requisition Closed	NOT APPLICABLE	NOT APPLICABLE	NOT_APPLICABLE	NOT_APPLICABLE
3		POR_REQUISITION_F 800 ULFILLMENT_STATUS	800*B~FULFILLED	Material Received	Material Received	FULFILLED	FULFILLED	NOT_APPLICABLE	NOT_APPLICABLE
4		POR_REQUISITION_F 800 ULFILLMENT_STATUS	800*B~UNFULFILLED_FULFI LLMENT_TRACKED	Material Not Received	Material Not Received	UNFULFILLED	UNFULFILLE D	FULFILLMENT_TR ACKED	Requisition Processed, Unfulfilled and Fulfillment Tracked
5		POR_REQUISITION_F 800 ULFILLMENT_STATUS	800*B~UNFULFILLED_FULFI LLMENT_NOT_TRACKED	Material Not Received	Material Not Received	UNFULFILLED	UNFULFILLE D	FULFILLMENT_NO T_TRACKED	Requisition Processed but Fulfillment will not be Tracked
6		POR_REQUISITION_F 800 ULFILLMENT_STATUS	800*B~A~UNPROCESSED_P ENDING_SOURCING	Fixed RFQ	Fixed RFQ	UNPROCESSE D	UNPROCES SED	PENDING_SOURC ING	Requisition approved and Pending sourcing

To configure file_APIInvoice_Spend_Type_SAP_ECC6.0.csv and To configure file_APIInvoice_Spend_Type_SAP_4.6c.csv

The file has configuration steps for AP Invoice Spend Type

1. CLIENT: A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational

unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

2. INTEGRATION_ID: Stores the primary key or the unique identifier of a record as in the source table.

' AP_SPEND' denotes that the Class for AP Invoice Spend Type and is represented as ' AP_SPEND'.

Client~ Value Represents the values in SAP Source & Adaptor's, Defined values and is represented by combination of Client~ Value

Value: "ITEM", "FREIGHT" & "TAX" are Adaptor's, Defined values and represent: -

- a. "ITEM" is used for extractions where the AP lines are from Account assignment lines in purchase order. "FREIGHT" is used for extractions where freight line is defined as freight condition in purchase order.
- b. "TAX" is used for extractions where tax lines posted.
- c. "BKPF" is SAP source defined value. "BKPF" is used for invoice lines that are not generated in Material management cycle but directly in finance and are treated as leakage through Material management cycle. The Value 'BKPF' is the value of AWTYP (reference Transaction) . AWTYP – Field from BKPF table, which identifies the SAP module that generated the document. "BKPF" represents Finance

3. XACT_CODE: the Class for AP Invoice Spend Type, it is represented as ' AP_SPEND'

4. XACT_CAT_CODE: It denotes that the document type is from SUPPLY CHAIN analytics. It is represented as 'SUPPLY_CHAIN '

5. XACT_TYPE_CODE : Represents the values in SAP Source & Adaptor's, Defined values. It is defined as a combination of Client~ Value

Value: "ITEM", "FREIGHT" & "TAX" are Adaptor's, Defined values and represent: -

- d. "ITEM" is used for extractions where the AP lines are from Account assignment lines in purchase order. "FREIGHT" is used for extractions where freight line is defined as freight condition in purchase order.
- e. "TAX" is used for extractions where tax lines posted.
- f. "BKPF" is SAP source defined value. "BKPF" is used for invoice lines that are not generated in Material management cycle but directly in finance and are treated as leakage through Material management cycle. The Value 'BKPF' is the value of AWTYP (reference Transaction) . AWTYP – Field from BKPF table, which identifies the SAP module that generated the document. "BKPF" represents Finance

6. XACT_TYPE_DESC. The Description is maintained as "Item", "Freight", "Tax" and "Miscellaneous" for Value: "ITEM", "FREIGHT" , "TAX" and "BKPF" Respectively. Use can maintain their, own descriptions.

7. W_XACT_TYPE_CODE: Two W_XACT_TYPE_CODE is defined "PO MATCHED" & "PO NOT REQUIRED".

8. "PO MATCHED" is used for all values which are generated against the purchase order. "PO NOT REQUIRED" is used for invoice lines that are not generated in Material management cycle but directly in finance and are treated as leakage through Material management cycle.

W_XACT_TYPE_CODE1 : Four W_XACT_TYPE_CODE1 are defined "ITEM", "FREIGHT", "TAX" and "MISCELLANEOUS" .

- a. "ITEM" is used for extractions where the AP lines are from Account assignment lines in purchase order.

- b. "FREIGHT" is used for extractions where freight line is defined as freight condition in purchase order.
- c. "TAX" is used for extractions where tax lines posted.
- d. "MISCELLANEOUS" is used for invoice lines that are not generated in Material management cycle but directly in finance and are treated as leakage through Material management cycle.

Figure 144.

	A	B	C	D	E	F	G	J
1	CLIENT	INTEGRATION_ID	XACT_CODE	XACT_CAT_CODE	XACT_TYPE_CODE	XACT_TYPE_DESC	W_XACT_TYPE_CODE	W_XACT_TYPE_CODE1
2	800	AP_SPEND*800*ITEM	AP_SPEND	SUPPLY_CHAIN	800*ITEM	Item	PO MATCHED	ITEM
3	800	AP_SPEND*800*TAX	AP_SPEND	SUPPLY_CHAIN	800*TAX	Tax	PO MATCHED	TAX
4	800	AP_SPEND*800*FREIGHT	AP_SPEND	SUPPLY_CHAIN	800*FREIGHT	Freight	PO MATCHED	FREIGHT
5	800	AP_SPEND*800*BKPF	AP_SPEND	SUPPLY_CHAIN	800*BKPF	Miscellaneous	PO NOT REQUIRED	MISCELLANEOUS

To configure domainValues_Xact_Types_ConsignedType_ECC60.csv and To configure domainValues_Xact_Types_ConsignedType_4.6c.csv

The file has configuration steps for Consignment type

1. CLIENT:

A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

2. INTEGRATION_ID:

Stores the primary key or the unique identifier of a record as in the source table.

Represents the values Adaptor's, Defined values. It is defined as "CONSIGNED" & "REGULAR" Value:

- a. "CONSIGNED" is used for Consignment Orders and helps in identification.
- b. "REGULAR" is used for all other orders than Consignment Orders.

3. XACT_CODE:

It denotes that the Consignment or Order for Purchase Order. It is represented as 'PURCH_ORDER_CI'.

4. XACT_CAT_CODE:

It denotes that the document type is from SUPPLY CHAIN analytics. It is represented as 'SUPPLY_CHAIN '

5. XACT_TYPE_CODE :

Represents the values Adaptor's, Defined values. It is defined as "CONSIGNED" & "REGULAR" Value:

- c. "CONSIGNED" is used for Consignment Orders and helps in identification.
- d. "REGULAR" is used for all other orders than Consignment Orders.

6. XACT_TYPE_DESC.

The Description is maintained as "Consigned" and "Regular" for Value: "CONSIGNED" & "REGULAR" respectively. Use can maintain their, own descriptions.

7. W_XACT_TYPE_CODE:

Two W_XACT_TYPE_CODE is defined “CONSIGNED” & “REGULAR”:

- a. “CONSIGNED” is used for Consignment Orders and helps in identification.
- b. “REGULAR” is used for all other orders than Consignment Orders.

Figure 145.

	A	B	C	D	E	F	G
1	CLIENT	INTEGRATION_ID	XACT_CODE	XACT_CAT_CODE	XACT_TYPE_CODE	XACT_TYPE_DESC	W_XACT_TYPE_CODE
2	800	CONSIGNED	PURCH_ORDER_CI	SUPPLY_CHAIN	CONSIGNED	Consigned	CONSIGNED
3	800	REGULAR	PURCH_ORDER_CI	SUPPLY_CHAIN	REGULAR	Regular	REGULAR

To configure file_Movement_SAP_ECC6.0.csv and To configure file_Movement_SAP_4.6c.csv

The file has configuration steps for Movement Type. The file is used to convert the Quantity & Amount value to negative where the reversal transaction with reversal movement is performed. This helps in representation of correct Quantity & Amount during aggregation.

1. MVMNT: It denotes SAP movement. Use Se16n Transaction and enter table T156 (Movement Type). Execute and check for description in field BWART (Movement Type).
2. Value: Use Se16n Transaction and enter table T156 (Movement Type). Execute and check for description in field RSTYP (Reserv. cat.) for the corresponding BWART (Movement Type).

Figure 146.

	A	B
1	MVMNT	Value
2	101	1
3	102	-1

To configure domainValues_Xact_Types_AgrLevType_SAP.csv

The file has configuration steps for Agreement Leverage type

1. CLIENT:

A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

2. INTEGRATION_ID:

Stores the primary key or the unique identifier of a record as in the source table.

Represents the adaptor’s values and values in SAP Source. It is defined as a combination of Client~ Purchasing Document Category ~ Adaptor’s Value/Item category Adaptor’s Value: “Contract”, “Quotation”, Scheduling Agreement & “Leakage”.

- a. “Contract” is used for purchase orders that has been created against contact and are also called release order.

- b. "Quotation" is used for purchase orders that have been created without a contact, were a valid Quotation was available.
 - c. "Scheduling Agreement" is used for Scheduling Agreement
 - d. "Leakage" is used for purchase orders that has not been created against valid contact.
- 3. XACT_CODE:

It denotes that the Agreement Leverage for Purchase Order. It is represented as 'PURCH_LEVERAGE_TYPE'.
- 4. XACT_TYPE_CODE :

Represents the adaptor's values and values in SAP Source. It is defined as a combination of Client~ Purchasing Document Category ~ Adaptor's Value/Item category Adaptor's Value: "Contract", "Quotation", Scheduling Agreement & "Leakage".

 - e. "Contract" is used for purchase orders that has been created against contact and are also called release order.
 - f. "Quotation" is used for purchase orders that have been created without a contact, were a valid Quotation was available.
 - g. "Scheduling Agreement" is used for Scheduling Agreement
 - h. "Leakage" is used for purchase orders that has not been created against valid contact.
- 5. Item Category:

The purchase order which do not fall under any of the adaptor values are represented by the item category of the purchase order. Use SE16n transaction and enter table T163Y (Texts for Item Categories). Provide language filter in SPRAS = "EN" and execute. Select all value for field PSTYP (Item Category). You can pick the STATUS NAME & STATUS DESC from the value of PTEXT(Text for ItCat.)
- 6. XACT_TYPE_DESC.

Description for Adaptor's is "PO Created against Contract", "PO Created against Quotation", Scheduling Agreement & "Contract Exists but PO created" for "Contract", "Quotation", Scheduling Agreement & "Leakage" respectively.
- 7. Item Category: The standard SAP values are maintained in the file. Use SE16n transaction and enter table T163Y (Texts for Item Categories). Provide language filter in SPRAS = "EN" and execute. Select all value for field PTEXT(Text for ItCat.) and maintain for the corresponding PSTYP (Item Category) values.
- 8. W_XACT_TYPE_CODE:

Two W_XACT_TYPE_CODE is defined "Negotiated" & "Non-Negotiated":

 - a. "Negotiated" is used for purchase orders which are created against the contract or quotation and for Scheduling Agreement..
 - b. "Non-Negotiated" is used for all other orders that are not created against the contract or quotation.
- 9. W_XACT_TYPE_CODE1:

Two W_XACT_TYPE_CODE1 is defined "Agreement" & "Non-Agreement":

 - a. "Agreement" is used for purchase orders which are created against the contract or quotation and for Scheduling Agreement..

- b. "Non-Agreement" is used for all other orders that are not created against the contract or quotation.

10. W_XACT_TYPE_CODE1:

Three W_XACT_TYPE_CODE2 is defined "Agreement", "Non-Agreement", "Leakage" :

- a. "Agreement" is used for purchase orders which are created against the contract or quotation and for Scheduling Agreement..
- b. "Non-Agreement" is used for all other orders that are not created against the contract or quotation but do not have any valid contract.
- c. "Leakage" is used for all other orders that are not created against the contract or quotation but has any valid contract against which the purchase order could be created.

Figure 147.

	A	B	C	E	F	G	H	I
1	CLIENT	INTEGRATION_ID	XACT_CODE	XACT_TYPE_CODE	XACT_TYPE_DESC	W_XACT_TYPE_CODE	W_XACT_TYPE_CODE1	W_XACT_TYPE_CODE2
2	800	800~F~Contract	PURCH_LEVERAGE_TYPE	800~F~Contract	PO Created against Contract	Negotiated	Agreement	Agreement
3	800	800~F~Quotation	PURCH_LEVERAGE_TYPE	800~F~Quotation	PO Created against Quotation	Negotiated	Agreement	Agreement
4	800	800~F~0	PURCH_LEVERAGE_TYPE	800~F~0	Standard	Non-Negotiated	Non-Agreement	Others
5	800	800~F~1	PURCH_LEVERAGE_TYPE	800~F~1	Limit	Non-Negotiated	Non-Agreement	Others
6	800	800~F~2	PURCH_LEVERAGE_TYPE	800~F~2	Consignment	Non-Negotiated	Non-Agreement	Others
7	800	800~F~5	PURCH_LEVERAGE_TYPE	800~F~5	Third Party	Non-Negotiated	Non-Agreement	Others
8	800	800~F~8	PURCH_LEVERAGE_TYPE	800~F~8	Material Group	Non-Negotiated	Non-Agreement	Others
9	800	800~F~9	PURCH_LEVERAGE_TYPE	800~F~9	Service	Non-Negotiated	Non-Agreement	Others
10	800	800~F~Leakage	PURCH_LEVERAGE_TYPE	800~F~Leakage	Contract Exists but PO created	Non-Negotiated	Non-Agreement	Leakage
11								

To configure file_PO_REJECT_SAP.csv and To configure file_PO_REJECT_SAP.csv

The file has configuration steps for Reject flag. The files help in defining if the purchase order is rejected or not.

1. RELEASE_CODE:

Represents the release indicator and its reject flag value. It is defined as a combination of Client~ Purchasing Document Category ~ Release indicator

2. Purchasing Document Category:

Differentiates the data within SAP between the various Business Objects like Purchase Requisition (B), Purchase Order (F) etc. Use Se16n Transaction and enter table EKPO (Purchasing Document Item). Find field BSTYP(Doc Category) and double click on field name BSTYP. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain- BSTYP. You will be in Dictionary: Display Domain Screen. Move to Value range tab. All the possible Purchasing Document Categories are displayed. Select the Purchasing Document Category for purchase order.

3. Release Indicator defined for purchasing documents. Use SE16n transaction and enter table T16FB (Release Indicator: Purchasing Document). Select all value for field FRGKE (Release Indicator).

4. VALUE: Define "Y" for Release code done rejection of Purchase order and for all other maintain "N"

Figure 148.

	A	B
1	RELEASE_CODE	VALUE
2	800~F~R	N
3	800~F~G	Y
4	800~F~B	N

To configure domainValues_Xact_Shipment_Type_ECC60.csv and To configure domainValues_Xact_Shipment_Type_4.6c.csv

The file has configuration steps for Shipment type

CLIENT: A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

XACT_CODE: It denotes the Subject area. It is represented as 'PROCUREMENT AND SPENT'.

XACT_CAT_CODE: It denotes that the Shipment type is from SUPPLY CHAIN analytics. It is represented as 'SUPPLY_CHAIN'.

XACT_TYPE_CODE: - It is defined as a combination of Client ~ Purchasing Document Category ~ Hardcoded Value.

Purchasing Document Category: differentiates the data within SAP between the various Business Objects like Purchase Requisition (B), Purchase Order (F) etc. Use Se16n Transaction and enter table EKPO (Purchasing Document Item). Find field BSTYP (Doc Category) and double click on field name BSTYP. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain- BSTYP. You will be in Dictionary: Display Domain Screen. Move to Value range tab. All the possible Purchasing Document Categories are displayed. Select the Purchasing Document Category.

Hardcoded Value: - There are four values “FO”, “Contract”, “PO”, “Scheduling_Agreement”: “FO” represents the Framework Order. “Contract” represents Purchase Orders created referencing the Contract. “PO” represents Purchase Order and “Scheduling_Agreement” represents Scheduling agreements.

W_XACT_TYPE_CODE and W_XACT_TYPE_DESC: is defined as “BLANKET” & “STANDARD”: “BLANKET” is used for purchase orders which references a contract or is used for scheduling agreements. “STANDARD” is used for purchase order which is not associated with the agreement.

Figure 149.

XACT_CODE	XACT_CAT_CODE	XACT_CAT_NAME	XACT_TYPE_CODE	XACT_TYPE_DESC	W_XACT_TYPE_CODE	W_XACT_TYPE_DESC
PROCUREMENT AND SPEND	PO_SHIPMENT_TYPE	PO_SHIPMENT_TYPE	800~F~FO	Framework Order	STANDARD	STANDARD
PROCUREMENT AND SPEND	PO_SHIPMENT_TYPE	PO_SHIPMENT_TYPE	800~F~Contract	Release Order	BLANKET	BLANKET
PROCUREMENT AND SPEND	PO_SHIPMENT_TYPE	PO_SHIPMENT_TYPE	800~F~PO	Purchase Order	STANDARD	STANDARD
PROCUREMENT AND SPEND	PO_SHIPMENT_TYPE	PO_SHIPMENT_TYPE	800~L~Scheduling_Agreement	Scheduling Agreement	BLANKET	BLANKET

To configure domainValues_Status_AP_Invoice_Payment_SAP.csv

The file has configuration steps for AP Invoice Payment

CLIENT: A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

INTEGRATION_ID: Stores the primary key or the unique identifier of a record as in the source table.

W_STATUS_CLASS: It denotes that the Class for Account Payable Invoice. It is represented as 'AP_INVOICE_PAYMENT'.

STATUS_CODE : Represents the values in Adaptor's, Defined values. It is defined as a combination of Client~ Value

Value: "FULLY PAID", "NOT PAID" & "PARTIALLY PAID" are Adaptor's, Defined values and represent: -
"FULLY PAID" is used for extractions where the AP Invoice are fully paid. "NOT PAID" is used for extractions where the AP Invoice are not Paid. "PARTIALLY PAID" is used for extractions where AP Invoices are partially paid.

STATUS_NAME & STATUS_DESC. The Description is maintained as "paid", "not paid", "partially paid" for Value: "FULLY PAID", "NOT PAID" & "PARTIALLY PAID" Respectively. Use can maintain their, own descriptions.

W_STATUS_CODE: Three W_XACT_TYPE_CODE is defined "FULLY PAID", "NOT PAID" & "PARTIALLY PAID".

"FULLY PAID" is used for extractions where the AP Invoice are fully paid.

"NOT PAID" is used for extractions where the AP Invoice are not Paid.

"PARTIALLY PAID" is used for extractions where AP Invoices are partially paid.

W_STATUS_DESC : The Description is maintained as "paid", "not paid", "partially paid" for Value: "FULLY PAID", "NOT PAID" & "PARTIALLY PAID" Respectively. Use can maintain their, own descriptions.

Figure 150.

	A	B	C	D	E	F	G
1	CLIENT	W_STATUS_CLASS	STATUS_CODE	STATUS_NAME	STATUS_DESC	W_STATUS_CODE	W_STATUS_DESC
2	800	AP_INVOICE_PAYMENT	800~FULLY PAID	Paid	Paid	FULLY PAID	Fully Paid
3	800	AP_INVOICE_PAYMENT	800~NOT PAID	Not Paid	Not Paid	NOT PAID	Not Paid
4	800	AP_INVOICE_PAYMENT	800~PARTIALLY PAID	Partially Paid	Partially Paid	PARTIALLY PAID	Partially Paid

To configure Code_D_Xact_Types_Shipment_Type_SAP.csv

The file has configuration steps for Shipment type

CLIENT: A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

CATEGORY : Identifies the Code within the Code Dimension

SOURCE_CODE : Stores the primary key or the unique identifier of a record as in the source table. It is defined as a combination of Client ~ Purchasing Document Category ~ Hardcoded Value.

Purchasing Document Category: differentiates the data within SAP between the various Business Objects like Purchase Requisition (B), Purchase Order (F) etc. Use Se16n Transaction and enter table EKPO (Purchasing Document Item). Find field BSTYP (Doc Category) and double click on field name BSTYP. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain- BSTYP. You will be in Dictionary: Display Domain Screen. Move to Value range tab. All the possible Purchasing Document Categories are displayed. Select the Purchasing Document Category.

Hardcoded Value: - There are four values "FO", "Contract", "PO", "Scheduling_Agreement": "FO" represents the Framework Order. "Contract" represents Purchase Orders created referencing the

Contract. “PO” represents Purchase Order and “Scheduling_Agreement” represents Scheduling agreements.

SOURCE_CODE_1 & SOURCE_CODE_2 : Store the spit values of the SOURCE_CODE if the SOURCE_CODE is a combination of values separated by “~”.

MASTER_CODE & MASTER_VALUE : is defined as “BLANKET” & “STANDARD”: “BLANKET” is used for purchase orders which references a contract or is used for scheduling agreements. “STANDARD” is used for purchase order which is not associated with the agreement.

Figure 151.

CLIENT	CATEGORY	SOURCE_CODE	SOURCE_CODE_1	SOURCE_CODE_2	MASTER_CODE	MASTER_VALUE
800	PO_SHIPMENT_TYPE~XACT_TYPE	800~F~FO	800~F	FO	STANDARD	STANDARD
800	PO_SHIPMENT_TYPE~XACT_TYPE	800~F~Contract	800~F	Contract	BLANKET	BLANKET
800	PO_SHIPMENT_TYPE~XACT_TYPE	800~F~PO	800~F	PO	STANDARD	STANDARD
800	PO_SHIPMENT_TYPE~XACT_TYPE	800~L~Scheduling_Agreement	800~L	Scheduling_Agreement	BLANKET	BLANKET

To configure Code_D_Xact_Types_Purch_Requisitions_SAP.csv

The file has Document Type defined for Purchase Requisition

CLIENT: A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

CATEGORY : Identifies the Code within the Code Dimension

SOURCE_CODE: Stores the primary key or the unique identifier of a record as in the source table.

Represents the values in SAP Source. It is defined as a combination of Client~ Purchasing Document Category ~ Document Type ~ Item Category. Item category helps to further differentiate the transaction type into internal & external

Purchasing Document Category: differentiates the data within SAP between the various Business Objects like Purchase Requisition (B), Purchase Order (F) etc. Use Se16n Transaction and enter table EKPO (Purchasing Document Item). Find field BSTYP(Doc Category) and double click on field name BSTYP. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain- BSTYP. You will be in Dictionary: Display Domain Screen. Move to Value range tab. All the possible Purchasing Document Categories are displayed. Select the Purchasing Document Category.

Document Type: Use SE16n transaction and enter table T161 (Purchasing Document Types). Provide Purchasing Document Category BSTYP = “B” and execute. Select all value for field BSART (Document Type).

Item Category: The standard SAP values are maintained in the file. Use SE16n transaction and enter table T163Y (Texts for Item Categories). Provide language filter in SPRAS = “EN” and execute. Select all value for field PSTYP (Item Category).

SOURCE_CODE_1, SOURCE_CODE_2 & SOURCE_CODE_3 : Store the spit values of the SOURCE_CODE if the SOURCE_CODE is a combination of values separated by “~”.

MASTER_CODE & MASTER_VALUE : Use SE16n transaction and enter table T161 (Purchasing Document Types). Provide Purchasing Document Category BSTYP = “B” and execute. Select all value for field BATXT(Doc. Type Descript.) for the corresponding field BSART (Document Type).

Figure 152.

CLIENT	CATEGORY	SOURCE_CODE	SOURCE_CODE_1	SOURCE_CODE_2	SOURCE_CODE_3	MASTER_CODE	MASTER_VALUE
800	PURCH_RQLNS~XACT_TYPE	800~B~MV~0	800~B	MV		0 Standard	Standard
800	PURCH_RQLNS~XACT_TYPE	800~B~MV~2	800~B	MV		2 Consignment	Consignment
800	PURCH_RQLNS~XACT_TYPE	800~B~MV~3	800~B	MV		3 Subcontracting	Subcontracting
800	PURCH_RQLNS~XACT_TYPE	800~B~MV~5	800~B	MV		5 Third-party	Third-party
800	PURCH_RQLNS~XACT_TYPE	800~B~MV~7	800~B	MV		7 Stock Tranfer	Stock Tranfer
800	PURCH_RQLNS~XACT_TYPE	800~B~MV~9	800~B	MV		9 Service	Service
800	PURCH_RQLNS~XACT_TYPE	800~B~EC~0	800~B	EC		0 Standard	Standard
800	PURCH_RQLNS~XACT_TYPE	800~B~EC~2	800~B	EC		2 Consignment	Consignment
800	PURCH_RQLNS~XACT_TYPE	800~B~EC~3	800~B	EC		3 Subcontracting	Subcontracting

To configure Code_D_Xact_Types_Purch_Receipt_SAP.csv

The file has Document Type defined for Goods Receipt against purchase order

CLIENT: A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

CATEGORY : Identifies the Code within the Code Dimension

SOURCE_CODE : Stores the primary key or the unique identifier of a record as in the source table. Represents the values in SAP Source. It is defined as a combination of Client~ Transaction Event Type~ Document Type~ Movement Type.

Transaction Event Type : A key allowing the user to differentiate between transactions and events that occur in Inventory Management (for example, goods movements and physical inventory transactions). SAP Standard for Goods Receipt for Purchase Order is "WE".

Use Se16n Transaction and enter table T158V(Inv. Mngmt - General: Transaction/Event Typ). Execute and check for description in field LTEXT(Line Text) "Goods Receipt for Purchase Order" and select the corresponding value for VGART(Trans./Event Type).

Document Type: Use Se16n Transaction and enter table T003 (Document Types). Execute and check for description in field LTEXT(Line Text) "Goods Receipt" and select the corresponding value for BLART (Document Type).

Movement Type: Specifies a key for the type of goods movement. Each goods movement (for example, purchase order to warehouse) is allocated to a movement type in the system. The Movement type for Purchase order is 101,102,103,104,105,106,122,123,411,412.

SOURCE_CODE_1, SOURCE_CODE_2 & SOURCE_CODE_3 : Store the spit values of the SOURCE_CODE if the SOURCE_CODE is a combination of values separated by "~".

MASTER_CODE & MASTER_VALUE : are defined as "RECEIVE", "RETURN TO VENDOR" which differentiates between receipt and return. "RECEIVE" is receipt against purchase order. "RETURN TO VENDOR", return of the receipt material to vendor.

Figure 153.

CLIENT	CATEGORY	SOURCE_CODE	SOURCE_CODE_1	SOURCE_CODE_2	SOURCE_CODE_3	MASTER_CODE	MASTER_VALUE
800	PURCH_RCPTS~XACT_TYPE	800~WE~WE~101	800~WE	WE		101 Good Receipt	Good Receipt
800	PURCH_RCPTS~XACT_TYPE	800~WE~WE~102	800~WE	WE		102 Good Receipt	Good Receipt
800	PURCH_RCPTS~XACT_TYPE	800~WE~WE~103	800~WE	WE		103 Good Receipt	Good Receipt
800	PURCH_RCPTS~XACT_TYPE	800~WE~WE~104	800~WE	WE		104 Good Receipt	Good Receipt
800	PURCH_RCPTS~XACT_TYPE	800~WE~WE~105	800~WE	WE		105 Good Receipt	Good Receipt
800	PURCH_RCPTS~XACT_TYPE	800~WE~WE~106	800~WE	WE		106 Good Receipt	Good Receipt
800	PURCH_RCPTS~XACT_TYPE	800~WE~WE~411	800~WE	WE		411 Good Receipt	Good Receipt
800	PURCH_RCPTS~XACT_TYPE	800~WE~WE~412	800~WE	WE		412 Good Receipt	Good Receipt
800	PURCH_RCPTS~XACT_TYPE	800~WE~WE~107	800~WE	WE		107 Good Receipt	Good Receipt
800	PURCH_RCPTS~XACT_TYPE	800~WE~WE~108	800~WE	WE		108 Good Receipt	Good Receipt
800	PURCH_RCPTS~XACT_TYPE	800~WE~WE~109	800~WE	WE		109 Good Receipt	Good Receipt
800	PURCH_RCPTS~XACT_TYPE	800~WE~WE~110	800~WE	WE		110 Good Receipt	Good Receipt
800	PURCH_RCPTS~XACT_TYPE	800~WE~WE~122	800~WE	WE		122 Return	Return
800	PURCH_RCPTS~XACT_TYPE	800~WE~WE~123	800~WE	WE		123 Return	Return

To configure Code_D_Xact_Types_Purch_Orders_SAP.csv

The file has Document Type defined for Purchase order

CLIENT: A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

CATEGORY : Identifies the Code within the Code Dimension

SOURCE_CODE : Stores the primary key or the unique identifier of a record as in the source table. Represents the values in SAP Source. It is defined as a combination of Client~ Purchasing Document Category ~ Document Type.

Purchasing Document Category: differentiates the data within SAP between the various Business Objects like Purchase Requisition (B), Purchase Order (F) etc. Use Se16n Transaction and enter table EKPO (Purchasing Document Item). Find field BSTYP(Doc Category) and double click on field name BSTYP. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain- BSTYP. You will be in Dictionary: Display Domain Screen. Move to Value range tab. All the possible Purchasing Document Categories are displayed. Select the Purchasing Document Category.

Document Type: Use SE16n transaction and enter table T161 (Purchasing Document Types). Provide Purchasing Document Category BSTYP = “F” and execute. Select all value for field BSART (Document Type).

SOURCE_CODE_1 & SOURCE_CODE_2 : Store the spit values of the SOURCE_CODE if the SOURCE_CODE is a combination of values separated by “~”.

MASTER_CODE & MASTER_VALUE : are defined as “Purchase Order”, “Scheduling Agreement”. “Purchase Order” is marked for documents type belonging to Purchase order. “Scheduling Agreement” is marked for Scheduling Agreement.

Figure 154.

CLIENT	CATEGORY	SOURCE_CODE	SOURCE_CODE_1	SOURCE_CODE_2	MASTER_CODE	MASTER_VALUE
800	PURCH_ORDERS~XACT_TYPE	800~F~DP00	800~F	DP00	Purchase order	Purchase order
800	PURCH_ORDERS~XACT_TYPE	800~F~DP01	800~F	DP01	Purchase order	Purchase order
800	PURCH_ORDERS~XACT_TYPE	800~F~DP02	800~F	DP02	Purchase order	Purchase order
800	PURCH_ORDERS~XACT_TYPE	800~F~DP03	800~F	DP03	Purchase order	Purchase order
800	PURCH_ORDERS~XACT_TYPE	800~F~DP04	800~F	DP04	Purchase order	Purchase order
800	PURCH_ORDERS~XACT_TYPE	800~F~DP05	800~F	DP05	Purchase order	Purchase order
800	PURCH_ORDERS~XACT_TYPE	800~F~DP06	800~F	DP06	Purchase order	Purchase order
800	PURCH_ORDERS~XACT_TYPE	800~F~DP07	800~F	DP07	Purchase order	Purchase order
800	PURCH_ORDERS~XACT_TYPE	800~F~DP08	800~F	DP08	Purchase order	Purchase order

To configure Code_D_Xact_Types_PO_Line_Type_SAP.csv

The file has Line type defined for Purchase requisition, Purchase order

CLIENT: A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

CATEGORY : Identifies the Code within the Code Dimension

SOURCE_CODE : Stores the primary key or the unique identifier of a record as in the source table. It is defined as a combination of Client~ Purchasing Document Category ~ Item Category.

Purchasing Document Category: differentiates the data within SAP between the various Business Objects like Purchase Requisition (B), Purchase Order (F) etc. Use Se16n Transaction and enter table EKPO (Purchasing Document Item). Find field BSTYP(Doc Category) and double click on field name BSTYP. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain- BSTYP. You will be in Dictionary: Display Domain Screen. Move to Value range tab. All the possible Purchasing Document Categories are displayed. Select the Purchasing Document Category.

SOURCE_CODE_1 & SOURCE_CODE_2 : Store the spit values of the SOURCE_CODE if the SOURCE_CODE is a combination of values separated by “~”.

MASTER_CODE & MASTER_VALUE : Use SE16n transaction and enter table T163Y (Texts for Item Categories). Provide language filter in SPRAS = “EN” and execute. Select all value for field PSTYP (Item Category) and maintain against the corresponding Item Category.

Figure 155.

CLIENT	CATEGORY	SOURCE_CODE	SOURCE_CODE_1	SOURCE_CODE_2	MASTER_CODE	MASTER_VALUE
800	PO_LINE_TYPE~XACT_TYPE	800~B~0	800~B		0 Standard	Standard
800	PO_LINE_TYPE~XACT_TYPE	800~B~2	800~B		2 Consignment	Consignment
800	PO_LINE_TYPE~XACT_TYPE	800~B~3	800~B		3 Subcontracting	Subcontracting
800	PO_LINE_TYPE~XACT_TYPE	800~B~5	800~B		5 Third-party	Third-party
800	PO_LINE_TYPE~XACT_TYPE	800~B~7	800~B		7 Stock Transfer	Stock Transfer
800	PO_LINE_TYPE~XACT_TYPE	800~B~9	800~B		9 Service	Service
800	PO_LINE_TYPE~XACT_TYPE	800~F~0	800~F		0 Standard	Standard
800	PO_LINE_TYPE~XACT_TYPE	800~F~1	800~F		1 Limit	Limit

To configure Code_D_Xact_Types_ConsignedType_SAP.csv

The file has configuration steps for Consignment type

CLIENT: A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

CATEGORY : Identifies the Code within the Code Dimension

SOURCE_CODE : Stores the primary key or the unique identifier of a record as in the source table. Represents the values Adaptor's, Defined values. It is defined as "CONSIGNMENT" & "REGULAR"

Value: "CONSIGNMENT" is used for Consignment Orders and helps in identification. "REGULAR" is used for all other orders than Consignment Orders.

MASTER_CODE & MASTER_VALUE : Is defined "CONSIGNMENT" & "REGULAR": "CONSIGNMENT" is used for Consignment Orders and helps in identification. "REGULAR" is used for all other orders than Consignment Orders.

Figure 156.

CLIENT	CATEGORY	SOURCE_CODE	MASTER_CODE	MASTER_VALUE
800	PURCH_ORDER_CI~XACT_TYPE	CONSIGNMENT	CONSIGNMENT	CONSIGNMENT
800	PURCH_ORDER_CI~XACT_TYPE	REGULAR	REGULAR	REGULAR

To configure domainValues_Xact_Types_AgrLevType_SAP.csv

The file has configuration steps for Agreement Leverage type

CLIENT: A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

CATEGORY : Identifies the Code within the Code Dimension

SOURCE_CODE : Stores the primary key or the unique identifier of a record as in the source table. Represents the adaptor's values and values in SAP Source. It is defined as a combination of Client~ Purchasing Document Category ~ Adaptor's Value/Item category

Adaptor's Value: "Contract", "Quotation", "Scheduling Agreement" & "Leakage". "Contract" is used for purchase orders that has been created against contact and are also called release order. "Quotation" is used for purchase orders that have been created without a contact, were a valid contact was available.

Item Category: The purchase order which do not fall under any of the adaptor values are represented by the item category of the purchase order. Use SE16n transaction and enter table T163Y (Texts for Item Categories). Provide language filter in SPRAS = "EN" and execute. Select all value for field PSTYP (Item Category). You can pick the STATUS NAME & STATUS DESC from the value of PTEXT(Text for ItCat.)

SOURCE_CODE_1 & SOURCE_CODE_2 : Store the spit values of the SOURCE_CODE if the SOURCE_CODE is a combination of values separated by "~".

MASTER_CODE & MASTER_VALUE. Master Code & Master values for Adaptor's is "PO Created against Contract", "PO Created against Quotation", "Scheduling Agreement" & "Contract Exists but PO created" for "Contract", "Quotation", "Scheduling Agreement" & "Leakage" respectively.

Item Category: The standard SAP values are maintained in the file. Use SE16n transaction and enter table T163Y (Texts for Item Categories). Provide language filter in SPRAS = "EN" and execute. Select all value for field PTEXT(Text for ItCat.) and maintain for the corresponding PSTYP (Item Category) values.

Figure 157.

CLIENT	CATEGORY	SOURCE_CODE	SOURCE_CODE_1	SOURCE_CODE_2	MASTER_CODE	MASTER_VALUE
800	PURCH_LEVERAGE_TYPE~XACT_TYPE	800~F~Contract	800~F	Contract	Contract	PO Created against Contract
800	PURCH_LEVERAGE_TYPE~XACT_TYPE	800~F~Quotation	800~F	Quotation	Quotation	PO Created against Quotation
800	PURCH_LEVERAGE_TYPE~XACT_TYPE	800~L~Scheduling_Agreement	800~L	Scheduling_Agreement	Scheduling_Agreement	Scheduling Agreement
800	PURCH_LEVERAGE_TYPE~XACT_TYPE	800~F~0	800~F		0 Standard	Standard
800	PURCH_LEVERAGE_TYPE~XACT_TYPE	800~F~1	800~F		1 Limit	Limit
800	PURCH_LEVERAGE_TYPE~XACT_TYPE	800~F~2	800~F		2 Consignment	Consignment
800	PURCH_LEVERAGE_TYPE~XACT_TYPE	800~F~3	800~F		3 Subcontracting	Subcontracting
800	PURCH_LEVERAGE_TYPE~XACT_TYPE	800~F~5	800~F		5 Third Party	Third Party
800	PURCH_LEVERAGE_TYPE~XACT_TYPE	800~F~8	800~F		8 Material Group	Material Group
800	PURCH_LEVERAGE_TYPE~XACT_TYPE	800~F~9	800~F		9 Service	Service
800	PURCH_LEVERAGE_TYPE~XACT_TYPE	800~F~Leakage	800~F	Leakage	Leakage	Contract Exists but not used by PO

To configure Code_D_Status_AP_Invoice_Payment_SAP.csv

The file has configuration steps for AP Invoice Payment

CLIENT: A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

CATEGORY : Identifies the Code within the Code Dimension

SOURCE_CODE : Stores the primary key or the unique identifier of a record as in the source table. Represents the values in Adaptor's, Defined values. It is defined as a combination of Client~ Value

Value: "FULLY PAID", "NOT PAID" & "PARTIALLY PAID" are Adaptor's, Defined values and represent: - "FULLY PAID" is used for extractions where the AP Invoice are fully paid. "NOT PAID" is used for extractions where the AP Invoice are not Paid. "PARTIALLY PAID" is used for extractions where AP Invoices are partially paid.

MASTER_CODE & MASTER_VALUE. is defined "FULLY PAID", "NOT PAID" & "PARTIALLY PAID".

"FULLY PAID" is used for extractions where the AP Invoice are fully paid.

"NOT PAID" is used for extractions where the AP Invoice are not Paid.

"PARTIALLY PAID" is used for extractions where AP Invoices are partially paid.

Figure 158.

CLIENT	CATEGORY	SOURCE_CODE	MASTER_CODE	MASTER_VALUE
800	AP_INVOICE_PAYMENT~XACT_TYPE	800~FULLY PAID	FULLY PAID	FULLY PAID
800	AP_INVOICE_PAYMENT~XACT_TYPE	800~NOT PAID	NOT PAID	NOT PAID
800	AP_INVOICE_PAYMENT~XACT_TYPE	800~PARTIALLY PAID	PARTIALLY PAID	PARTIALLY PAID

To configure Code_D_APInvoice_Spend_Type_SAP.csv

The file has configuration steps for AP Invoice Spend Type

CLIENT: A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

CATEGORY : Identifies the Code within the Code Dimension

SOURCE_CODE : Stores the primary key or the unique identifier of a record as in the source table. Represents the values in SAP Source & Adaptor's, Defined values. It is defined as a combination of Client~Value

Value: "ITEM", "FREIGHT" & "TAX" are Adaptor's, Defined values and represent: - "ITEM" is used for extractions where the AP lines are from Account assignment lines in purchase order. "FREIGHT" is used for extractions where freight line is defined as freight condition in purchase order. "TAX" is used for extractions where tax lines posted. "BKPF" is SAP source defined value. "BKPF" is used for invoice lines that are not generated in Material management cycle but directly in finance and are treated as leakage through Material management cycle. The Value 'BKPF' is the value of AWTYP (reference Transaction) . AWTYP – Field from BKPF table, which identifies the SAP module that generated the document. "BKPF" represents Finance

MASTER_CODE & MASTER_VALUE: are defined "ITEM", "FREIGHT", "TAX" and "MISCELLANEOUS" . "ITEM" is used for extractions where the AP lines are from Account assignment lines in purchase order. "FREIGHT" is used for extractions where freight line is defined as freight condition in purchase order. "TAX" is used for extractions where tax lines posted. "MISCELLANEOUS" is used for invoice lines that are not generated in Material management cycle but directly in finance and are treated as leakage through Material management cycle.

Figure 159.

CLIENT	CATEGORY	SOURCE_CODE	MASTER_CODE	MASTER_VALUE
800	AP_SPEND~XACT_TYPE	800~ITEM	ITEM	ITEM
800	AP_SPEND~XACT_TYPE	800~TAX	TAX	TAX
800	AP_SPEND~XACT_TYPE	800~FREIGHT	FREIGHT	FREIGHT
800	AP_SPEND~XACT_TYPE	800~BKPF	MISCELLANEOUS	MISCELLANEOUS
800	AP_INV_DISTRIB~XACT_TYPE	800~ITEM	ITEM	ITEM
800	AP_INV_DISTRIB~XACT_TYPE	800~TAX	TAX	TAX
800	AP_INV_DISTRIB~XACT_TYPE	800~FREIGHT	FREIGHT	FREIGHT
800	AP_INV_DISTRIB~XACT_TYPE	800~BKPF	MISCELLANEOUS	MISCELLANEOUS

Configuration Steps for Controlling Your Data Set

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

- Section , "About Configuring the Purchase Receipts Aggregate Table"
- Section , "How to Configure the Purchase Receipts Aggregate Table"
- Section , "About Configuring the Purchase Cycle Lines Aggregate Table"
- Section , "How to Configure the Purchase Cycle Lines Aggregate Table"

About Configuring the Purchase Receipts Aggregate Table

The Purchase Receipts aggregate table (W_PURCH_RCPT_A) is used to capture information about the product receipts received from your suppliers and the purchase orders placed on them by your purchasing organization.

For your initial ELT run, you need to configure the GRAIN parameter for the time aggregation level in the Purchase Receipts Aggregate fact table.

For the incremental ELT run, you need to configure the time aggregation level and the source identification. The source identification value represents the source system you are sourcing data from.

You need to configure two parameters to aggregate the Purchase Receipts table for your incremental run:

- GRAIN
- TIME_GRAIN

These parameters have a preconfigured value of Month. The possible values for the parameter are:

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

The Purchase Receipt Lines aggregate table is fully loaded from the base table in the initial ELT run. The table can grow to millions of records. Thus, the Purchase Receipts aggregate table is not fully reloaded from the base table after each incremental ELT run. Oracle Business Analytics Warehouse minimizes the incremental aggregation effort, by modifying the aggregate table incrementally as the base table is updated. This process is done in four steps:

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

How to Configure the Purchase Receipts Aggregate Table

To load the Purchase Receipts aggregate table (W_PURCH_RCPT_A), you need to configure the post-load-processing parameter file and the source system parameter files, and run the initial workflow and then the incremental workflow. To configure the Purchase Receipts Aggregate Table:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the Administer ELT Parameters link.
3. Display the Application Specific tab, and select Order Management Analytics from the Select BI Application drop down list.
4. Locate the following parameters and use the Parameter Value field to set the value:
 - TIME_GRAIN (default is MONTH, for the SIL_PurchaseReceiptAggregate_Derive_PreLoadImage scenario).
5. Save your changes.

For more information about specifying parameter values, see Section ,

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

About Configuring the Purchase Cycle Lines Aggregate Table

To aggregate the Purchase Cycle Lines table (W_PURCH_CYCLE_LINE_A), you need to configure the parameters in Oracle BI Applications Configuration Manager, and run the initial ELT workflow and then the incremental ELT workflow.

For your initial ELT run, you need to configure the GRAIN parameter for the time aggregation level in the Purchase Cycle Lines Aggregate fact table.

For the incremental ELT run, you need to configure the time aggregation level and the source identification. The source identification value represents the source system you are sourcing data from.

You need to configure two parameters to aggregate the Purchase Cycle Lines table for your incremental run:

- GRAIN
- TIME_GRAIN

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

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

The Purchase Cycle Lines aggregate table is fully loaded from the base table in the initial ELT run. The table can grow to millions of records. The Purchase Cycle Lines aggregate table is not fully reloaded from the base table after an ELT run. Oracle Business Analytics Warehouse minimize the incremental aggregation effort, by modifying the aggregate table incrementally as the base table gets updated. This process is done in four steps:

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

How to Configure the Purchase Cycle Lines Aggregate Table

Before you load the Purchase Cycle Lines aggregate table (W_PURCH_CYCLE_LINE_A), you need to configure the post-load-processing parameter run the initial workflow and then the incremental workflow, as follows.

To configure the Purchase Cycle Lines Aggregate Table:

1. Start Oracle BI Applications Configuration Manager (for more information, see Section 4.5.8.9, "How to Login to Oracle BI Applications Configuration Manager Using A Connection").
2. Select the Administer ELT Parameters link.
3. Display the Application Specific tab, and select Order Management Analytics from the Select BI Application field.
4. Locate the following parameters and use the Parameter Value field to set the value:
 - TIME_GRAIN (default is MONTH, for the SIL_PurchaseCycleLinesAggregate_Derive_PreLoadImage scenario).
 - GRAIN (default is MONTH, for the PLP_PurchaseCycleLinesAggregate_Load_Full scenario).
5. Save your changes.

For more information about specifying parameter values, see Section "How to Set E-LT Parameters In Oracle BI Applications Configuration Manager".

Chapter 8:Configuring Oracle Supply Chain and Order Management Analytics

This chapter describes how to configure Oracle Supply Chain and Order Management Analytics for particular sources to meet your business needs, and contains the following topics:

- Section , "Overview of Oracle Supply Chain and Order Management Analytics for SAP"

- Section , "Configuration Required Before A Full Load for Oracle Supply Chain and Order Management Analytics for SAP"

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

Overview of Oracle Supply Chain and Order Management Analytics for SAP

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

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

The Oracle Supply Chain and Order Management Analytics application consists of orders, invoices, backlogs and inventory. Sales orders are the entry point for the sales process. Invoices are the exit point from the fulfillment process. Backlogs are points of congestion in your fulfillment process. This coverage includes insight into which items are booked, backlogged, and invoiced. This allows you to evaluate the sales performance of individual sales representatives or departments. Oracle Supply Chain and Order Management Analytics also provides you with information on Inventory Transactions, Inventory Balances, Bill of Materials and Customer and Supplier Returns. This enables companies to monitor inventory levels trend to Sales performance to improve cost exposure, increase turnover through inventory level reduction and increased velocity, deploy inventory at the right place and the right time. and better understand Customer and Supplier Returns to maintain quality.

The Oracle Supply Chain and Order Management Analytics application also requires post-load processing mappings to populate its aggregate and derived tables.

Configuration Required Before A Full Load for Oracle Supply Chain and Order Management Analytics for SAP

This section contains configuration steps that you need to perform on Oracle Supply Chain and Order Management Analytics before you do a full data load, and contains the following topics:

- Section 8.2.1, "About Configuring Domain Values and CSV Worksheet Files for Oracle Supply Chain and Order Management Analytics".
- Section "How to Configure Invoice Type Domain Values".
- Section "How to Configure Pick Types Domain Values".
- Section "How to Configure Order Types Domain Values"
- Section "How to Configure Pick Status Domain Values"
- Section "How to Configure Invoice Status Domain Values"

- Section "How to Configure Order Overall Status Domain Values"

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

The table below lists the CSV worksheet files containing domain values for Oracle Supply Chain and Order Management Analytics in the <ODI_HOME>\biapps_odi\odifiles\odidatafiles\lkpfiles directory.

Table 39. (Cont.) Domain Values and CSV Worksheet Files for Oracle Supply Chain and

Worksheet File Name	Description	Session
domainValues_PickTypes_SAP.csv	Lists the Picking Document Type column and the corresponding domain values for the with an SAP R/3 release 4.6c and ECC 6.0 source system For more information about updating the values in this file, see Section "How to Configure Pick Types Domain Values".	SDE_SAP_TransactionTypeDimension_SalesPickLines
domainValues_OrderTypes_SAP.csv	Lists the Order Document Type column and the corresponding domain values with an SAP R/3 release 4.6c and ECC 6.0 source system For more information about updating the values in this file, see Section "How to Configure Order Types Domain Values".	SDE_SAP_TransactionTypeDimension_SalesOrderLines
domainValues_InvoiceTypes_SAP.csv	Lists the Invoice Document Type column and the corresponding domain values with an SAP R/3 release 4.6c and ECC 6.0 source system For more information about updating the values in this file, see Section "How to Configure Invoice Types Domain Values".	SDE_SAP_TransactionTypeDimension_SalesInvoiceLines

domainValues_PickStatus_SAP.csv	Lists the Picking Status Code and the Status Desc columns, and the corresponding domain with an SAP R/3 release 4.6c and ECC 6.0 source system. For more information about updating the values in this file, see Section "How to Configure Pick Status Domain Values".	SDE_SAP_StatusDimension_SalesPickLines
domainValues_InvoiceStatus_SAP.csv	Lists the Invoice Status Code and the Status Desc columns, and the corresponding domain values with an SAP R/3 release 4.6c and ECC 6.0 source system. For more information about updating the values in this file, see Section "How to Configure Invoice Status Domain Values".	SDE_SAP_StatusDimension_SalesInvoiceLine
DomainValue_OrderOverallStatus_SAP.csv	List the Order Status Code column and the corresponding domain values with an SAP R/3 release 4.6c and ECC 6.0 source system. For more information about updating the values in this file, see Section "How to Configure Order Overall Status Domain Values".	SDE_SAP_StatusDimension_SalesOrderLineCycle
DomainValue_OrderScheduleStatus_SAP.csv	List the Order Schedule Status Code column and the corresponding domain values with an SAP R/3 release 4.6c and ECC 6.0 source system. For more information about updating the values in this file, see Section "How to Configure Order Schedule Status Domain Values".	SDE_SAP_StatusDimension_SalesScheduleLineCycle

How to Configure Invoice Type Domain Values

This section explains how to configure Invoice Type Domain Values using the

To configure domainValues_InvoiceTypes_SAP.csv file.

1. CLIENT:

A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

2. Integration ID

Stores the primary key or the unique identifier of a record as in the source table .It is defined as a combination of Client~ XACT_TYPE_CODE ~ XACT_SUBTYPE_CODE

3. XACT_CODE:

It denotes that the line type is for Billing/Invoice. It is represented as 'SALES_IVCLNS'

4. XACT_TYPE_CODE :

Represents the values in SAP Source.

SD Document Category: differentiates the data within SAP between the various Business Objects like Invoice (M), Credit Memo (O) etc. Use Se16n Transaction and enter table TVFK (Billing: Document Types). Find field VBTP (Document cat.) and double click on field name VBTP. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain-VBTP. You will be in Dictionary: Display Domain Screen. Move to Value range tab. All the possible Sales Document Categories are displayed. Select SD Document Category for Invoice or Credit Memo or Debit Memo.

5. XACT_TYPE_DESC:

Document Category Description: The standard SAP values are maintained in the file.

6. XACT_SUBTYPE_CODE:

The standard SAP R/3 System includes a variety of pre-defined sales document types. Use SE16n transaction and enter table TVFK (Billing Document Types). Provide Document category VBTP = "M" and execute. Select all values for field FKART (Billing Type). Repeat procedure for Credit memo and Debit Memo as Document category VBTP = "P" and Document category VBTP = "O" respectively.

7. XACT_SUBTYPE_NAME:

The standard SAP values are maintained in the file. Use SE16n transaction and enter table TVFKT (Texts for Sales Document Types). Provide language filter in SPRAS = "EN" and execute. Select all value from the value of VTEXT (Description for billing document type.)

8. W_XACT_TYPE_CODE: Three W_XACT_TYPE_CODE are defined "STANDARD INVOICE" "CREDIT MEMO" & "DEBIT MEMO".

- a. "STANDARD INVOICE" is the Corresponding domain value to Invoice in SAP,
- b. "CREDIT MEMO" is the Corresponding domain value to Credit Memo in SAP,

c. "DEBIT MEMO" is the Corresponding domain value to Debit Memo in SAP,

9. W_XACT_TYPE_CODE1: W_XACT_TYPE_CODE1 defined are "ORDER ENTRY" and " INTERCOMPANY"

Figure 160.

	A	B	C	D	E	F	G	H	I	J
1	CLIENT	INTEGRATION_ID	XACT_CODE	XACT_TYPE_CODE	XACT_TYPE_DESC	XACT_SUBTYPE_CODE	XACT_SUBTYPE_NAME	W_XACT_TYPE_CODE	W_XACT_TYPE_DESC	W_XACT_TYPE_CODE1
2	800	800*M~F2	SALES_IVCLNS	M	Invoice	F2	Invoice (F2)	Standard Invoice	Standard Invoice	ORDER ENTRY
3	800	800*M~F1	SALES_IVCLNS	M	Invoice	F1	Invoice (F1)	Standard Invoice	Standard Invoice	ORDER ENTRY
4	800	800*M~BIND	SALES_IVCLNS	M	Invoice	BIND	Indir. Invoice Reb.	Standard Invoice	Standard Invoice	ORDER ENTRY
5	800	800*M~BINP	SALES_IVCLNS	M	Invoice	BINP	Ind. Plan. Settlmnt Reb	Standard Invoice	Standard Invoice	ORDER ENTRY
6	800	800*M~BV	SALES_IVCLNS	M	Invoice	BV	Cash Sale	Standard Invoice	Standard Invoice	ORDER ENTRY
7	800	800*M~CBF2	SALES_IVCLNS	M	Invoice	CBF2	BEV Invoice	Standard Invoice	Standard Invoice	ORDER ENTRY
8	800	800*M~CBID	SALES_IVCLNS	M	Invoice	CBID	Ind Invoice Reb CB	Standard Invoice	Standard Invoice	ORDER ENTRY
9	800	800*M~CHFK	SALES_IVCLNS	M	Invoice	CHFK	Bill. Ext. CH Trans. D	Standard Invoice	Standard Invoice	ORDER ENTRY
10	800	800*M~ETO3	SALES_IVCLNS	M	Invoice	ETO3	ETO: Invoice	Standard Invoice	Standard Invoice	ORDER ENTRY

How to Configure Pick Types Domain Values

This section explains how to configure Pick Types Domain Values using the

To configure domainValues_PickTypes_SAP.csv file.

1. CLIENT:

A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

2. Integration ID

Stores the primary key or the unique identifier of a record as in the source table .It is defined as a combination of Client~ XACT_TYPE_CODE ~ XACT_SUBTYPE_CODE

3. XACT_CODE:

It denotes that the line type is for Delivery. It is represented as 'SALES_PCKLNS'

4. XACT_TYPE_CODE :

Represents the values in SAP Source.

Sales Document Category: differentiates the data within SAP between the various Business Objects like Delivery (J). Use Se16n Transaction and enter table TVLK (Delivery Types). Find field VBTP (Document cat.) and double click on field name VBTP. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain-VBTP. You will be in Dictionary: Display Domain Screen. Move to Value range tab. All the possible Sales Document Categories are displayed. Select SD Document Category for Delivery.

5. XACT_TYPE_DESC:

Document Category Description: The standard SAP values are maintained in the file.

6. XACT_SUBTYPE_CODE:

The standard SAP R/3 System includes a variety of pre-defined sales document types. Use SE16n transaction and enter table TVLK (Sales Document Types). Provide Document category VBTP = "J" and execute. Select all values for field LFART (Delivery Type)

7. XACT_SUBTYPE_NAME:

The standard SAP values are maintained in the file. Use SE16n transaction and enter table TVLKT (Texts for Delivery Type Types). Provide language filter in SPRAS = "EN" and execute. Select all value from the value of VTEXT (Description for delivery type.)

8. W_XACT_TYPE_CODE: Three W_XACT_TYPE_CODE are defined “STANDARD INVOICE” “CREDIT MEMO” & “DEBIT MEMO”.
 - a. “STANDARD DELIVERY” is the Corresponding domain value to DeLIVERY

Figure 161.

	A	B	C	D	E	F	G	H	I
1	CLIENT_ID	INTEGRATION_ID	XACT_CODE	XACT_TYPE_CODE	XACT_TYPE_DESC	XACT_SUBTYPE_CODE	XACT_SUBTYPE_NAME	W_XACT_TYPE_CODE	W_XACT_TYPE_DESC
2	800	800~J~BV	SALES_PCKLNS	J	Delivery	BV	Cash Sale	Standard Delivery	Standard Delivery
3	800	800~J~CBLF	SALES_PCKLNS	J	Delivery	CBLF	BEVStandard Delivery	Standard Delivery	Standard Delivery
4	800	800~J~DOG	SALES_PCKLNS	J	Delivery	DOG	Outb.deliv.gds mvmnt	Standard Delivery	Standard Delivery
5	800	800~J~DTR	SALES_PCKLNS	J	Delivery	DTR	Central postng chnge	Standard Delivery	Standard Delivery
6	800	800~J~HOD	SALES_PCKLNS	J	Delivery	HOD	Outb.deliv.GI mvmnt	Standard Delivery	Standard Delivery
7	800	800~J~HTP	SALES_PCKLNS	J	Delivery	HTP	Deliv.for post.chge	Standard Delivery	Standard Delivery
8	800	800~J~JF	SALES_PCKLNS	J	Delivery	JF	Outbound delivery IN	Standard Delivery	Standard Delivery
9	800	800~J~LN	SALES_PCKLNS	J	Delivery	LN	Replenishment div.	Standard Delivery	Standard Delivery
10	800	800~J~LB	SALES_PCKLNS	J	Delivery	LB	Delivery for subcon.	Standard Delivery	Standard Delivery

How to Configure Order Types Domain Values

To configure domainValues_OrderTypes_SAP.csv

1. CLIENT:

A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

2. Integration ID

Stores the primary key or the unique identifier of a record as in the source table .It is defined as a combination of Client~ XACT_TYPE_CODE ~ XACT_SUBTYPE_CODE

3. XACT_CODE:

It denotes that the line type is for Sales Order. It is represented as 'SALES_ORDLNS'

4. XACT_TYPE_CODE :

Represents the values in SAP Source.

Sales Document Category: differentiates the data within SAP between the various Business Objects like Quotation (B), Sales Order (C) etc. Use Se16n Transaction and enter table TVAK (Sales Document Types). Find field VBTP (Document cat.) and double click on field name VBTP. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain-VBTP. You will be in Dictionary: Display Domain Screen. Move to Value range tab. All the possible Sales Document Categories are displayed. Select Sales Document Category.

5. XACT_TYPE_DESC:

Document Category Description: The standard SAP values are maintained in the file.

6. XACT_SUBTYPE_CODE:

The standard SAP R/3 System includes a variety of pre-defined sales document types. Use SE16n transaction and enter table TVAK (Sales Document Types). Provide Document category VBTP = “C” and execute. Select all values for field AUART (Sales Document Type)

7. XACT_SUBTYPE_NAME:

The standard SAP values are maintained in the file. Use SE16n transaction and enter table TVAKT (Texts for Sales Document Types). Provide language filter in SPRAS = "EN" and execute. Select all value from the value of BEZEI (Description for sales document type.)

8. W_XACT_TYPE_CODE: Two W_XACT_TYPE_CODE are defined "REGULAR" & RETURNS".
 - a. "REGULAR" is the Corresponding domain value to Sales order Type,
 - b. "RETUNRS" is the Corresponding domain value to Sales order Type for Returns,

Figure 162.

	A	B	C	D	E	F	G	H	I
1	CLIENT_ID	INTEGRATION_ID	XACT_CODE	XACT_TYPE_CODE	XACT_TYPE_DESC	XACT_SUBTYPE_CODE	XACT_SUBTYPE_NAME	W_XACT_TYPE_CODE	W_XACT_TYPE_DESC
2	800	800~C~AA	SALES_ORDLNS	C	Order	AA	Promotion Order	Regular	Regular
3	800	800~C~AEBO	SALES_ORDLNS	C	Order	AEBO	AEBO order	Regular	Regular
4	800	800~C~AGIS	SALES_ORDLNS	C	Order	AGIS	Quotation (/ Order)	Regular	Regular
5	800	800~C~BIND	SALES_ORDLNS	C	Order	BIND	Indir. Sales Rebate	Regular	Regular
6	800	800~C~BV	SALES_ORDLNS	C	Order	BV	Cash Sale	Regular	Regular
7	800	800~H~CBBO	SALES_ORDLNS	H	Returns	CBBO	BEV Credit Memo Req	Returns	Returns
8	800	800~H~CBDC	SALES_ORDLNS	H	Returns	CBDC	BEV CI Driver Credit	Returns	Returns
9	800	800~C~CBDD	SALES_ORDLNS	C	Order	CBDD	BEV CI Driver Debit	Regular	Regular
10	800	800~C~CBIC	SALES_ORDLNS	C	Order	CBIC	BEV Stand. O. IC CRM	Regular	Regular

How to Configure Pick Status Domain Values

This section explains how to configure Pick Status Domain Values using the

To configure domainValues_Status_Sales_Pckln_SAP.csv file.

1. CLIENT:

A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

2. Integration ID:

Stores the primary key or the unique identifier of a record as in the source table .It is defined as a combination of Client~ SALES_PCKLN~STATUS_CODE, where SALES_PCKLNS is the hard code domain value for Sales Pick line.

3. STATUS_CODE:

It denotes that the Status type is for Pick line.

Document Status: It represents picking status of the delivery item. The various status in SAP are Not Relevant (), Not Yet Processed (A), Partially Processed (B), and Completely Processed (C) etc. To view Status code in SAP, Use Se16n Transaction and enter table VBUP (Sales Document Item Status). Find field KOSTA (Picking Status) and double click on field name KOSTA. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain-STATV. You will be in Dictionary: Display Domain Screen. Click on Value range tab. All the available Document Status are displayed. Select all Values and copy it in column Status_code in domain value file. Here null value for Not Relevant status should be maintained as Not_Relevant in Status Code Column.

4. STATUS_NAME & STATUS_DESC.

The Description is maintained as "Not Relevant", " Open", "Being Processed", " Closed", for Value: "Not Relevant", "Not yet processed", "Partially Processed" & " Completely Processed" Respectively.

5. W_STATUS_CODE:

Four W_STATUS_CODE is defined “Not Relevant”, “Fully Picked”, “Fully Shipped” & “Fully Shipped” Which are mapped with STATUS_Code, as shown in below screen shot.

6. W_STATUS_DESC :

The Description is maintained as “Not Relevant”, “Fully Picked”, “Fully Shipped” & “Fully Shipped” Respectively.

7. W_STATUS_CLASS:

It denotes that the Class for Picking Process.

Figure 163.

	A	B	C	D	E	F	G	H
1	CLIENT	INTEGRATION_ID	STATUS_CODE	STATUS_NAME	STATUS_DESC	W_STATUS_CODE	W_STATUS_DESC	W_STATUS_CLASS
2	800	800~SALES_PCKLN~Not_Relevant	Not_Relevant	Not_Relevant	Not Relevant	Not Relevant	Not Relevant	SALES_PICKING_PROCESS
3	800	800~SALES_PCKLN~A	A	Open	Not yet processed	Fully Picked	Fully Picked	SALES_PICKING_PROCESS
4	800	800~SALES_PCKLN~B	B	Being Processed	Partially Processed	Fully Shipped	Fully Shipped	SALES_PICKING_PROCESS
5	800	800~SALES_PCKLN~C	C	Closed	Completely Processed	Fully Shipped	Fully Shipped	SALES_PICKING_PROCESS

How to Configure Invoice Status Domain Values

This section explains how to configure Invoice Status Domain Values using the domainValues_Status_Sales_Invoice_SAP.csv file.

1. CLIENT:

A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

2. INTEGRATION_ID:

Stores the primary key or the unique identifier of a record as in the source table .It is defined as a combination of Client~ SALES_INVOICE~STATUS_CODE, where SALES_INVOICE is the hard code domain value for Sales Invoice line.

3. STATUS_CODE:

It denotes that the Status type is for Invoice line.

Document Status: It represents status for transfer to accounting of an Invoice. The various status in SAP, such as Billing document blocked for forwarding to FI (A), Posting document has been created (C) etc. To view Status code in SAP, Use Se16n Transaction and enter table VBRK (Billing Document Header Data). Find field RFBSK (Posting Status) and double click on field name RFBSK. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain-RFBSK. You will be in Dictionary: Display Domain Screen. Click on Value range tab. All the available Document Status are displayed. Select all Values and copy it in column Status_code in domain value file.

4. STATUS_NAME & STATUS_DESC.

Enter transaction code Se16n. Enter table VBRK (Billing Document). Search for field RFBSK (Posting status) and double click on Technical name - Double click on the domain - Go to value range tab. Copy and paste the description in domain file with respect to status code.

5. W_STATUS_CODE:

Four W_STATUS_CODE is defined “Open”, “Pending”, “Completed” & “Cancelled”. Which are mapped with STATUS_Code, as shown in below screen shot.

6. W_STATUS_DESC. :

The Description is maintained as “Open”, “Pending”, “Completed” & ” Cancelled” Respectively.

7. W_STATUS_CLASS:

It denotes that the Class for Invoice Process

Figure 164.

	A	B	C	D	E	F	G	H
1	CLIENT	INTEGRATION_ID	STATUS_CODE	STATUS_NAME	STATUS_DESC	W_STATUS_CODE	W_STATUS_DESC	W_STATUS_CLASS
2	800	800~SALES_INVOICE~A	A	Pending	Billing document blocked for forwarding to FI	Pending	Pending	SALES_INVOICE_PROCESS
3	800	800~SALES_INVOICE~B	B	Open	Posting document not created	Open	Open	SALES_INVOICE_PROCESS
4	800	800~SALES_INVOICE~C	C	Completed	Posting document has been created	Completed	Completed	SALES_INVOICE_PROCESS
5	800	800~SALES_INVOICE~D	D	Open	Billing document is not relevant for accounting	Open	Open	SALES_INVOICE_PROCESS
6	800	800~SALES_INVOICE~E	E	Cancelled	Billing Document Canceled	Cancelled	Cancelled	SALES_INVOICE_PROCESS
7	800	800~SALES_INVOICE~F	F	Pending	Posting document not created	Pending	Pending	SALES_INVOICE_PROCESS
8	800	800~SALES_INVOICE~G	G	Pending	Posting document not created	Pending	Pending	SALES_INVOICE_PROCESS
9	800	800~SALES_INVOICE~H	H	Completed	Posted via invoice list	Completed	Completed	SALES_INVOICE_PROCESS
10	800	800~SALES_INVOICE~I	I	Open	Posted via invoice list	Open	Open	SALES_INVOICE_PROCESS
11	800	800~SALES_INVOICE~K	K	Open	Accounting document not created	Open	Open	SALES_INVOICE_PROCESS
12	800	800~SALES_INVOICE~L	L	Open	Billing doc. blocked for transfer to manager	Open	Open	SALES_INVOICE_PROCESS
13	800	800~SALES_INVOICE~M	M	Pending	Analyst Approval refused	Pending	Pending	SALES_INVOICE_PROCESS
14	800	800~SALES_INVOICE~N	N	Open	No posting document due to fund management	Open	Open	SALES_INVOICE_PROCESS

How to Configure Order Overall Status Domain Values

This section explains how to configure Order Overall Status Domain Values using the domainValues_Status_Sales_Order_SAP.csv file.

To configure Order Overall Status Domain Values:

1. CLIENT:

A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

2. INTEGRATION_ID:

Stores the primary key or the unique identifier of a record as in the source table .It is defined as a combination of Client~ SALES_ORDER~STATUS_CODE, where SALES_ORDER is the hard code domain value for Sales Order line.

3. STATUS_CODE:

It denotes that the Status type is for Sales Order line.

Document Status: It represents Overall status of the Sales order item. The various status in SAP are Not Relevant (), Not Yet Processed (A), Partially Processed (B), and Completely Processed (C) etc. To view Status code in SAP, Use Se16n Transaction and enter table VBUP (Sales Document Item Status). Find field GBSTA (Overall Status of the SD Document Item) and double click on field name GBSTA. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain-STATV. You will be in Dictionary: Display Domain Screen. Click on Value range tab. All the available Document Status are displayed. Select all Values and copy it in column Status_code in domain value file. Here null value for Not Relevant status should be maintained as Not_Relevant in Status Code Column. The Status ‘Cancelled’ and ‘Blocked’ are user defined status which are being controlled based on Order Status mapping in Sales Order line table.

4. STATUS_NAME & STATUS_DESC.

The Description is maintained as “Not Relevant”, “Open”, “Being Processed”, “Closed”, “Cancelled”, “Blocked” for Value: “Not Relevant”, “Not yet processed”, “Partially Processed”, “Completely Processed”, “Cancelled” & “Blocked” Respectively.

5. W_STATUS_CODE:

Six W_STATUS_CODE is defined “Entered”, “Booked”, “Being processed”, “Closed”, “Cancelled” & “Blocked”. Which are mapped with STATUS_Code, as shown in below screen shot.

6. W_STATUS_DESC.:

The Description is maintained as Entered”, “Booked”, “Being processed”, “Closed”, “Cancelled” & “Blocked” Respectively.

7. W_STATUS_CLASS:

It denotes that the Class for Sales Order Process

Figure 165.

	A	B	C	D	E	F	G	H
1	CLIENT	INTEGRATION_ID	STATUS_CODE	STATUS_NAME	STATUS_DESC	W_STATUS_CODE	W_STATUS_DESC	W_STATUS_CLASS
2	800	800~SALES_ORDER~Not_Relevant	Not_Relevant	Not Relevant	Not Relevant	Entered	Entered	SALES_ORDER_PROCESS
3	800	800~SALES_ORDER~A	A	Open	Not yet processed	Booked	Booked	SALES_ORDER_PROCESS
4	800	800~SALES_ORDER~B	B	Being Processed	Partially Processed	Being Processed	Being Processed	SALES_ORDER_PROCESS
5	800	800~SALES_ORDER~C	C	Closed	Completely Processed	Closed	Closed	SALES_ORDER_PROCESS
6	800	800~SALES_ORDER~Cancelled	Cancelled	Cancelled	Cancelled	Cancelled	Cancelled	SALES_ORDER_PROCESS
7	800	800~SALES_ORDER~Blocked	Blocked	Blocked	Blocked	Blocked	Blocked	SALES_ORDER_PROCESS

How to Configure Order Schedule Status Domain Values

This section explains how to configure Order Overall Status Domain Values using the domainValues_Status_Sales_Order_SAP.csv file.

To configure Order Schedule Status Domain Values:

8. CLIENT:

A commercially, organizationally and technically self-contained unit within an SAP system is called Client. It has its own master records and set of tables. The definition of client organizational unit is obligatory, as it is the highest level in the SAP system hierarchy. Specifications that entered at this level are valid for all company codes and for all other organizational structure.

9. INTEGRATION_ID:

Stores the primary key or the unique identifier of a record as in the source table .It is defined as a combination of Client~ SALES_SCH~STATUS_CODE, where SALES_SCH is the hard code domain value for Sales Schedule line.

10. STATUS_CODE:

It denotes that the Status type is for Sales Schedule line.

Document Status: It represents Order Schedule status of the Sales order item. The various status in SAP are Not Relevant (), Not Yet Processed (A), Partially Processed (B), and Completely Processed (C) etc. To view Status code in SAP, Use Se16n Transaction and enter table VBUP (Sales Document Item Status). Find field GBSTA (Overall Status of the SD Document Item) and double click on field name GBSTA. You will be in Dictionary: Display Data Element Screen – Data Type Tab. Double click on Domain-STATV. You will be in Dictionary: Display Domain Screen. Click on Value range tab. All the available Document Status are displayed. Select all Values and copy it in column Status_code in domain value file. Here null value for Not Relevant status should be maintained as Not_Relevant in

Status Code Column. The Status 'Cancelled' and 'Blocked' are user defined status which are being controlled based on Order Status mapping in Sales Order line table.

11. STATUS_NAME & STATUS_DESC.

The Description is maintained as "Not Relevant", "Open", "Being Processed", "Closed", "Cancelled", "Blocked", "Not Relevant", "Not yet processed", "Partially Processed", "Completely Processed", "Cancelled" & "Blocked" Respectively.

12. W_STATUS_CODE:

Four W_STATUS_CODE is defined are "Being processed", "Closed", "Cancelled" & "Blocked". Which are mapped with STATUS_Code, as shown in below screen shot.

13. W_STATUS_DESC. :

The Description is maintained as Entered", "Booked", "Being processed", "Closed", "Cancelled" & "Blocked" Respectively.

14. W_STATUS_CLASS:

It denotes that the Class for Sales Schedule Process.

Figure 166.

CLIENT	INTEGRATION_ID	STATUS_CODE	STATUS_NAME	STATUS_DESC	W_STATUS_CODE	W_STATUS_DESC	W_STATUS_CLASS
800	800~SALES_SCH~Not_Relevant	Not_Relevant	Not Relevant	Not Relevant	Being Processed	Being Processed	SALES_SCHEDULE_PROCESS
800	800~SALES_SCH~A	A	Open	Not yet processed	Being Processed	Being Processed	SALES_SCHEDULE_PROCESS
800	800~SALES_SCH~B	B	Being Processed	Partially Processed	Being Processed	Being Processed	SALES_SCHEDULE_PROCESS
800	800~SALES_SCH~C	C	Closed	Completely Processed	Closed	Closed	SALES_SCHEDULE_PROCESS
800	800~SALES_SCH~Cancelled	Cancelled	Cancelled	Cancelled	Cancelled	Cancelled	SALES_SCHEDULE_PROCESS
800	800~SALES_SCH~Blocked	Blocked	Blocked	Blocked	Blocked	Blocked	SALES_SCHEDULE_PROCESS

Chapter 9: Configuring the Oracle Business Intelligence Applications Repository

This chapter describes how to configure the Oracle BI Repository for the Oracle Business Analytics Warehouse, and contains the following topics:

- Configuring the Oracle BI Repository Connections
- Setting up Date Specific Metrics
- Setting Up Additional Time Series Metrics for Oracle Business Analytics Warehouse
- Setting Up Additional Dimension Tables for Oracle Business Analytics Warehouse
- About Oracle BI Time Repository Variables
- About User Authentication
- About the Security or Visibility Configuration
- About the Group Variable
- About the Period Ago Keys for Oracle Business Analytics Warehouse
- About Configuring Usage Tracking for Oracle Business Analytics Warehouse
- About the Incremental Deployment of the Oracle BI Applications Repository

Note: When you access the EnterpriseBusinessAnalytics.rpd repository in Oracle BI Administration Tool, you must log on as the username and password Administrator\Admin123.

Configuring the Oracle BI Repository Connections

The Oracle Business Intelligence Applications repository uses two databases defined in the Physical layer, as follows:

- Oracle Data Warehouse

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 "How to Configure Oracle Business Intelligence Applications Repository Connections".
- For information about configuring repository variables, see "How to Configure Oracle Business Intelligence Applications Repository Variables".

About The Predefined Connection Pools In The Oracle Business Analytics Warehouse

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

- Oracle Business Analytics Warehouse Connection Pool. The Oracle Business Analytics Warehouse Connection Pool is the main connection pool in the Oracle Business Intelligence Applications Repository. You need to configure this connection pool to connect to your physical data warehouse. The connection is used by the session initialization blocks. You can use this connection pool to set up a dynamic data source name.
- Oracle Business Analytics Warehouse Repository Initblocks Connection Pool. You need to configure the Oracle Business Analytics Warehouse Repository Initblocks Connection Pool to connect to the 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 "How to Configure Dynamic Data Source Names".

You also need to configure the following Static variables:

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

How to Configure Dynamic Data Source Names

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

1. Create new session variables: Session_OLAP_DSN and Session_OLAP_USER
2. Create a Session Init Block which uses 'Oracle Business Analytics Warehouse Repository Initblocks Connection Pool' to populate these session variables based on the user login.

3. Add this Initialization Block to the Execution Precedence list of the Authorization Initialization block.
4. Modify the values of Data Source Name and User Name fields in 'Oracle Data Warehouse Connection Pool' to be VALUEOF(Session_OLAP_DSN) and VALUEOF(Session_OLAP_USER) respectively.

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

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

How to Configure Oracle Business Intelligence Applications Repository Connections

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

- Configuring the Oracle Data Warehouse and, database.
- Configuring the Oracle Business Intelligence Applications Repository connection pools.

To configure the Oracle Data Warehouse or Oracle EBS OLTP database:

1. Using the Oracle BI Administration Tool, open the EnterpriseBusinessAnalytics.rpd file in the \$SAHome\OracleBI\Server\Repository folder.
2. In the Physical pane, double-click the Oracle Data Warehouse object.
3. Save the repository.
4. Click Yes to Check Global Consistency.
5. Click OK when the Warnings are displayed.

To configure the Oracle Business Intelligence Applications Repository connection pools:

1. Using the Oracle BI Administration Tool, open the EnterpriseBusinessAnalytics.rpd file in the \$SAHome\OracleBI\Server\Repository folder.
2. In the Physical pane:
 - a. Double-click the Oracle Data Warehouse Connection Pool within the Oracle Data Warehouse.
 - b. Type the database source name in the Data source name box.
 - c. Type your database User ID in the User name box.
 - d. Type your password in the Password box.
3. Repeat Steps a. to d. above for the other connection pools listed above.
4. Save the repository.
5. Click **Yes** to Check Global Consistency.
6. Click **OK** when the Warnings are displayed.

How to Configure Oracle Business Intelligence Applications Repository Variables

The topic explains how to configure the Oracle BI repository variables.

To configure the Oracle Business Intelligence Applications Repository variables:

1. Using the Oracle BI Administration Tool, open the EnterpriseBusinessAnalytics.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, 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.

Deploying the Metadata Repository and Presentation Catalog

After an Oracle BI Applications installation, the Metadata Repository file (RPD) and Presentation Catalog are located on the installation machine. You must use Fusion Middleware Control to deploy and configure the Oracle BI Applications RPD file and Presentation Catalog files. The target Oracle BI EE machine can be the installation machine, or a separate machine.

For information about extending and scaling a deployment, see Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition

To deploy the Metadata Repository and Presentation Catalog:

1. Copy the Metadata Repository and Presentation Catalog to the target Oracle BI EE machine, as follows:

Table 40. *Copying Oracle BI EE Components to an Oracle BI EE Machine*

Copy these files	From here:	To here:
EnterpriseBusinessAnalytics.rpd	Installation machine:ORACLE_HOME\biapps\repository\	Target Oracle BI EE machine:ORACLE_INSTANCE\bifoundation\OracleBIServerComponent\coreapplication_obis<n>\repository
EnterpriseBusinessAnalytics.zip	Installation machine:ORACLE_HOME\biapps\catalog\	Target Oracle BI EE machine:ORACLE_INSTANCE\bifoundation\OracleBIPresentationServicesComponent\coreapplicationobips<n>\catalog The EnterpriseBusinessAnalytics.zip file must be unzipped and the contents copied into this \catalog\ directory.

Note: The target Oracle BI EE machine can be the installation machine, or a separate machine.

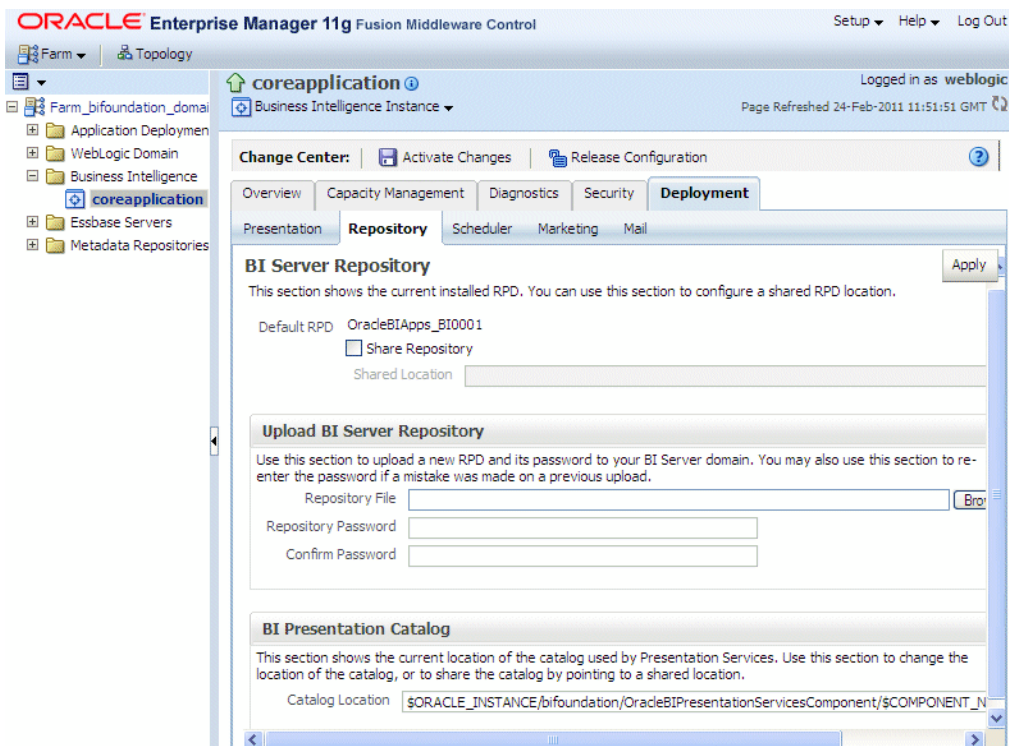
After copying the RPD file, you can verify the repository by using Oracle BI Administrator to open the EnterpriseBusinessAnalytics.rpd file. When you first open the RPD file, use Admin123 as the password to access the repository, then Oracle recommends that you use the File\Change Password option to change the password.

To open the EnterpriseBusinessAnalytics.rpd file using the Oracle Business Intelligence Administration Tool, use the password Admin123.

Note: You should change the default password. Use the Oracle Business Intelligence Administration Tool to do so. For instructions, see the Oracle Business Intelligence Server Administration Guide.

2. Use Oracle Fusion Middleware Control to deploy the Metadata Repository and Presentation Catalog, as follows:
 - a. In Fusion Middleware Control, navigate to the Business Intelligence Overview page.
For example, connect to `http://<hostname>:7001/em`, and select <Farm>, then Business Intelligence, then coreapplication, then Overview.
 - b. Display the Repository tab of the Deployment page.

Figure 167.



On the Repository tab, you can view the name of the current published repository (called the Default RPD).

- c. Click Lock and Edit Configuration to allow changes to be made.
- d. To publish the repository in shared mode, select Share Repository and provide a Shared Location. If you do not select this option, then the repository that you upload on this page is published in distributed mode by default.

- e. Use the Upload BI Repository Server area to specify the location of the `OracleBIAnalytics.rpd` file and the repository password. Click the Help button on the page to access the page-level help for the following options: Repository File option, Repository Password option.
- f. Use the BI Presentation Catalog area to specify the location of the BI Presentation Catalog. Click the Help button on the page to access the page-level help for the following options: Catalog Location option.
- g. Click Apply, then click Activate Changes.
- h. Return to the Business Intelligence Overview page and click Restart.

Applying the Oracle BI Applications Security Policy to the BI Domain

If your deployed BI EE system with Oracle BI Applications is different from the BI EE system used to install Oracle BI Applications, then you must apply the security policy to the BI Domain on the deployed BI EE system by following the steps below.

To apply the Oracle BI Applications Security Policy to the BI Domain:

Note: Machine A is the installation machine. Machine B is the deployment machine.

Shutdown all processes in the BI EE system. Specifically:

the Administration Server

(if there is a cluster) all managed servers in the `bi_cluster` cluster

all opmn managed processes

On machine B, backup and rename the existing `DOMAIN_HOME/config/fmwconfig/system-jazn-data.xml`.

For example, if the BI EE root folder is named `OracleBIEE11g`, then the domain folder location (on Windows) might be `C:\OracleBIEE11g\user_projects\domains\bifoundation_domain\config\fmwconfig`.

Copy the Oracle BI Applications jazn file from machine A at `ORACLE_HOME/biapps/admin/provisioning/system-jazn-data.xml` to machine B at `DOMAIN_HOME/config/fmwconfig`.

Start all the processes in the BI EE system for the Oracle BI Applications security policy to take effect. Specifically:

the Administration Server

(if there is a cluster) all managed servers in the `bi_cluster` cluster

all opmn managed processes

Localizing the Policy Store

The Oracle BI Applications for SAP installer contains different policy stores for different languages. The default policy store that is deployed during the installation is in English. If you want to configure the policy store for any of the other available languages, perform the procedure below.

To configure the policy store for languages other than English

Stop the Oracle BI Services.

Copy the `system-jazn-data_<LN>.xml` file from `$ORACLE_BI_HOME/biapps/admin/provisioning/localization\` to `$DOMAIN_HOME/config/fmwconfig\`.

For example, to implement the policy store for French, you would copy the file system-jazn-data_fr.xml into the \fmwconfig directory.

Back up the existing system-jazn-data.xml file in \$DOMAIN_HOME\config\fmwconfig\.

Rename \$DOMAIN_HOME\config\fmwconfig\system-jazn-data_<LN>.xml to system-jazn-data.xml.

Start Oracle BI Services.

Importing the Oracle BI Applications Version 7.9.7.2 Identity Store (LDIF) File into the Embedded LDAP Server

In this procedure, you will import the standard (out-of-the-box) Oracle BI Applications version 7.9.7.2 LDIF file into the WebLogic Server (embedded LDAP server). This is available under directory ORACLE_BI_HOME\biapps\admin\provisioning\EnterpriseBusinessAnalytics.ldif.

To import the Oracle BI Applications version 7.9.7.2 LDIF file

Log in to the WebLogic Server Administration Console (for example: http://<host name>:7001/console).

Select the name of the security realm into which the LDIF file is to be imported (for example, myrealm).

Select Providers TAB, and select Authentication TAB, then choose the provider into which the LDIF file is to be imported (for example, Providers, and then Authentication and then DefaultAuthenticator).

Select Migration, and then select Import. Enter the full path of LDIF file into text box "Import File on Server" (for example,

D:\BISHIPHOME111150\Oracle_BI1\biapps\admin\provisioning\EnterpriseBusinessAnalytics.ldif)

Click Save.

Refresh the GUIDs after importing the LDIF file

Note: Please refresh the User GUIDs using the steps in section "Refreshing the User GUIDs" of Oracle® Fusion Middleware Administrator's Guide.

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 the following table, 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 41. Invoice Fact Table Example

Date	January	Invoice Amount	4000
Fulfill Amount	5000	Payment Amount	4500

To implement date specific metrics:

1. Using the Oracle BI Administration Tool, open the EnterpriseBusinessAnalytics.rpd. The EnterpriseBusinessAnalytics.rpd file is located in the \OracleBI\server\Repository folder.
2. Right-click on Oracle Business Analytics Warehouse in the Physical layer, and create a new physical alias for the fact table.
3. Create Joins for the physical alias which are similar to the base fact table.
The Join to the date dimension is changed to use the date role in question.
4. Create a new logical table source in the logical fact table that maps the metrics for the physical fact alias.

The grain of the fact table is the same as the base fact table.

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

Setting Up Additional Time Series Metrics for Oracle Business Analytics Warehouse

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

To set up additional time series metrics:

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

Setting Up Additional Dimension Tables for Oracle Business Analytics Warehouse

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

To set up additional dimension tables:

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

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

2. Using the Oracle BI Administration Tool, open the EnterpriseBusinessAnalytics.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 EnterpriseBusinessAnalytics.rpd file.

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.

About Oracle BI Time Repository Variables

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

The following table lists the Oracle BI repository date variables and their descriptions.

Table 42. *Oracle BI Repository Date Variables*

Variable Name	Description
CAL_MONTH_YEAR_AGO	Returns the value of Previous Year Month in the YYYY/MM format.
CURRENT_BALANCE_DK_	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_	Returns the value of the last date key for the available Accounts AR Receivables balance. It is used in Accounts Receivable Account Balance Computation.
CURRENT_BALANCE_DK_	Returns the value of the last date key for the available General GL
CURRENT_DAY	Ledger balance. It is used in General Ledger Account Balance Computation. 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 Fiscal Month in the YYYY / MM format.
NEXT_FSCL_WEEK	Returns the value of Next Fiscal Week in the YYYY Weeknn format.
NEXT_FSCL_YEAR	Returns the value of Next Fiscal Year in the FYYYYY format.
NEXT_MONTH	Returns the value of Next Month in the YYYY / MM format.
NEXT_QUARTER	Returns the value of Next Quarter in the YYYY Q n.
NEXT_WEEK	Returns the value of Next Week in the YYYY Weeknn format.
NEXT_YEAR	Returns the value of Next Year in the YYYY format.
PREVIOUS_FSCL_MONTH	Returns the value of Previous Fiscal Month in the YYYY/MM format.
PREVIOUS_FSCL_QUARTER	Returns the value of Previous Quarter in the YYYY Q n format.
PREVIOUS_FSCL_WEEK	Returns the value of Previous Fiscal Week in the YYYY Weeknn format
PREVIOUS_FSCL_YEAR	Returns the value of Previous Fiscal Year in the FYYYYY format.
PREVIOUS_MONTH	Returns the value of Previous Month in the YYYY/MM format.

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

About User Authentication

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

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

About the Security or Visibility Configuration

The Oracle Business Analytics Warehouse repository is preconfigured with a set of user groups. These groups control the visibility of catalogs in the presentation layer. For more information on adding a user to repository user group, see Oracle Business Intelligence Server Administration Guide. The following table lists the groups in the Oracle Business Analytics Warehouse repository.

Table 43. Repository User Groups

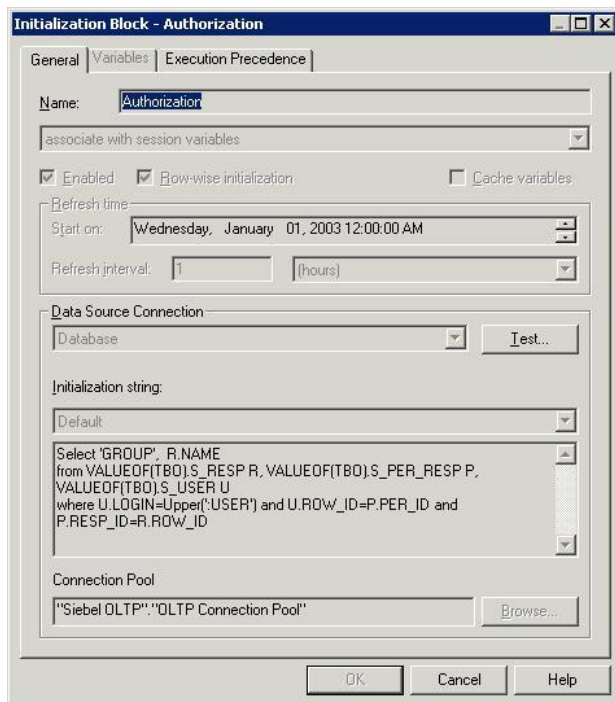
Repository User Group	Description
Administrators	The Administrators user group has all rights and privileges. It cannot be removed.
Agent Scorecard User	This user group is able to view Agent Scorecard application content.
AP Analyst	This user group is able to view application content for Oracle Payables Analytics.
AP Manager	This user group is able to view high-level application content for Oracle Payables Analytics.
CFO	This user group is able to view most of the Oracle Financial Analytics application content.
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.
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).
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.
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 operational application content for Oracle BI Contact Center Telephony Analytics application.
Sales Operations Analytics	This user group is able to view low-level application content for Oracle BI Contact Center Telephony Analytics application.
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.

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 EnterpriseBusinessAnalytics.rpd for the security filters to take effect.

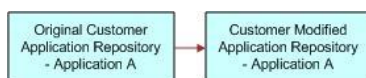
Figure 168. Screenshot of the Initialization Block - Authorization screen



About Configuring Usage Tracking for Oracle Business Analytics Warehouse

Oracle Business Analytics Warehouse supports the accumulation of usage tracking statistics. The Oracle BI repository for Oracle Business Analytics Warehouse is preconfigured with a connection pool to enable the population of the Usage Tracking log table. You need to configure this connection pool to connect to the S_NQ_ACCT table. For more information the Usage Tracking application administering Usage Tracking, see the Oracle Business Intelligence Server Administration Guide.

Figure 169. Oracle Business Analytics Warehouse environment



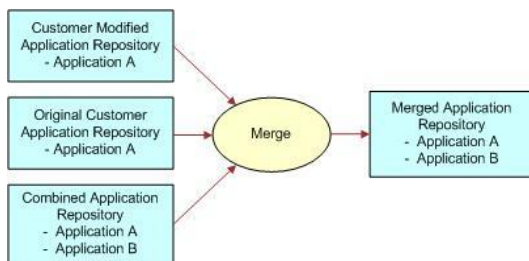
When you purchase another Oracle Business Analytics Warehouse application, you need to extract a new application repository, containing the metadata for all the modules that you have licensed. Use the Administration merge utility to perform a three-way merger of the original repository, the modified 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.

About the Incremental Deployment of the Oracle BI Applications Repository

Oracle Business Analytics Warehouse consist of various families of Oracle Business Intelligence Applications, for example, Supplier Performance Analytics, Contact Center Telephony Analytics, General Ledger & Profitability Analytics, and so on. You can purchase these applications at different times. You can customize functionality and incrementally add new application or applications.

This topic describes the procedure for deploying multiple applications. You can repeat the procedure to add applications incrementally. The figure below shows a single Oracle Business Analytics Warehouse application environment. During installation, you will be asked to specify the application module(s) you have licensed, and the installer will extract the metadata project(s) corresponding to this module(s) into one repository file. You can then modify the Oracle BI repository to suit your business needs.

Figure 170. Merging with an Oracle Business Intelligence Applications Repository



You can repeat this merging procedure to add more Oracle Business Analytics Warehouse applications to the Oracle BI repository

Chapter 10: Customizing the Oracle Business Analytics Warehouse

This chapter describes concepts and techniques for customizing the E-LT functionality in Oracle Business Intelligence Applications, and contains the following topics:

- Overview of Customization in Oracle Business Intelligence Applications
- Category 1 Customizations: Adding Columns to Existing Fact or Dimension Tables
- Category 2 Customizations: Adding Additional Tables

Overview of Customization in Oracle Business Intelligence Applications

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

- What is Customization in Oracle Business Intelligence Applications?
- About the Impact of Patch Installation on Customizations

What is Customization in Oracle Business Intelligence Applications?

In Oracle Business Intelligence Applications, customization is defined as changing the out-of-the-box behavior to enable you to analyze new information in your business intelligence dashboards. For example, you might want to add a column to a dashboard by extracting data from the field MARA.RAUBE and MARA.TEMPB and storing it in the Oracle Business Analytics Warehouse in the STORAGE_COND and TEMP_COND_IND fields field.

The type of data source that you have determines the type of customization that you can do. Data sources can be one of the following types:

- Packaged applications (for example, SAP R/3 4.6c), which use prepackaged adapters.
- Non-packaged data sources Customizations are grouped into the following categories:
- Category 1. In a Category 1 customization, you add additional columns from source systems that have pre-packaged adapters and load the data into existing Oracle Business Analytics Warehouse tables. For more information about performing Category 1 customizations, see Category 1 Customizations: Adding Columns to Existing Fact or Dimension Tables.
- Category 2. In a Category 2 customization, you use pre-packaged adapters to add new fact or dimension tables to the Oracle Business Analytics Warehouse. Category 2 customizations normally require that you build new SDE and SIL mappings. For more information about performing Category 2 customizations, see Category 2 Customizations: Adding Additional Tables
- Category 3. In a Category 3 customization, you use a custom adapter to load data from sources that do not have pre-packaged adapters. For more information about performing Category 3 customizations, see use the Universal adapter to load data from sources that do not have pre-packaged adapters.

The figure below summarizes the category of customization that you can perform for each type of data source and type of modification.

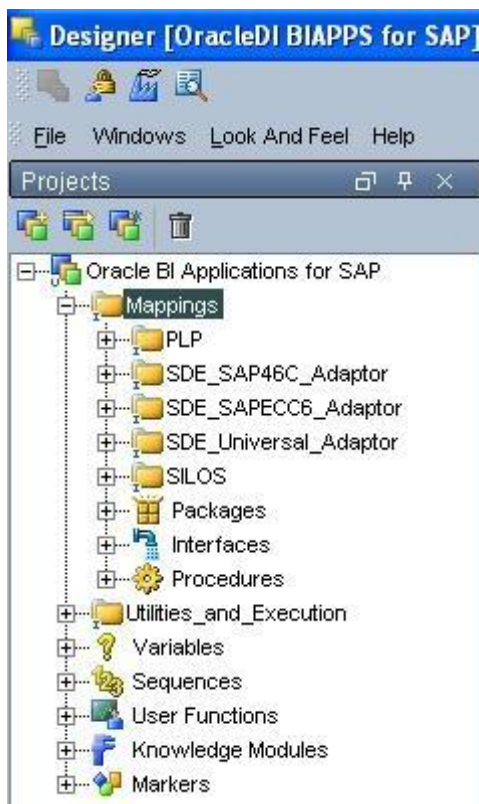
Figure 171. Supported customizations based on data source

		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.

When you customize E-LT Packages and Interfaces, you usually work in the \OracleBI Applications for SAP\Mappings folder in the Models view in ODI Designer.

Figure 172. \OracleBI Applications for SAP\Mappings folder



Note: When you make customizations to any object, create a version of that object before and after the modifications. These versions will enable you to revert to previous functionality if required, and also to manage customizations after a patch upgrade.

About the Customization Process

This chapter explains how to customize your E-LT functionality, after you have performed a Business Analysis and Technical Analysis. This chapter does not cover the other typical tasks that you need to perform, as follows:

- Business Analysis - before you start customization, you typically analyze your current BI dashboards to determine the changes you need to support your business or organization.

- Technical Analysis - when you have agreed your business requirements, you need to determine the technical changes you need to make, by identifying source tables, staging tables, target tables, and ODI Packages and Interfaces that you need to modify.
- RPD Modification - having made the customizations in the E-LT functionality, you need to modify your RPD to expose the new data in your dashboards. For more information about RPD modification, refer to the Oracle Business Intelligence Enterprise Edition documentation library.

About the Impact of Patch Installation on Customizations

This topic explains what you must do to re-apply a customization that you have made if you apply an Oracle Business Intelligence Applications patch that overwrites that customization. For example, if you install a patch that modifies the Financial Analytics application, you might need to manually re-apply customizations that you have made to this application.

Before you apply a patch, you need to 'version' your existing Work Repository (for example, in ODI Designer so that you can detect and re-apply customizations after the patch installation.

A patch only installs changed repository objects, not the whole Work Repository. Therefore, you only need to re-apply customizations to mappings that have been changed by the patch. For example, if a patch only modifies the Financial Analytics application, you only need to manually re-apply customizations that you have made to this application.

Customizations in other applications are not affected by the patch.

To minimize the amount of effort required to re-apply customizations after a patch installation, Oracle recommends that you follow the customization methodology that is described in this chapter.

To maintain customizations after a patch installation:

1. Before you apply a patch, version your existing Work Repository and customized objects by following the steps in [How to Version A Work Repository And Work Repository Objects](#)
2. After you apply a patch, re-apply customizations that were overwritten during the patch installation by following the steps in [How to Re-apply Customizations After a Patch Installation](#).
3. Create a version of the customized objects.

Notes

- When you customize objects, you must evaluate the options and determine the best approach for your environment. If you find that the custom object approach allows the ELT to run in an acceptable amount of time, then this is the preferred approach. If the custom object causes the ELT process to take too long, you might want to consider incorporating the extension into an existing object package or interface.
- When you add custom columns to the Oracle Business Analytics Warehouse, you must make the change in all Oracle Business Analytics Warehouse objects (for example, the W_XXX_D/_F tables as well as the staging tables).

How to Version A Work Repository And Work Repository Objects

This topic explains how to 'version' your Work Repository in ODI Designer so that you can detect and re-apply customizations after a patch installation. After you apply a patch, you must re-apply customizations by following the steps in [How to Re-apply Customizations After a Patch Installation](#)

After you 'version' an object in ODI Designer, use the Version Browser to manage versions (that is, select File, then Version Browser).

To re-apply customizations after a patch installation:

1. In ODI Designer, display the Projects view.
2. Right-click the Oracle BI Applications for SAPproject and choose Version, then Create to display the Create: <Project Name> dialog.
3. Use the Create: <Project Name> dialog to specify a unique version number and optional description, and click OK.
4. Run the package 'Export Changed Objects' in the folder 'Oracle BI Applications 7.9.7.2\Utilities_and_Execution\Utilities\User'.
5. This package will create a file called 'List_Of_Modified_Objects.txt' in the folder location that you specify. This TXT file contains a list of customized objects. This package will also create XML export files for the changed objects in the folder location that you specify, within the date range.

How to Re-apply Customizations After a Patch Installation

This topic explains how to re-apply customizations after you have applied a patch. Initially, any customized objects in the area affected by a patch are over-written with a new version of the object, which overwrites the customization. You use the Version Comparison utility in ODI Designer on the customized objects to compare the new version of an object and the original customized version of that object. When Version Comparison utility has performed the comparison, you can do one of the following:

- Manually re-apply the customizations to the object.
- Retain the new non-customized version object provided by Oracle (in the patch).
- Restore the previous customized version of the object.

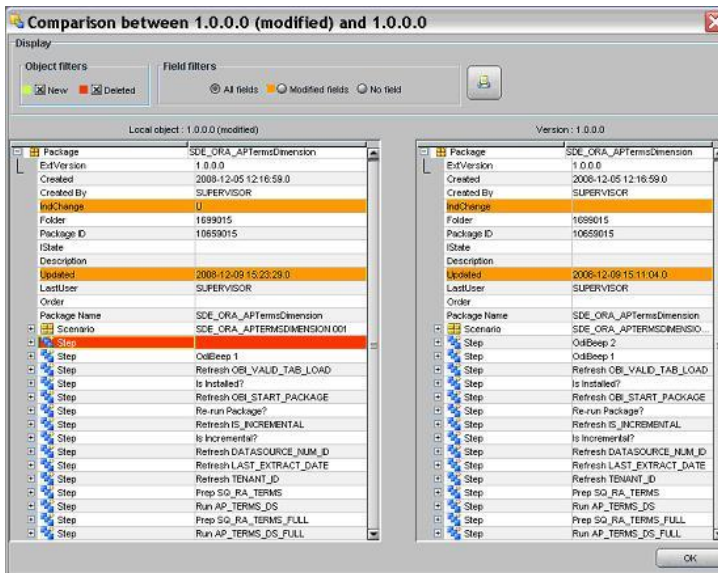
For example, before a patch installation, you might have modified the expression of the CONTAINER_CODE column in the Interface SDE_SAP_ProductDimension.W_PRODUCT_DS. After the patch installation, the expression of the CONTAINER_CODE column might be reverted back. To reapply the customization, check the expression difference using the comparison dialog, and then replace the current version column expression with the previous version of the column expression.

Before you can re-apply customizations, you must have versioned your Work Repository and customized objects by following the steps in [How to Version A Work Repository And Work Repository Objects](#).

To re-apply a customization after a patch installation:

1. In ODI Designer, locate the object that you want to compare with the original (customized version).
2. For example, if you customized the Package SDE_ORA_APTermsDimension in the SDE_ORA_APTermsDimension project folder, locate this object in the Project tab.
3. Right-click the object (for example, a package), and choose Version, then 'Compare with version' to display the Compare With dialog.
4. Use the Compare With dialog to select the original version, then click OK to display the Comparison dialog.

Figure 173.



5. Use the Comparison dialog to determine what customizations need to be re-applied.
6. Manually re-apply the customizations on the respective objects. For example, to restore a previous (customized) version of an object, right click on the object and select Version, then Restore, and select the version of the object that contains the customization.
7. Alternatively, you could manually apply a customization to the new version of the object by editing the new object in ODI Designer and making the changes again. For example, you edit an Interface, and change the expression for a column in the Target Datastore by editing the value in the Implementation field.
8. Repeat steps 1 to 5 for every customization that you need to re-apply.
9. Create a new version of each customized object.

Category 1 Customizations: Adding Columns to Existing Fact or Dimension Tables

Category 1 customizations add additional columns from source systems that have pre-packaged adapters and load the data into existing Oracle Business Analytics Warehouse tables.

This topic contains the following topics:

- About Extending Mappings
- Typical Steps to Extend Mappings in the Oracle Business Analytics Warehouse
- Example of Extending the Oracle Business Analytics Warehouse

Including a Source Table for the Change Capture Process

If you are bringing in data from a new table that was not previously included in an existing SDE package, you might need to create a change capture mapping. When a row changes in the new table, the change capture mapping marks the corresponding row in the main table as changed. Change capture processes can degrade ELT performance. Therefore, change capture processes should only be implemented if required.

ABAP Process (Consult ABAP Consultant)

Create additional entries in table ZTOBI_CONFIG for the new enhanced fields using the instructions below:

1. Use transaction code SE37, and provide the function module name FM: ZOBI_CONFIG_TABLE_UPDATE. Click on change button for use shortcut key F6.
2. Find the code "IF NOT it_config[] IS INITIAL." In the source code. Normally it's at line no. 5346.
3. Insert the below code before line 5346 :

```
wa_config-srctable='MARA'.
```

```
wa_config-fieldname='TEMPB'.
```

```
wa_config-fieldname=' RAUBE '.
```

```
wa_config-trgtable='W_PRODUCT_D'.
```

```
APPEND wa_config TO it_config.
```

```
CLEAR wa_config.
```

ETL Process

1. Create or modify the Z table with the additional fields from the new source table. Only key fields from the new table needs to be included.
2. Include the Ztable into interface and join with the new source table to fetch the incremental data.
3. Steps under the section " Example for Extracting Data from an SAP ERP system" needs to be repeated for mapping the fields from the new source to the staging table and target table.

Silos process (Full and Incremental load process):

1. Identify whether the columns from the new fields to be mapped to the existing target fields or new fields need to be added to the target table. In the second case follow the instruction of adding new columns in section "Example for Extracting Data from an SAP ERP system".
2. If any new field are to be added , we need to define its behavior in the datastore for target table(e.g. overwrite on change or add row on change) based on you business logic.
3. Define/Modify the target table mapping by following the instruction in "Example for Extracting Data from an SAP ERP system".

- Other Types of Customizations Requiring Special Handling

About Extending Mappings

Category 1 customizations involve extracting additional columns from source systems for which pre-packaged adapters are included (for example, Oracle) and loading the data into existing Oracle Business Analytics Warehouse tables. For Category 1 customizations, data can also come from non-packaged sources, but this section assumes that the sources have already been mapped with a Universal adapter and only need to be extended to capture additional columns. (The initial mapping of a Universal adapter is considered a Category 3 customization. For information, see Category 3 Customizations

Category 3 customizations use the Universal adapter to load data from sources that do not have pre-packaged adapters.

In order to see additional columns in the Oracle Business Analytics Warehouse, the columns must first be passed through the ELT process. The existing mappings and tables are extensible. Oracle Business

Intelligence Applications provides a methodology to extend preconfigured mappings to include these additional columns and load the data into existing tables.

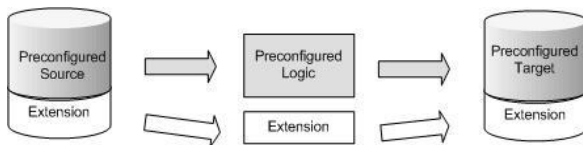
Oracle Business Intelligence Applications recognizes two types of customization: extension and modification. The supported extension logic allows you to add to existing objects. For example, you can extract additional columns from a source, pass them through existing mappings, and populate new columns added to an existing table. Generally, Oracle Business Intelligence Applications does not allow you to modify existing logic or columns. You should not change existing calculations to use different columns, and you should not remap existing columns to be loaded from different sources.

For example, if you want to calculate revenue differently from the existing logic, you should create a new transformation and connect that calculation to a new column, for example, X_REVENUE. You can then remap the Oracle Business Intelligence repository to point to the new X_REVENUE column.

Most mappings have a single placeholder column, named X_CUSTOM, that marks a safe path through the ODI interfaces. If you add transformations to the mapping, they should follow the same route through the mapping as X_CUSTOM.

In the figure below, the preconfigured logic is shaded in gray. You should not modify anything contained within these objects. You should add customizations to existing objects rather than creating new packages and interfaces, which allows them to run parallel to the existing logic.

Figure 174. Preconfigured logic and customizations



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 an ODI package in the Oracle Business Analytics Warehouse:

1. Create a version of the folder that contains the ODI Packages and Interfaces that you need to customize.
2. For example, in ODI Designer, display the Projects view, right-click on a folder, and choose Version, then Create.
3. Extend the source and target tables by making changes to the tables in the database. You then can use ODI to reverse-engineer the additive changes on the source and target definitions into models in ODI (which replaces the existing definitions), or manually edit the existing definition.
4. As a best practice, Oracle recommends that you name custom columns with a X_ prefix to make it easier to distinguish custom columns that have been added to an existing table and to ensure there are no name conflicts with any columns Oracle might add later to that table.
5. Extend the SDE Package and Interfaces by mapping the additional columns, as follows:

- a. In the ODI Package editor, display the Projects view, and open the package.
For example, you might open the Oracle BI Applications .1\Mappings\SDE_SAPECC6_Adaptor\SDE_SAP_ProductDimension folder, and edit the SDE_SAP_ProductDimension Package.
 - b. Display the Diagram tab.
 - c. Edit either the SQ_BCI_ interface or the LKP_ interface to display the Interface: <Name> dialog.
 - d. For example, the SDE_ORA_OrganizationDimension_Customer Package, you might edit the SQ_BCI_CUSTOMERS Interface.
 - e. Display the Diagram tab.
 - f. Map the new column in the Source table to the Target Datastore.
 - g. Click OK to save the changes.
 - h. In the ODI Package editor, edit the last interface in the sequence (for example, named RUN <target table name>).
 - i. Display the Diagram tab.
 - j. Map the new column in the Source table to the Target Datastore.
 - k. Click OK to save the changes.
 - l. Repeat steps a. to j. for both branches within the package (both full load and incremental load).
For example, in the SDE_ORA_OrganizationDimension_Customer Package, you might also edit the SQ_BCI_CUSTOMERS_FULL Interface.
6. Repeat step 3 for the SIL package and interfaces.
 7. For example, you might open the Oracle BI Applications 7.9.7.2\Mappings\SILOS\SIL_OrganizationDimension folder, and edit the SIL_OrganizationDimension Package.
 8. For a detailed example, see Example of Loading Data from the Staging Area into an Existing Target Table
 9. Regenerate the scenarios for the packages that you have modified.

You are now ready to perform E-LT with the modified scenarios.

Example of Extending the Oracle Business Analytics Warehouse

This section contains a worked example of adding additional columns from source systems that have pre-packaged adapters and loading the data into existing Oracle Business Analytics Warehouse tables (known as a Category 1 customization).

This section contains the following topics:

- Overview to the Example
- Example of Loading Data from an SAP ERP Data Packaged Source into the ODI Staging Area
- Example of Loading Data from the Staging Area into an Existing Target Table
- Tips for Modifying the SQ_BCI_ Interface
- Including a Source Table for the Change Capture Process

Overview to the Example

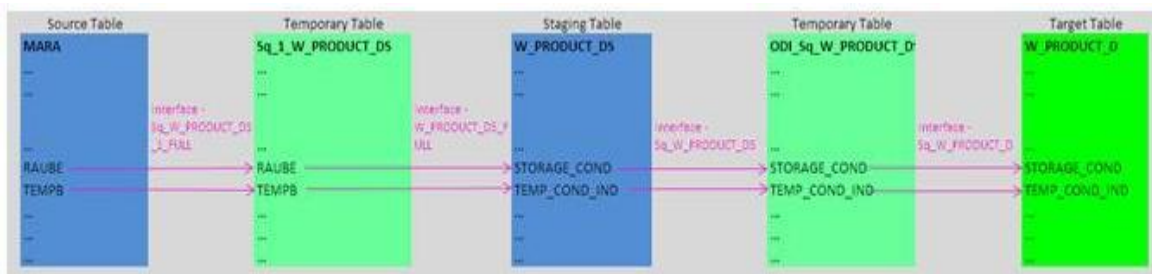
In this example, a company has identified additional fields in a source system table MARA that need to be added to the Oracle Business Analytics Warehouse table W_PRODUCT_D. Data is passed from an existing source table to an existing target table, known as a category 1 customization. The company uses an extension field to capture information related to organizations referred to as STORAGE_COND and TEMP_COND_IND.

In this example, you want to extract information from the following two fields that are not extracted by the out-of-the-box application:

- MARA.RAUBE RAUBE and MARA.TEMPB are currently not extracted from the source table MARA into the temporary table SQ_1_W_PRODUCT_D.

The diagram below shows the two fields RAUBE and TEMPB as they pass from the source system table to the target table via the tables: MARA to SQ_1_W_PRODUCT_D to W_PRODUCT_DS to ODI_Sq_W_PRODUCT_DS to W_PRODUCT_D.

Figure 175. Passing two new fields from the source table to the target table



The customization is done in two parts, as follows:

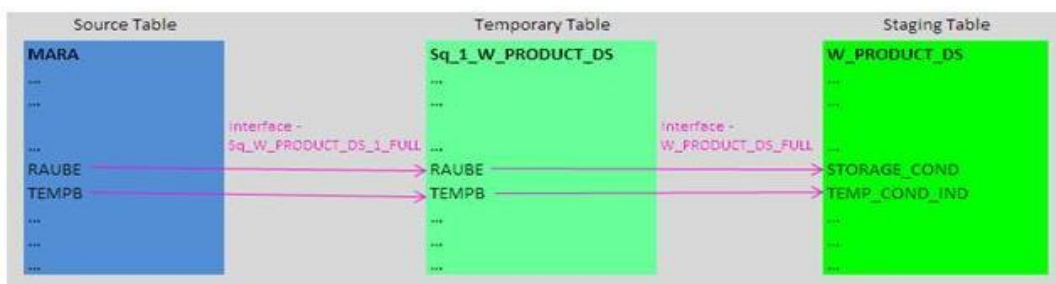
- SDE Processing, which extracts data from the source system and loads it into the staging area (for more information, see Example of Loading Data from the Staging Area into an Existing Target Table)
- SIL Processing, which extracts data from the staging area and loads it into the target table (for more information, see Example of Loading Data from the Staging Area into an Existing Target Table)

Example of Loading Data from an SAP ERP Data Packaged Source into the ODI Staging Area

Source into the ODI Staging Area This section shows how data is extracted from an existing source table into the staging area.

The diagram below shows the new E-LT mappings that you need to load the new data into the staging area, and the ODI Interfaces that you need to modify.

Figure 176. Required new mappings for loading data into the staging area

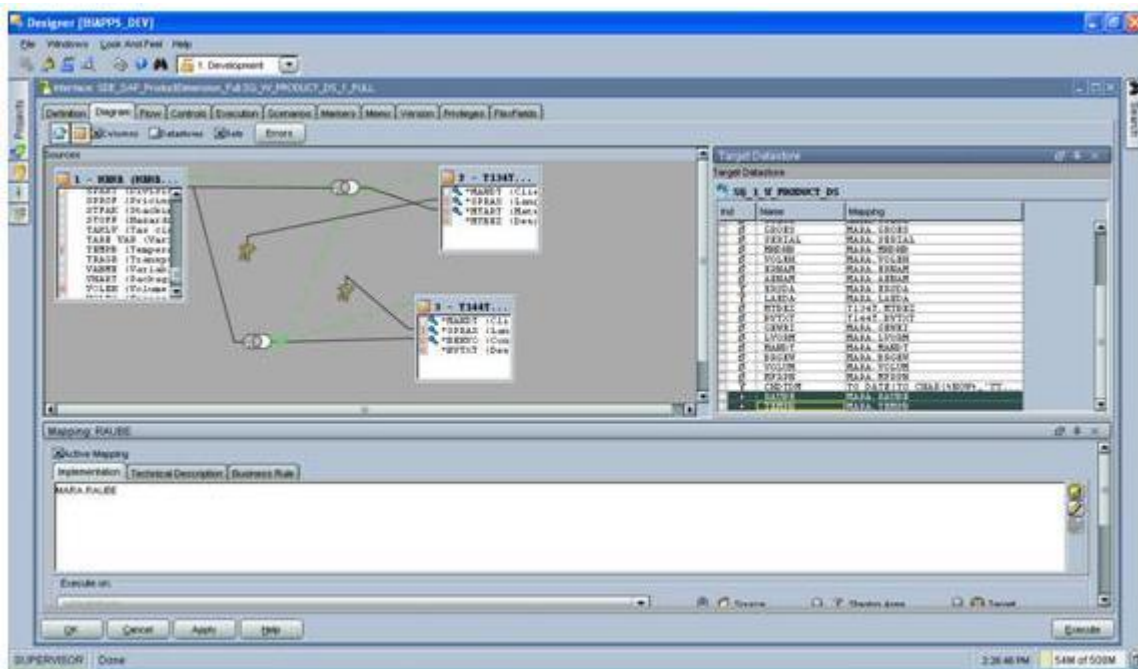


Note: The diagram above only shows the incremental interfaces.

To customize the E-LT process to load these two fields into the staging area, you need to:

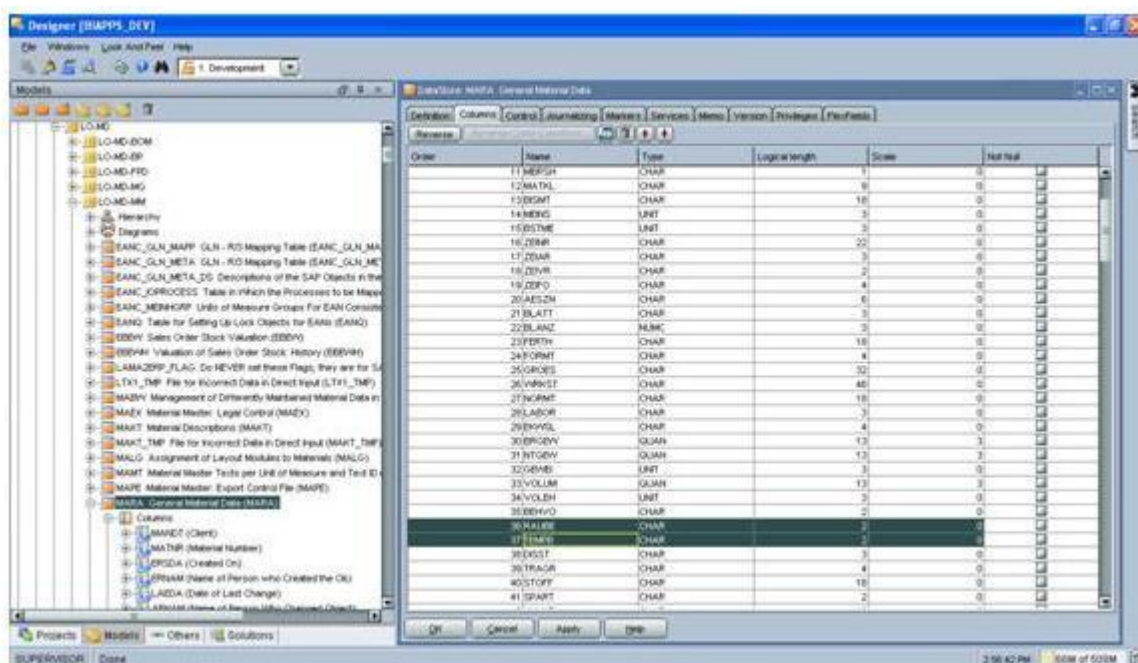
1. Extract the field MARA.RAUBE and MARA.TEMPB value from the source table MARA into the temporary table SQ_1_W_PRODUCT_D using the Interfaces SDE_SAP_ProductDimension_Full.SQ_W_PRODUCT_DS_1_FULL.

Figure 177.



- If the field does not exist in the SAP model you will need to reverse it first from SAP.

Figure 178.



2. Then, load the SQ_1_W_PRODUCT_DS.RAUBE and SQ_1_W_PRODUCT_DS.TEMPB value from the temporary table SQ_1_W_PRODUCT_DS into the STORAGE_COND and TEMP_COND_IND field in the

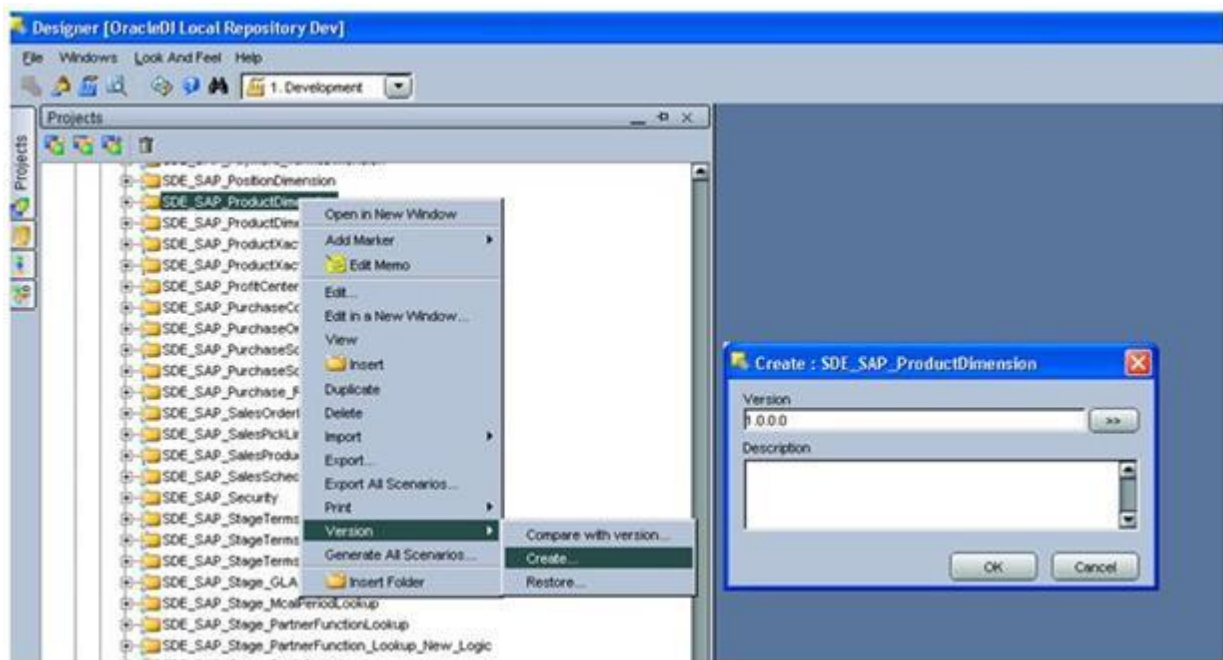
staging table W_PRODUCT_DS using the SDE_SAP_ProductDimension_Full .W_PRODUCT_DS_FULL Interface.

Note: Remember that STORAGE_COND and TEMP_COND_IND value is already extracted from the source table MARA into the temporary table SQ_1_W_PRODUCT_DS, but is not loaded into the staging table W_PRODUCT_DS.

To extract data from an SAP ERP Source:

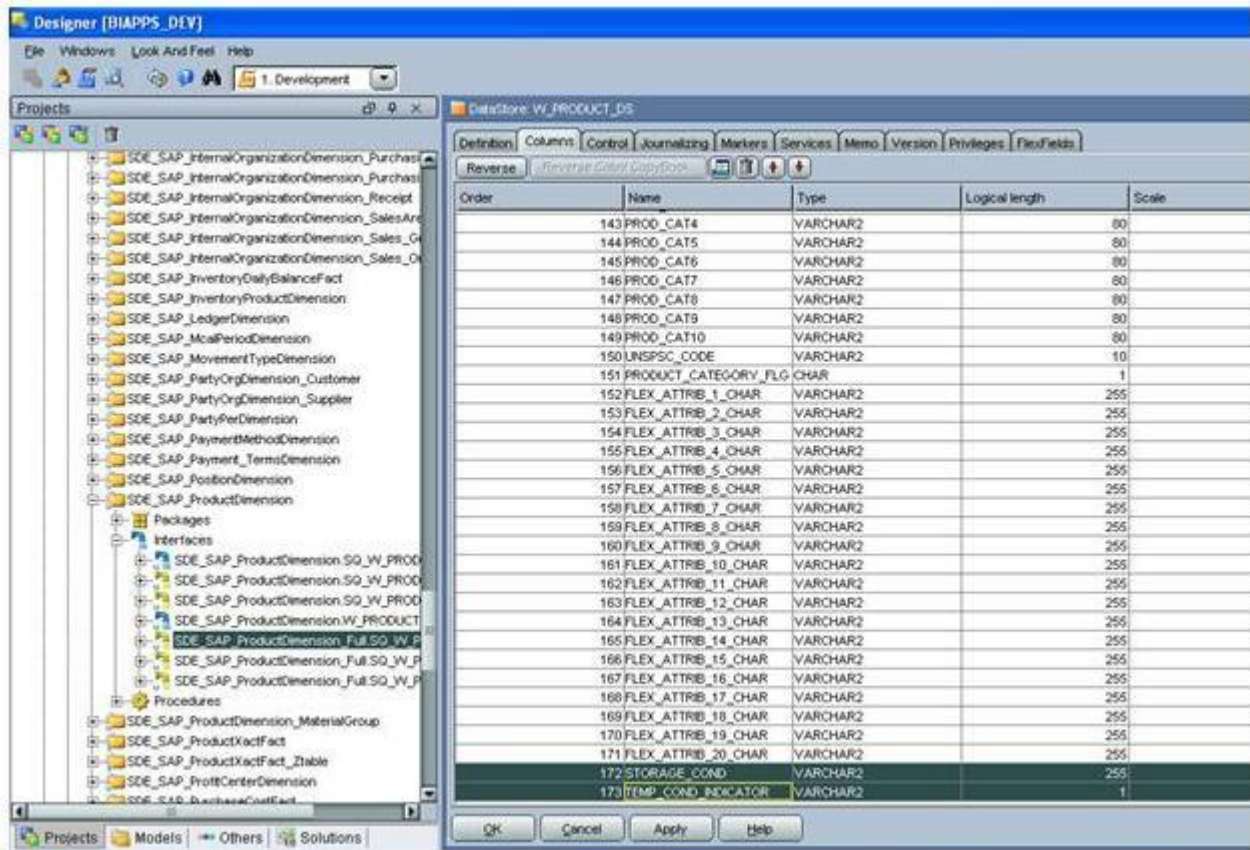
1. In ODI Designer, display the Projects view, expand the 'Oracle BI Applications 7.9.7.2'\Mappings\SDE_SAPECC6_Adaptor folder.
2. Right-click on the SDE_SAP_ProductDimension folder, and choose Version, then Create to display the Create: <Object> dialog, and specify a unique version number and optional version description.

Figure 179.



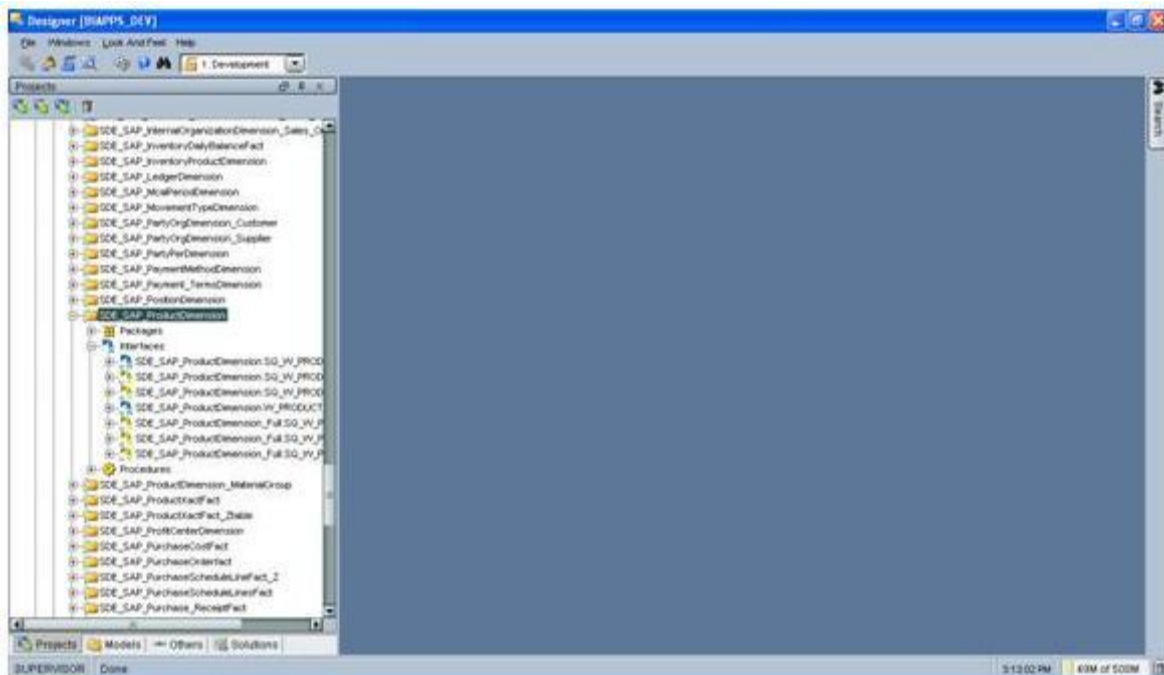
3. Display the Models view, expand the Dimension Stage folder, and edit the W_PRODUCT_DS data store to display the DataStore: <Name> dialog, and display the Columns tab.

Figure 180.



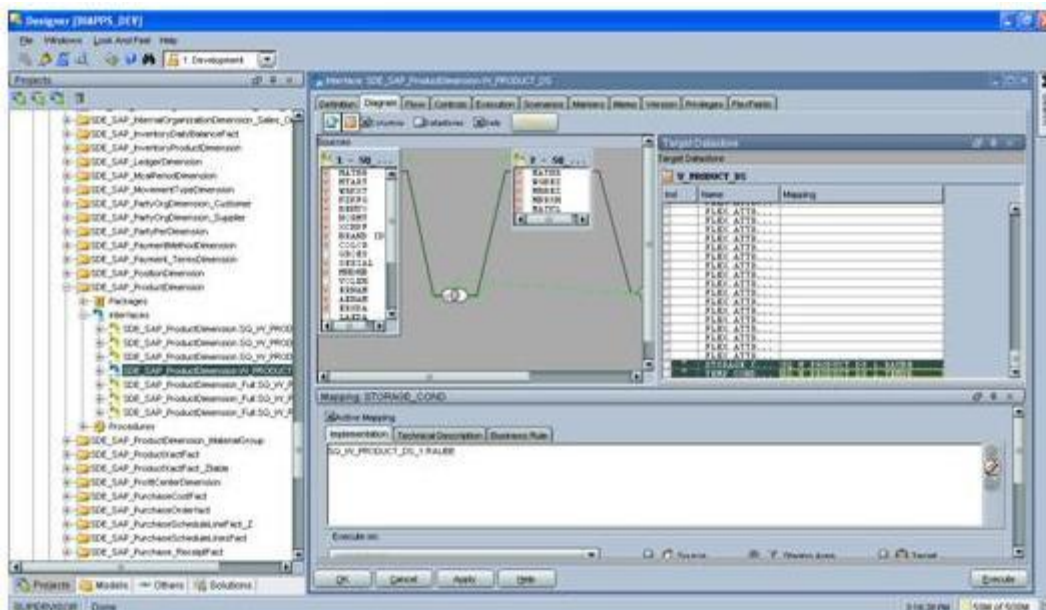
4. Create the following columns:
 - STORAGE_COND(VARCHAR2(255))
 - TEMP_COND_IND (VARCHAR2(1))
5. In the Models view, right click on the model 'Oracle BI Applications 7.9.7.2' and select Generate DDL to display the Generate DDL dialog.
 The Generate DDL option deploys the changes in the database.
6. Select the check-box in the Synchronize column next for the W_PRODUCT_DS table.
7. Click the (...) button to the right of the Generation Folder field to display the
8. Select a folder dialog, and select the \Utilities\System folder, and click OK.
9. When the Procedure: DDL <Name> dialog is displayed, click Execute. Display ODI Operator and make sure that the procedure executes successfully.
10. Display the Projects view, expand the Mappings folder, and expand the SDE_SAP_ProductDimension folder.

Figure 181.



11. Edit the Interface SDE_SAP_ProductDimension.W_PRODUCT_DS to display the Interface: SDE_SAP_ProductDimension.W_PRODUCT_DS dialog, and do the following:
 - a. Display the Diagram tab, and select the STORAGE_COND and TEMP_COND_IND field in the Target Datasource area.
 - b. Use the Launch Expression Editor icon to display the Expression Editor dialog, and use this dialog to select SQ_1_W_PRODUCT_DS.RAUBE and SQ_1_W_PRODUCT_DS.TEMPB as the value in the Implementation field respectively.
- Click OK to save the details.

Figure 182.



12. Repeat Step 11 for Interface SDE_SAP_ProductDimension.W_PRODUCT_DS_FULL.

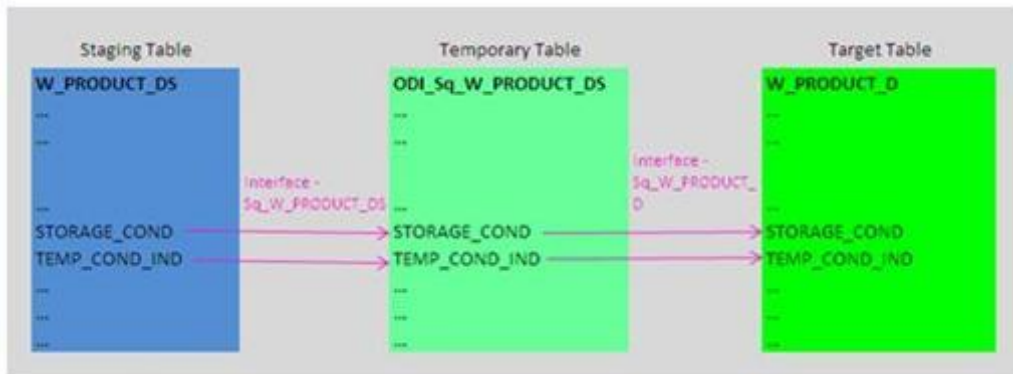
13. Regenerate scenario SDE_SAP_ProductDimension (that is, right click on the scenario and select Regenerate).

Now that you have set up the E-LT process for extracting and staging the data, you need to load the new data into the data warehouse (for more information, see Example of Loading Data from the Staging Area into an Existing Target Table)

Example of Loading Data from the Staging Area into an Existing Target Table

This section shows how data is loaded from the staging area into an existing target table. The diagram below shows the ODI Interfaces that you need to modify.

Figure 183. Required new mappings for loading data into the target table



Note: The diagram above only shows the incremental interfaces.

To customize the E-LT process to load these two fields into the target table, you need to:

- Load the `STORAGE_COND` value and `TEMP_COND_IND` value from the staging table `W_PRODUCT_DS` into the temporary table `ODI_Sq_W_PRODUCT_DS` and `ODI_Sq_W_PRODUCT_DS_FULL` using the Interfaces `SIL_ProductDimension.Sq_W_PRODUCT_DS` and `SIL_ProductDimension.Sq_W_PRODUCT_DS_FULL`.
- Load the `STORAGE_COND` value and `TEMP_COND_IND` value from the temporary table `ODI_Sq_W_PRODUCT_DS` and `ODI_Sq_W_PRODUCT_DS_FULL` into the Target table `W_PRODUCT_D` using the Interfaces `SIL_ProductDimension.PRODUCT_D` and `SIL_ProductDimension.PRODUCT_D_FULL`, and `SIL_ProductDimension.Unspecified.PRODUCT_D_UNSPC`.

To extract data from a Staging Area:

1. In ODI Designer, display the Projects view, expand the 'Oracle BI Applications 7.9.7.2'\Mappings\SILOS folder.
2. Right-click on the `SIL_ProductDimension` folder, and choose Version, then Create to display the Create: <Object> dialog, and specify a unique version number and optional version description.
3. Display the Models view, expand the Dimension Stage folder, and edit the `W_PRODUCT_D` data store to display the DataStore: <Name> dialog, and display the Columns tab.
4. Make sure that the following columns are setup:
 - `STORAGE_COND(VARCHAR2(255))`
 - `TEMP_COND_IND (VARCHAR2(1))`

5. In the Models view, right click on the model 'Oracle BI Applications 7.9.7.2' and select Generate DDL to display the Generate DDL dialog. The Generate DDL option deploys the changes in the database.
6. Select the check-box in the Synchronize column next for the W_PRODUCT_D table.
7. Click the (...) button to the right of the Generation Folder field to display the Select a folder dialog, and select the \Utilities\System folder, and click OK.
8. When the Procedure: DDL <Name> dialog is displayed, click Execute. Display ODI Operator and make sure that the procedure executes successfully.
9. Display the Projects view, expand the Mappings folder, and expand the SIL_ProductDimension folder.
10. Edit the Interface SIL_ProductDimension.Sq_W_PRODUCT_DS to display the Interface: <Name> dialog, and do the following:
 - a. Display the Diagram tab, and select the STORAGE_COND and TEMP_COND_IND field in the Target Datastore area.
 - b. Use the Launch Expression Editor icon to display the Expression Editor dialog, and use this dialog to select W_PRODUCT_DS.STORAGE_COND and W_PRODUCT_DS.TEMP_COND_IND as the value in the Implementation field.
 - c. Click OK to save the details.
11. Repeat Step 10 for the Interface Sq_W_PRODUCT_DS_FULL.
12. Edit the Interface PRODUCT_D to display the Interface dialog, and do the following:
 - a. Display the Diagram tab, and select the STORAGE_COND and TEMP_COND_IND field in the Target Datastore area.
 - b. Use the Launch Expression Editor icon to display the Expression Editor dialog, and use this dialog to select Sq_W_PRODUCT_DS.STORAGE_COND and Sq_W_PRODUCT_DS.TEMP_COND_IND as the value in the Implementation field.
 - c. Click OK to save the details.
13. Repeat Step 12 for the Interface PRODUCT_D_FULL.
14. Regenerate scenario SILOS\SIL_ProductDimension (that is, right click on the scenario and select Regenerate).

Tips for Modifying the SQ_BCI_ Interface (EBS Specific)

- A new source table should always be defined on right side of a LEFT OUTER join syntax with existing source tables. Using an INNER join or a RIGHT OUTER join can result in loss of records.
- Make sure that you define joins to match on a unique set of values. If you do not define a join that ensures a unique relationship, you might get a Cartesian product, which changes the granularity and results in duplicate errors. If it is not possible to define a unique join, then do the following:
 1. Create an inline view interface sourcing from the new table, outputting necessary columns plus a column LKP_ACTIVE. For example, you might specify the expression for LKP_ACTIVE as:
 2. IS_FIRST(ARG_GROUP(columns to be partitioned by), ARG_GROUP(columns to be ordered by))
 3. **Note:** In the above example, the IS_FIRST command and the matching filter are only needed if multiple records might be returned.

4. Bring the inline view interface into an existing interface with a filter LKP_ACTIVE=1, which guarantees that at most one record will be returned.

As a best practice, you should comment custom code that you introduce. Comments should include the developer's name and the date that the code was added.

Including a Source Table for the Change Capture Process

If you are bringing in data from a new table that was not previously included in an existing SDE package, you might need to create a change capture mapping. When a row changes in the new table, the change capture mapping marks the corresponding row in the main table as changed. Change capture processes can degrade ELT performance. Therefore, change capture processes should only be implemented if required.

ABAP Process (Consult ABAP Consultant)

Create additional entries in table ZTOBI_CONFIG for the new enhanced fields using the instructions below:

1. Use transaction code SE37, and provide the function module name FM: ZOB_I_CONFIG_TABLE_UPDATE. Click on change button for use shortcut key F6.
2. Find the code "IF NOT it_config[] IS INITIAL." In the source code. Normally it's at line no. 5346.
3. Insert the below code before line 5346 :

```
wa_config-srctable='MARA'.  
wa_config-fieldname='TEMPB'.  
  
wa_config-fieldname=' RAUBE '.  
  
wa_config-trgtable='W_PRODUCT_D'.  
APPEND wa_config TO it_config.  
CLEAR wa_config.
```

ETL Process

1. Create or modify the Z table with the additional fields from the new source table. Only key fields from the new table needs to be included.
2. Include the Ztable into interface and join with the new source table to fetch the incremental data.
3. Steps under the section " Example for Extracting Data from an SAP ERP system" needs to be repeated for mapping the fields from the new source to the staging table and target table.

Silos process (Full and Incremental load process):

1. Identify whether the columns from the new fields to be mapped to the existing target fields or new fields need to be added to the target table. In the second case follow the instruction of adding new columns in section "Example for Extracting Data from an SAP ERP system".
2. If any new field are to be added , we need to define its behavior in the datastore for target table(e.g. overwrite on change or add row on change) based on you business logic.
3. Define/Modify the target table mapping by following the instruction in "Example for Extracting Data from an SAP ERP system".

Other Types of Customizations Requiring Special Handling

This section contains the following topics:

- How to Modify Category 2 SCD Triggers
- How to Add A Dimension to an Existing Fact
- How to Add a Date Dimension to an Existing Fact

How to Modify Category 2 SCD Triggers

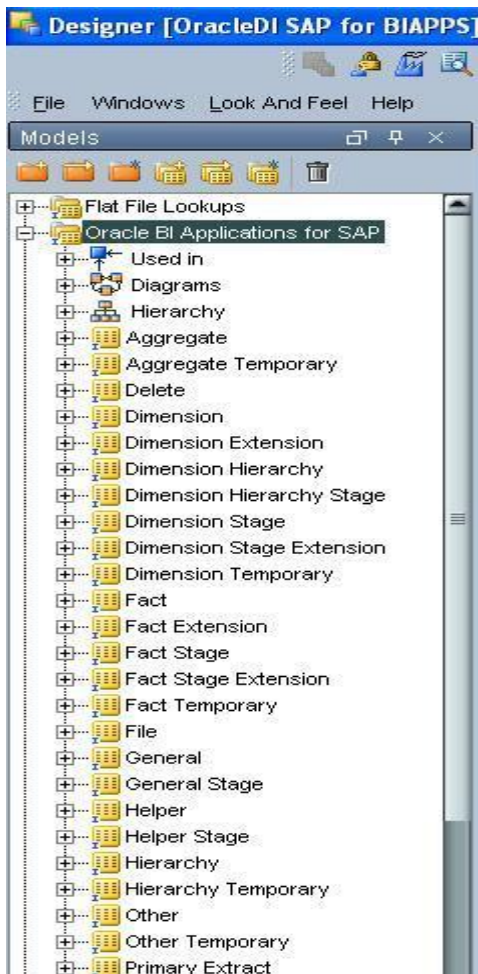
If a dimension is enabled to capture Type-II Change, you can modify the criteria that trigger a Type-II change in a dimension. Most changes in a dimension are treated as Type-I changes in that the existing column is simply overwritten with the new value. Once enabled, there are only a small number of columns that will trigger a Type-II change. You can extend the logic that triggers a Type-II change by adding additional columns to the logic that tracks Type-II changes. In addition, you can remove columns from this logic in case you do not want these types of changes to trigger a Type-II change. The Logic that tracks Type-II changes is contained in the data model.

Note: Modifying the Type-II tracking logic is the only change that you should make to shipped logic.

To modify a Category 2 SCD Trigger:

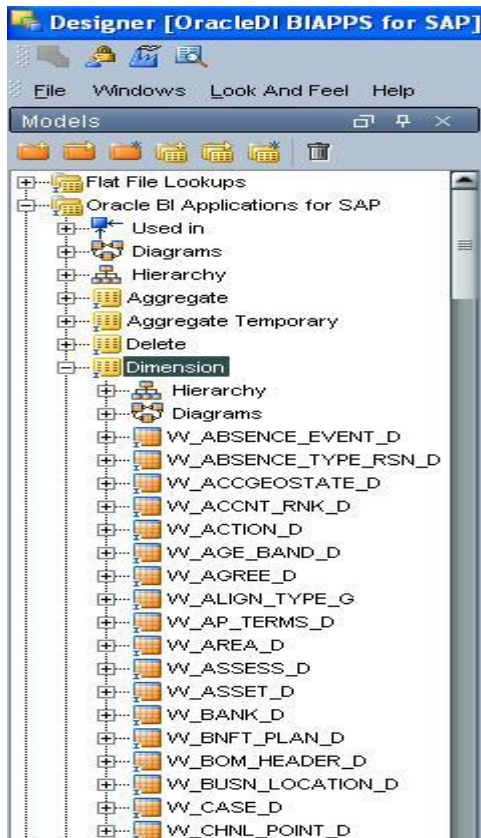
1. In ODI Designer, display the Models view, and expand the 'Oracle BI Applications 7.9.7.2' folder.

Figure 184.



2. Expand the Dimension node.

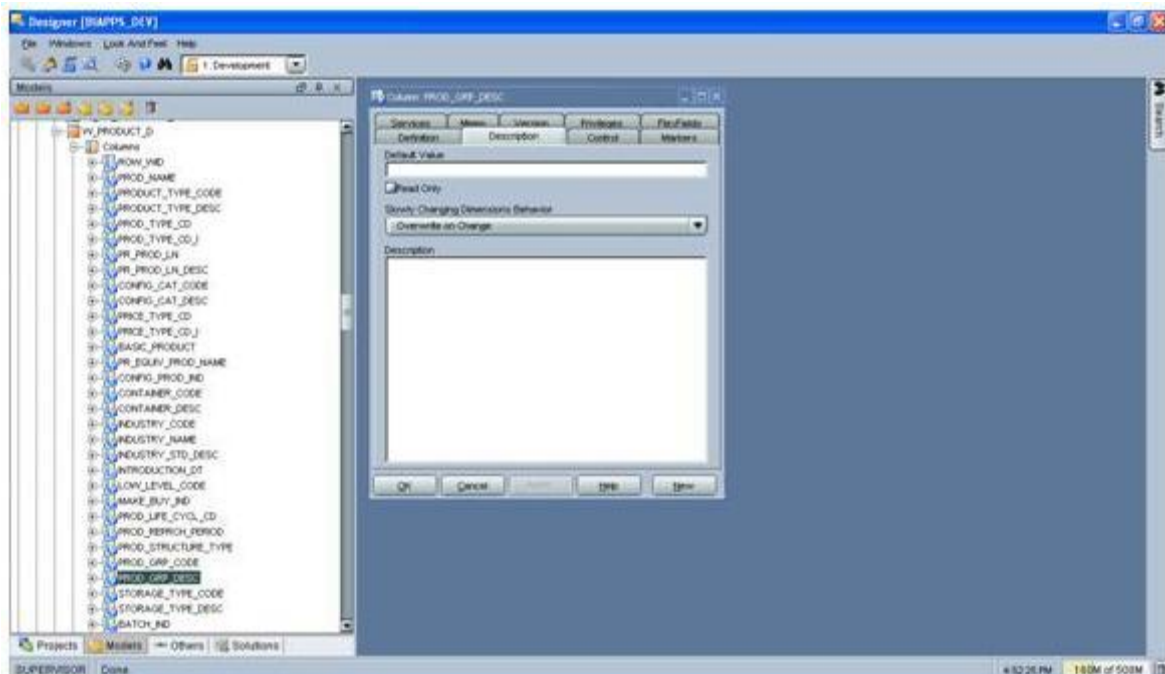
Figure 185. Dimension node



3. Select the Dimension and column on which you want to implement the Category 2 trigger.

For example, you might select the W_PRODUCT_D dimension and the PROD_GRP_DESC column.

Figure 186.

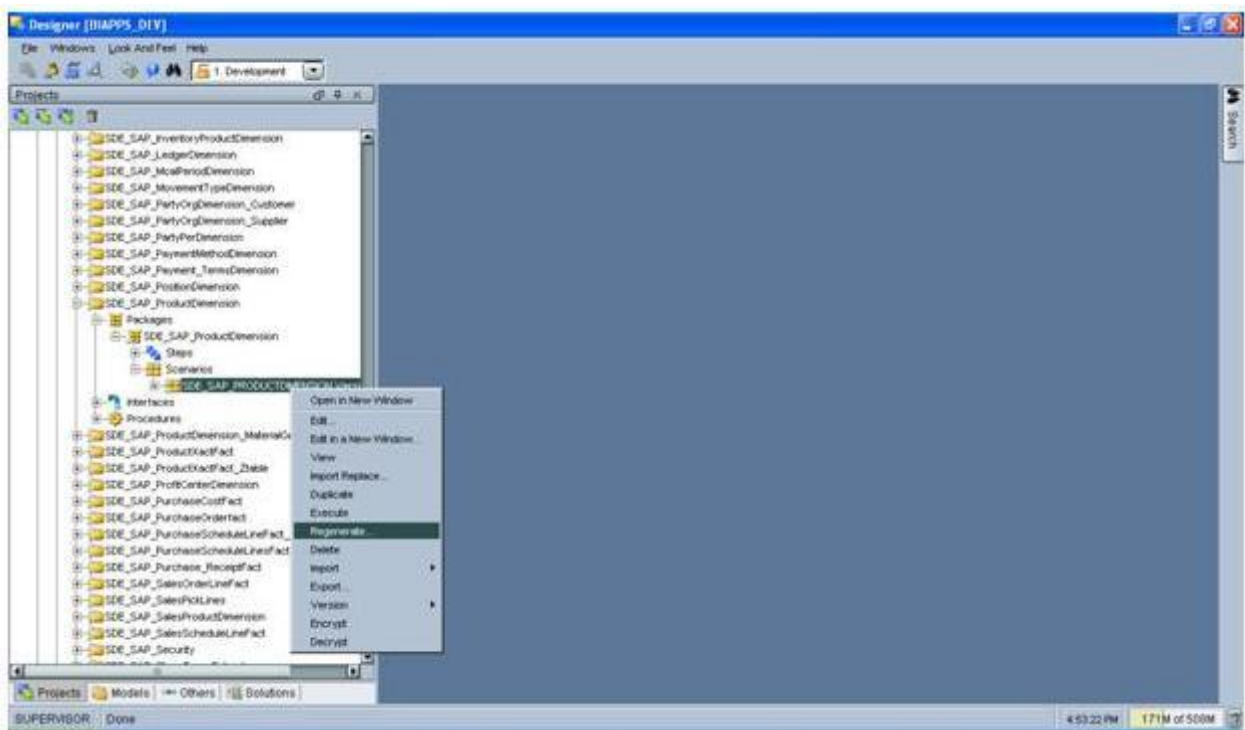


4. Double click the column name to display the Column: <Name> dialog.

5. Display the Description tab.

6. Use the Slowly Changing Dimensions Behavior drop down list to specify the behavior that you want, as follows:
 - If you want to trigger a Category 2 change, select 'Add Row on Change'.
 - If you want to trigger a Category 1 change, select 'Overwrite on Change'.
 7. Re-generate the Scenario for the SIL Dimension mapping to reflect the data model change.
- This is, right-click the SIL Dimension, then select Regenerate.

Figure 187.



For more information about customizing Slowly Changing Dimensions, see [Configuring Slowly Changing Dimensions](#).

How to Add A Dimension to an Existing Fact

This section explains how to add a dimension (pre-existing or custom) to an existing fact. It assumes that you have already built the required process to populate this dimension.

1. In ODI Designer, define the staging table column as a varchar2(80) field and named with an ID suffix.
2. Define the Oracle Business Analytics Warehouse table column as an integer and named with a _WID suffix.
3. Modify the SDE fact package to pass through the unique identifier of the dimension key. There must be a relationship between the base table and this unique identifier. It might already be stored in the base table or stored by joining to a related table. This identifier can be based on a single column or derived from multiple columns. The table below depicts various formats used to derive the

INTEGRATION_ID, which is used to identify a dimension key. The INTEGRATION_ID value should be passed to the fact staging table.

Table 44. Formats to Derive INTEGRATION_ID

Dimension	Foreign Key	When Source is SAP Application
W_AP_TERMS_D	PAY_TERMS_WID	CLIENT ID ~ TERMS OF PAYMENT KEY ~ DAY LIMIT
W_BANK_D	BANK_WID	CLIENT ID ~ COMPANY CODE ~ SHORT KEY FOR A HOUSE BANK ~ ACCOUNT ID
W_BUDGET_D	BUDGET_WID	Client ID ~ Financial Management Area ~ Fund
W_BUSN_LOCATION_D	PLANT_LOC_WID	CLIENT ~ PLANT
W_BUSN_LOCATION_D	STORAGE_LOC_WID	CLIENT~"SLOC"~PLANT~STORAGE LOCATION
W_BUSN_LOCATION_D	SALES_OFC_LOC_WID	CLIENT ~ "SALES_OFC" ~ SALES OFFICE
W_BUSN_LOCATION_D	SHIPPING_LOC_WID	CLIENT ~"SHIPPING_LOC"~ Plant ~ SHIPPING POINT
W_BUSN_LOCATION_D	COMPANY_LOC_ID	CLIENT~"COMPANY"~COMPANY
W_BUSN_LOCATION_D	RCPT_LOC_WID	Client ID~"RECIPIENT_LOC"~ Plant~Storage Location
W_BUSN_LOCATION_D	LOADING_LOC_WID	CLIENT ~ "LOADING_LOC" ~ SHIPPING POINT ~ LOADING LOCATION
W_BOM_HEADER_D	BOM_HEADER_WID	CLIENT~BILL OF MATERIAL~BOM CATEGORY~ALTERNATIVE BOM~MATERIAL
W_COST_CENTER_D	COST_CENTER_WID	CLIENT ID ~ CONTROLLING AREA ~ COST CENTER
W_CUSTOMER_ACCOUNT_D	CUSTOMER_ACCNT_WID	CLIENT ID ~ COMPANY CODE ~ CUSTOMER NUMBER
W_CUSTOMER_FIN_PROFILE_D	CUSTOMER_FIN_PROFL_WID	CLIENT ID ~ CUSTOMER NUMBER ~ CREDIT CONTROL AREA
W_EMPLOYEE_D	SALES_REP_WID	CLIENT ID ~ EMPLOYEE NUMBER
W_FREIGHT_TERMS_D	FREIGHT_TERMS_WID	CLIENT~INCOTERMS

W_GL_ACCOUNT_D	GL_ACCOUNT_WID	CLIENT ID ~ COMPANY CODE ~ CONTROLLING AREA ~ G/L ACCOUNT NUMBER ~ COST CENTER ~ PROFIT CENTER ~ BUSINESS AREA ~ SEGMENT FOR SEGMENTAL REPORTING ~ ORDER NUMBER
W_GL_ACCOUNT_D	GL_ACCOUNT_WID	CLIENT ID ~ FM AREA ~ COMMITMENT ITEM
W_INT_ORG_D	COMPANY_ORG_WID	CLIENT ID ~ 'COMPANY' ~ COMPANY CODE
W_INT_ORG_D	RECEIVABLES_ORG_WID	CLIENT ID ~ 'COMPANY' ~ COMPANY CODE
W_INT_ORG_D	PAYABLES_ORG_WID	CLIENT ID ~ 'COMPANY' ~ COMPANY CODE
W_INT_ORG_D	CTRL_AREA_ORG_WID	CLIENT ID ~ 'CONTROL AREA' ~ CONTROLLING AREA
W_INT_ORG_D	BUDG_FIN_AREA_ORG_WID	CLIENT ID ~ 'FIN_AREA' ~ FINANCIAL MANAGEMENT AREA
W_INT_ORG_D	FIN_AREA_ORG_WID	CLIENT ID ~ 'FIN_AREA' ~ FINANCIAL MANAGEMENT AREA
W_INT_ORG_D	BUSN_AREA_ORG_WID	CLIENT ID ~ 'BUSINESS_AREA' ~ BUSINESS AREA
W_INT_ORG_D	BUDG_BUSN_AREA_ORG_WID	CLIENT ~ 'FUND' ~ FM_AREA ~ FUND
W_INT_ORG_D	BUDG_CTRL_AREA_ORG_WID	Client ID ~ 'BUDGET' ~ Financial Management Area ~ Funds Center
W_INT_ORG_D	OPERATING_UNIT_ORG_WID	CLIENT ID ~ 'COMPANY' ~ COMPANY CODE
W_INT_ORG_D	OPERATING_UNIT_ORG_WID	CLIENT ID ~ 'PURCH_ORG' ~ PURCHASING ORG
W_INT_ORG_D	OPERATING_UNIT_ORG_WID	CLIENT ID ~ 'SALES_ORG' ~ SALES_ORG
W_INT_ORG_D	PURCH_ORG_WID	CLIENT~"PURCH_GROUP_ORG"~PURCHASE GROUP
W_INT_ORG_D	INVENTORY_ORG_WID	CLIENT~"INVT_ORG"~PLANT
W_INT_ORG_D	EXPENDITURE_ORG_WID	CLIENT ID ~ 'COMPANY' ~ COMPANY CODE

W_INT_ORG_D	PAYABLES_ORG_WID	CLIENT ID ~ 'COMPANY' ~ COMPANY CODE
W_INT_ORG_D	RQSTN_ORG_WID	CLIENT~"PURCHASE_ORG"~PURCHASIN G ORG
W_INT_ORG_D	SALES_GROUP_ORG_WID	CLIENT ~ "SALES_GROUP" ~ SALES GROUP
W_INT_ORG_D	SALES_ORG_WID	CLIENT ~ "SALES_ORG" ~ SALES ORGANIZATION
W_INVENTORY_PRODUCT_D	INVENTORY_PROD_WID	CLIENT ID ~ MATERIAL NUMBER ~ PLANT ~ STORAGE LOCATION
W_INVENTORY_PRODUCT_D	INVENTORY_PROD_WID	CLIENT ID ~ MATERIAL NUMBER ~ PLANT
W_LEDGER_D	LEDGER_WID	CLIENT ID ~ COMPANY CODE ~ LEDGER
W_MCAL_CAL_D	MCAL_CAL_WID	CLIENT ID ~ FISCAL YEAR VARIANT
W_MVMNT_TYPE_D	MVMNT_TYPE_WID	CLIENT~Movement Type~Special Stock Indicator~Movement Indicator~Receipt Indicator~Consumption Posting
W_PARTY_ORG_D	SUPPLIER_WID	CLIENT~"SPLR_ORG"~VENDOR ID
W_PARTY_ORG_D	CUSTOMER_WID	CLIENT ID ~ 'CUST_ORG' ~ CUSTOMER NUMBER
W_PAYMENT_METHOD_D	PAY_METHOD_WID	CLIENT ID ~ COUNTRY KEY ~ PAYMENT METHOD
W_PAYMENT_TERMS_D	PAY_TERMS_WID	CLIENT ID ~ TERMS OF PAYMENT KEY ~ DAY LIMIT
W_PRODUCT_D	PRODUCT_WID	CLIENT ID ~ MATERIAL NUMBER
W_PRODUCT_D	PRODUCT_WID	Client ID~"NON_INVT"~ material group
W_PROFIT_CENTER_D	PROFIT_CENTER_WID	CLIENT ID ~ CONTROLLING AREA ~ PROFIT CENTER
W_SALES_PRODUCT_D	SALES_PROD_WID	CLIENT ID ~ MATERIAL NUMBER ~ SALES ORG ~ DISTRIBUTION CHANNEL
W_SUPPLIER_ACCOUNT_D	SPLR_ACCT_WID	CLIENT ID ~ COMPANY CODE ~ VENDOR ID
W_SUPPLIER_PRODUCT_D	SUPPLIER_PROD_WID	CLIENT ID ~ VENDOR ID ~ MATERIAL NUMBER ~ PURCHASE ORG ~ PLANT ~ INFO RECORD NUM ~ INFO CATEGORY

W_TAX_TYPE_D	TAX_TYPE_WID	CLIENT ID ~ COUNTRY ~ SALES TAX CODE

4. To resolve the value of the new WID column in the SIL package, do the following:
 - d. Create an Inline View Interface X_LKP_W_Dim_D for the dimension table.
 - e. In the 'SQ_XXXX' interfaces, use an outer join to join the Fact table to the Inline View of the dimension table using INTEGRATION_ID, DATASOURCE_NUM_ID. If the dimension is a slowly changing dimension, the fact table's standard or 'canonical' date should be used as well as the join condition, even if the dimension has not been enabled to capture Category 2 changes.
5. Use ROW_WID of the dimension table as the WID value of the fact table. Add logic to default the WID value to 0 if no record is returned from the join.
6. Save the changes to the SDE and SIL packages.
7. Make sure that the new dimension extract and load packages are already part of the Master packages, and that it is loaded before the Fact load.

How to Add a Date Dimension to an Existing Fact

If you want to add a date dimension to a fact table, you pass the date through the SIL Interfaces to the target table.

To add a date dimension to an existing fact:

1. Add a new field called DT_WID to the fact table, as follows:
 - a. In ODI Designer, display the Models view, and open the Oracle BI Applications for SAP folder.
 - b. Expand the Fact folder, and double-click the fact table to display the DataStore:<Name> dialog. For example, you might edit the W_GL_REVN_F table.
 - c. Click the Add Column icon to add a new column to the list, and change the default column name to DT_WID, and change the Type to DATE.
 - d. In the Models view, right click on the model 'Oracle BI Applications 7.9.7.2' and select Generate DDL to display the Generate DDL dialog. The Generate DDL option deploys the changes in the database.
 - e. Select the check-box in the Synchronize column next for the table that you added.
 - f. Click the (...) button to the right of the Generation Folder field to display the Select a folder dialog, and select the \Utilities\System folder, and click OK.
 - g. When the Procedure: DDL <Name> dialog is displayed, click Execute. Display ODI Operator and make sure that the procedure executes successfully.
2. Modify the new DT_WID column in the SIL Interface, as follows:
 - a. In ODI Designer, display the Projects view, and open the SIL package.
 - b. For example, you added a new date field to the W_GL_REVN_F table, you might want to edit the SIL_GLRevenueFact\SIL_GLRevenueFact Package.
 - c. Edit the last Interface located at the end of the flow. For example, if you edit the SIL_GLRevenueFact\SIL_GLRevenueFact Package, you edit the 'Run GL_REVN_F_FULL' Interface.

- d. Display the Diagram tab.
- e. In the Target Datastore area, select the DT_WID column that you created in step 1.
- f. In the Mapping pane below, add the following expression in the Implementation field:
- g. `COALESCE(IIF(ISNULL(Sq_Fact_Table.Newly_added_DT),NULL,TO_INTEGER(TO_CHAR_FORMAT(Sq_Fact_Table.Newly_added_DT,'YYYYMMDD'))),0)`
- h. Save the details.
- i. Regenerate the interface.

Category 2 Customizations: Adding Additional Tables

Category 2 customizations use pre-packaged adapters to add new fact or dimension tables to the Oracle Business Analytics Warehouse.

This section contains the following topics:

- About Creating New Dimension or Fact Tables
- Creating Custom ODI Master Packages
- Adding a New Dimension to the Oracle Business Analytics Warehouse
- Adding a New Fact Table to the Oracle Business Analytics Warehouse
- Adding a New Dimension Table for a New Fact Table in the Oracle Business Analytics Warehouse

About Creating New Dimension or Fact Tables

This section relates to building entirely new tables that will be loaded with data from a source table that is not already extracted from. For example, you might want to create a new Project dimension table. In this case, you create new dimension and staging tables as well as new extract and load ELT mappings. When creating a new custom table, use the prefix WC_ to help distinguish custom tables from tables provided by Oracle as well as to avoid naming conflicts in case Oracle later releases a table with a similar name. For example, for your Project dimension you might create a WC_PROJECT_DS and a WC_PROJECT_D table. When you create a new dimension or fact table, use the required system columns that are part of each of the Oracle Business Analytics Warehouse tables to maintain consistency and enable you to reference existing table structures. When you create a new table, you need to define the table and indices in ODI Designer Models area first. The destination model for the Oracle Business Analytics Warehouse is 'Oracle BI Applications 7.9.7.2'.

About the Main Required Columns

For custom staging tables, the following columns are required:

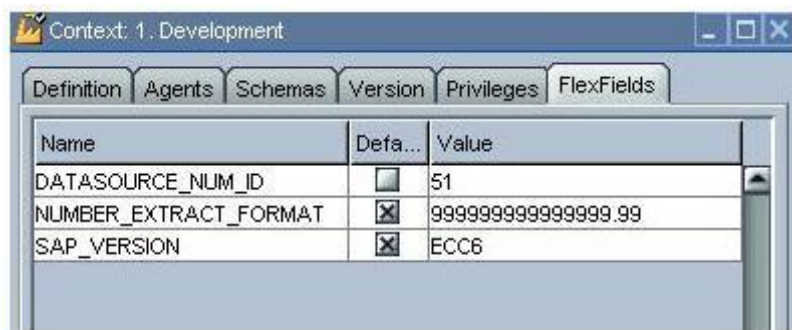
- INTEGRATION_ID. Stores the primary key or the unique identifier of a record as in the source table.
- DATASOURCE_NUM_ID. Stores the data source from which the data is extracted.
- For dimension and fact tables, the required columns are the INTEGRATION_ID and DATASOURCE_NUM_ID columns as well as the following:
- ROW_WID. A sequence number generated during the ELT process, which is used as a unique identifier for the Oracle Business Analytics Warehouse.
- ELT_PROC_WID. Stores the ID of the ELT process information. The details of the ELT process are stored in the W_ELT_RUN_S table on the Oracle Business Analytics Warehouse side.

About the DATASOURCE_NUM_ID Column

The tables in the Oracle Business Analytics Warehouse schema have DATASOURCE_NUM_ID as part of their unique user key. While the transactional application normally ensures that a primary key is unique, it is possible that a primary key is duplicated between transactional systems. To avoid problems when loading this data into the data warehouse, uniqueness is ensured by including the DATASOURCE_NUM_ID as part of the user key. This means that the rows can be loaded in the same data warehouse tables from different sources if this column is given a different value for each data source. The DATASOURCE_NUM_ID is maintained by a Flex Field value that you can specify using ODI Topology Manager. The out-of-the-box Flex Field value is '1', but you change this to '51' (for SAP Source Application ECC6.0) and for SAP 4.6C (DATASOURCE_NUM_ID is 50) as part of the installation and setup process (for more information, see "How to set up the Data Source Number").

For SAP ECC6.0

Figure 188.

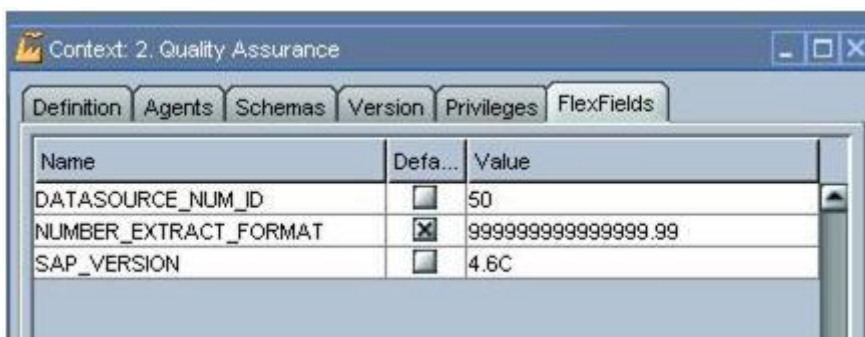


The screenshot shows the 'FlexFields' tab in the ODI Topology Manager for 'Context: 1. Development'. It displays a table with three rows: DATASOURCE_NUM_ID, NUMBER_EXTRACT_FORMAT, and SAP_VERSION. The DATASOURCE_NUM_ID is set to 51, NUMBER_EXTRACT_FORMAT is checked and set to 9999999999999999.99, and SAP_VERSION is checked and set to ECC6.

Name	Defa...	Value
DATASOURCE_NUM_ID	☐	51
NUMBER_EXTRACT_FORMAT	☒	9999999999999999.99
SAP_VERSION	☒	ECC6

For SAP 4.6C

Figure 189



The screenshot shows the 'FlexFields' tab in the ODI Topology Manager for 'Context: 2. Quality Assurance'. It displays a table with three rows: DATASOURCE_NUM_ID, NUMBER_EXTRACT_FORMAT, and SAP_VERSION. The DATASOURCE_NUM_ID is set to 50, NUMBER_EXTRACT_FORMAT is checked and set to 9999999999999999.99, and SAP_VERSION is set to 4.6C.

Name	Defa...	Value
DATASOURCE_NUM_ID	☐	50
NUMBER_EXTRACT_FORMAT	☒	9999999999999999.99
SAP_VERSION	☐	4.6C

Creating Custom ODI Master Packages

Creating a custom ODI Master Package enables you to create a custom Subject Area. **Note:** If you customize an existing package or interface, then you do not need to modify the master packages since the modified package is already being executed. If you create a new customized package, you need to include the customized package in the master packages to be executed during a regular load. To include a package in the master package, it should be called in either a level 3 subject area package or a level 4 task group package. To create one of these level 3 or level 4 packages, you can use of the package templates:

- 3_Master_PLP_<App>_<Subj>
- 3_Master_SDE_Facts_<App>_<Subj>
- 3_Master_SIL_Facts_<App>_<Subj>
- 4_Master_PLP_TG_<Table>

- Category 2 Customizations: Adding Additional Tables
- 4_Master_SDE_Dimensions_TG_<Dim>
- 4_Master_SIL_Dimensions_TG_<Dim>
- 4_Master_SDE_Facts_TG_<Fact>
- 4_Master_SIL_Facts_TG_<Fact>

Notes on Using the Package Templates

- You need to make a copy of the appropriate template that you wish to use for your customized package. If your customized package is for a dimension table and is for an existing category, you just need to append it to their respective level 3 SDE/SIL dimension category package.
- If your customized package is for a dimension table and needs a new category, create 3_Master_SDE_Dimensions_Custom and 3_Master_SIL_Dimensions_Custom. There is no template package in the master package folder for these packages. This is because there are no other steps in these packages aside from the package that you need to execute. Use the 'Insert package' option in ODI for creating these packages. Put the step before the 'wait for child session' step in these packages.
- If the customized package is a new PLP dimensions package, you need to append this in 3_Master_PLP_Dimensions package. There is no need to create a new level 3 PLP dimension master package.
- If your customized package is for a Fact or PLP table and is for an existing subject area, you just need to append this package to their appropriate level 3 packages.
- If your customized package is for a Fact or PLP table and is for a new subject area, use the other appropriate level 3 templates. There are steps in this template for refresh and evaluate of the variable OBI_EXEC_PACKAGE. You need to append the steps you need to execute after the evaluate variable step. Use the 'ok' connection from the 'Execute Package?' step to your customized step. Change <App> with CUSTOM and <Subj> with your own customized subject area name. For new subject areas, you need to use Oracle BI Applications Configuration Manager to add the subject area and enable it for execution in its appropriate subject area. Before you can use Oracle BI Applications Configuration Manager to add a new subject area, use the 'Add Custom Subject Area' and 'Refresh Package Module Data' packages located in the Projects folder in ODI Designer (in \Utilities_and_Execution\Utilities\User\Packages\).
- After creating the level 3 package, you need to generate the scenario for this package. Add a scenario execution step in their appropriate level 2 master package, as follows.
- After creating the level 3 package, you need to generate the scenario for this package. Create the following level 2 master packages to execute these customized packages: 2_Master_PLP_CUSTOM, 2_Master_SDE_Facts_CUSTOM, and 2_Master_SIL_Facts_CUSTOM. Add a scenario execution step in their appropriate customized level 2 master packages. Set the agent code in these steps to "WORKFLOW". Set the appropriate "Synchronous / Asynchronous" setting for the steps. Add an "OdiWaitForChildSession" step if necessary. Generate the scenarios for these customized level 2 packages, add a scenario step to: 1_Master_PLP, 1_Master_SDE_Facts, and 1_Master_SIL_Facts.
- For level 4 task group packages, this is required if there are multiple customized package loading the same target table or you are using temporary table.
- If your customized package is for an existing task group, just append it to their appropriate level 4 task group package.

- If your customized package is for a dimension table and require a new task group package, use either 4_Master_SDE_Dimensions_TG_<Dim> or 4_Master_SIL_Dimensions_TG_<Dim> template.
- In this template, there are steps for refresh and evaluate of OBI_START_TASK_GROUP, a refresh of IS_INCREMENTAL variable and an Update Load Date procedure call. These steps are mandatory for this type of task group. The other steps are Drop Indexes, Truncate Table, and Create Indexes. These steps are optional if you want such process to be handled in the task group level. The parameters for these steps should be set as follows:
- For drop index step, set the value in additional variable tab for OBI_MANAGE_TABLE_MASK and OBI_INDEX_DROP_ELT.
- For truncate table step, set the value in additional variable tab for OBI_MANAGE_TABLE_MASK, OBI_TRUNCATE_TABLE, and OBI_TRUNCATE_TASK_GROUP.
- For create index step, set the value in additional variable tab for OBI_MANAGE_TABLE_MASK and OBI_INDEX_CREATE_ELT.
- Your customized package should be added after the truncate table step and before create indexes step. Use an OdiWaitForChildSession step if necessary for asynchronous jobs.
- If your customized package is for a Fact or PLP table and require a new task group package, then use 4_Master_SDE_Facts_TG_<Fact>, 4_Master_SIL_Facts_<Fact>, or 4_Master_PLP_TG_<Table> template.
- The steps in these templates are similar to those in the dimensions task group. The only difference is it has steps for refresh and evaluate of OBI_START_PACKAGE instead of OBI_START_TASK_GROUP variable. Set the other steps the same way as the dimensions task group. These customized level 4 task group packages should be added to their appropriate level 3 master packages.

Note: For all packages modified, you need to regenerate the scenario.

Additional Information About Customizing

This section contains additional miscellaneous information about customization in Oracle Business Intelligence Applications.

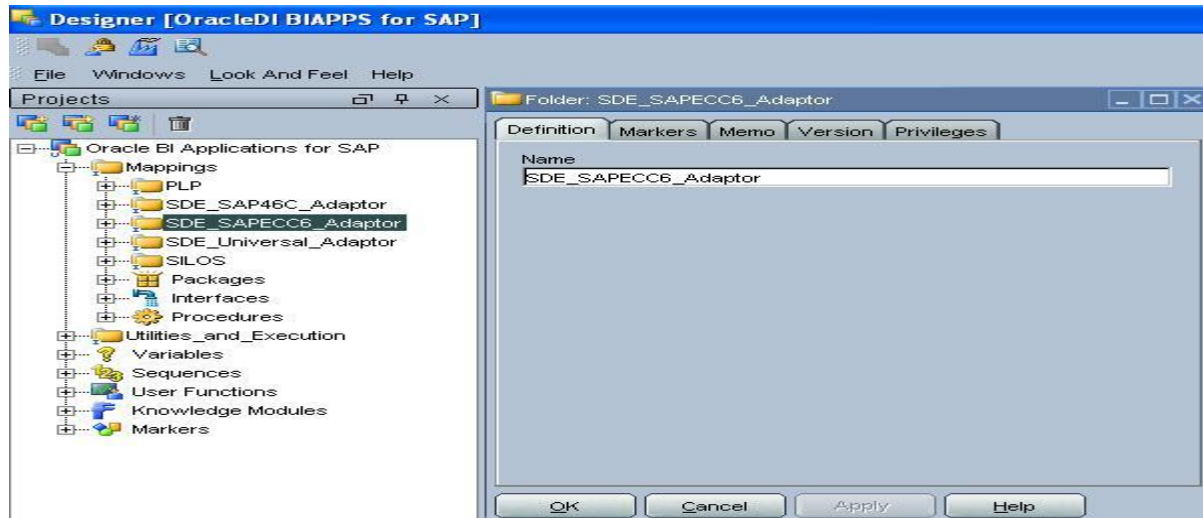
About Table Definitions in ODI

When you import table definitions from external data sources to an ODI work repository, make sure that the 'Technology' is set to SAP.

To import table definitions into an ODI Work Repository:

1. In ODI Designer, display the Models view, and double-click the folder that holds the table definitions (for example, 'Oracle eBusiness Suite 11.5.10') to display the Model: <Name> dialog.

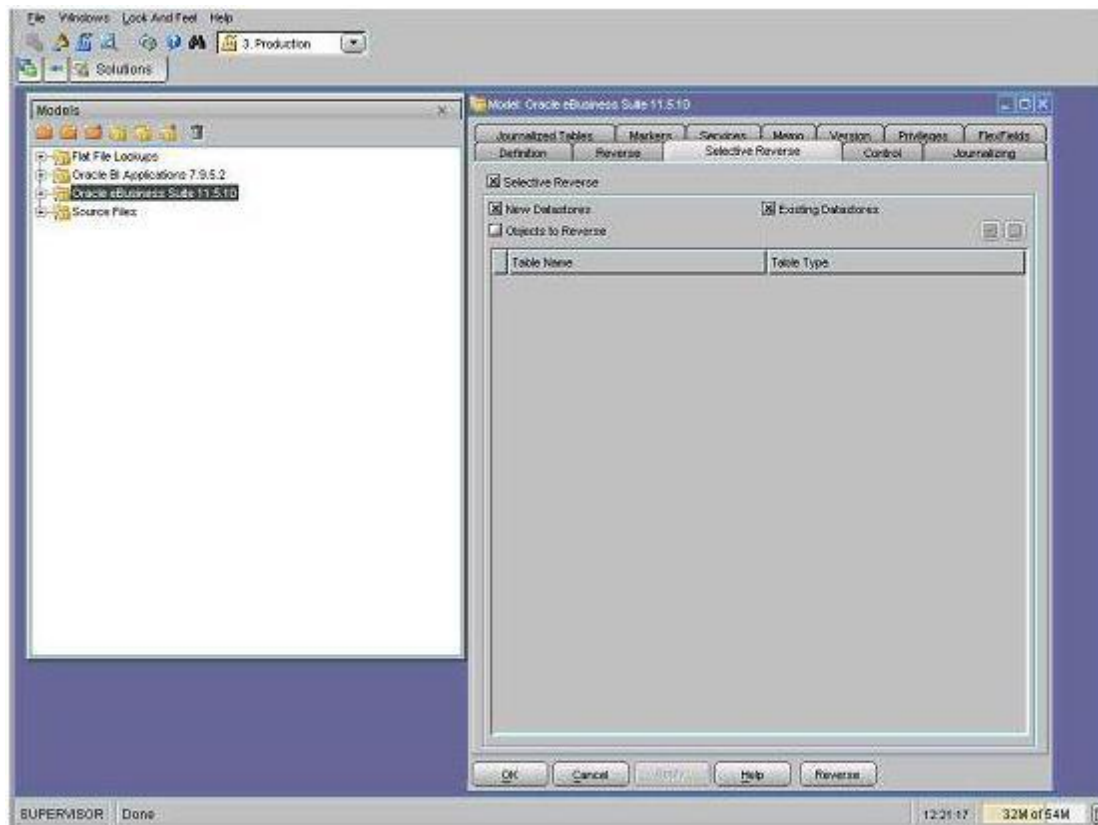
Figure 190.



2. Display the Selective Reverse tab.

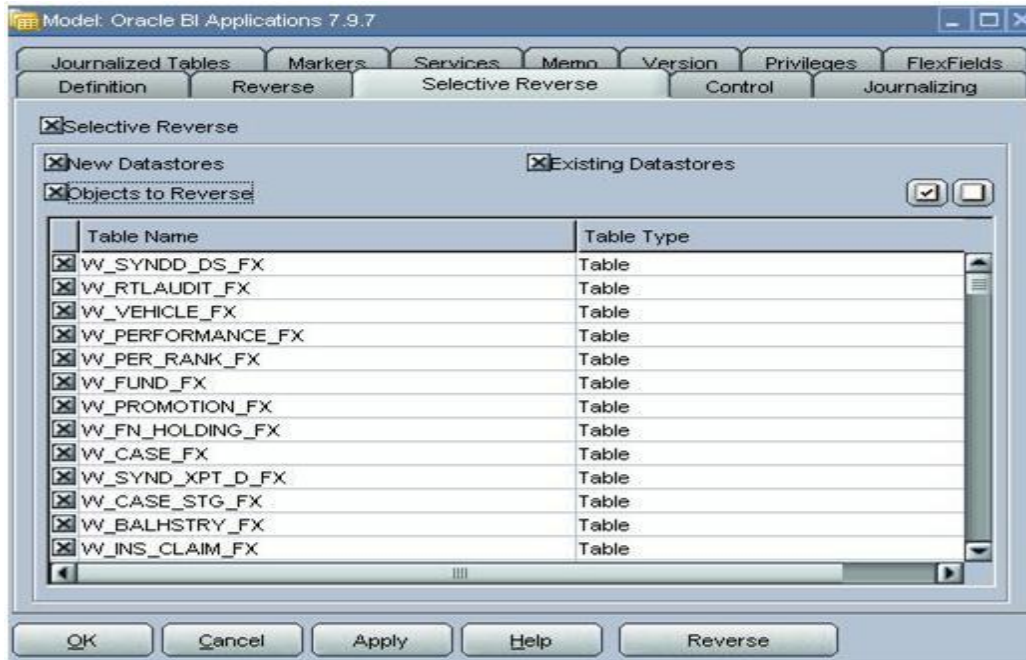
Category 2 Customizations: Adding Additional Tables

Figure 191.



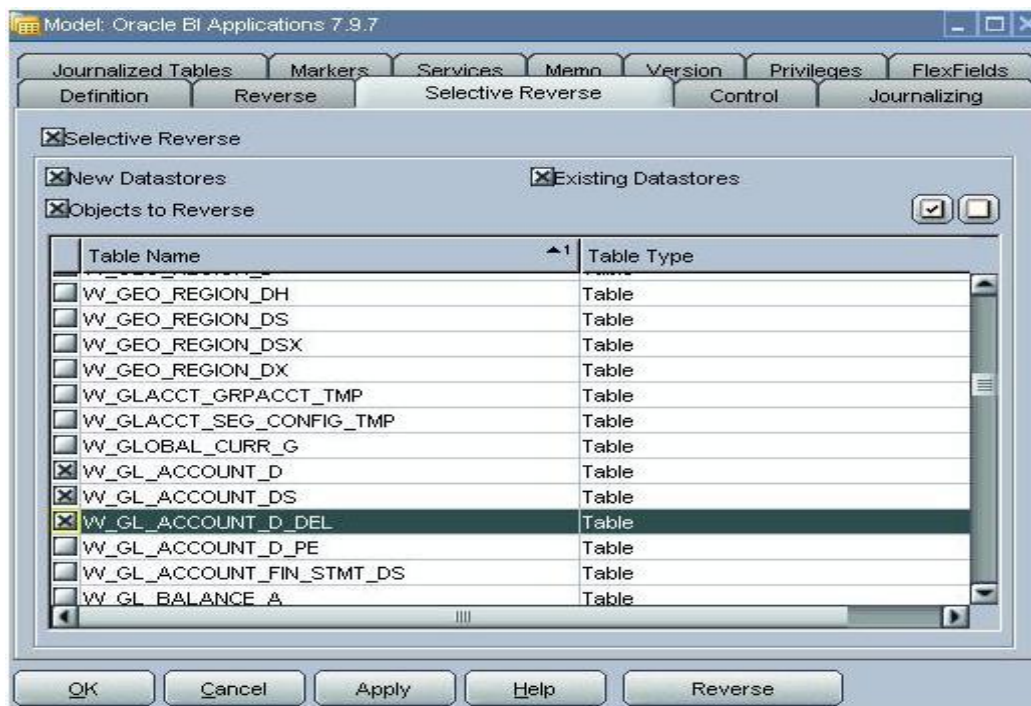
3. Select the Objects To Reverse check box.

Figure 192.



4. Select the objects (Table/View/Synonym) that you want to import into the ODI Repository.

Figure 193.

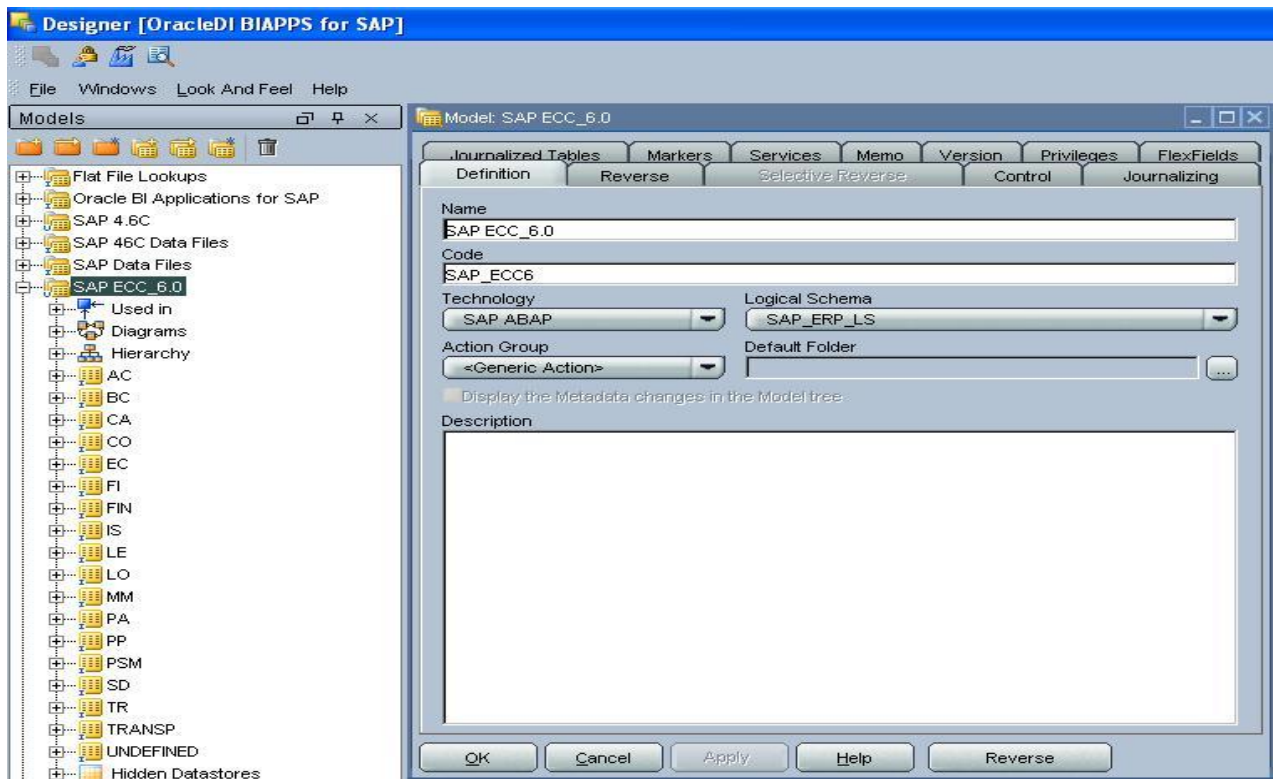


5. Click OK to reverse-engineer the objects.

Adding additional table in SAP technology:

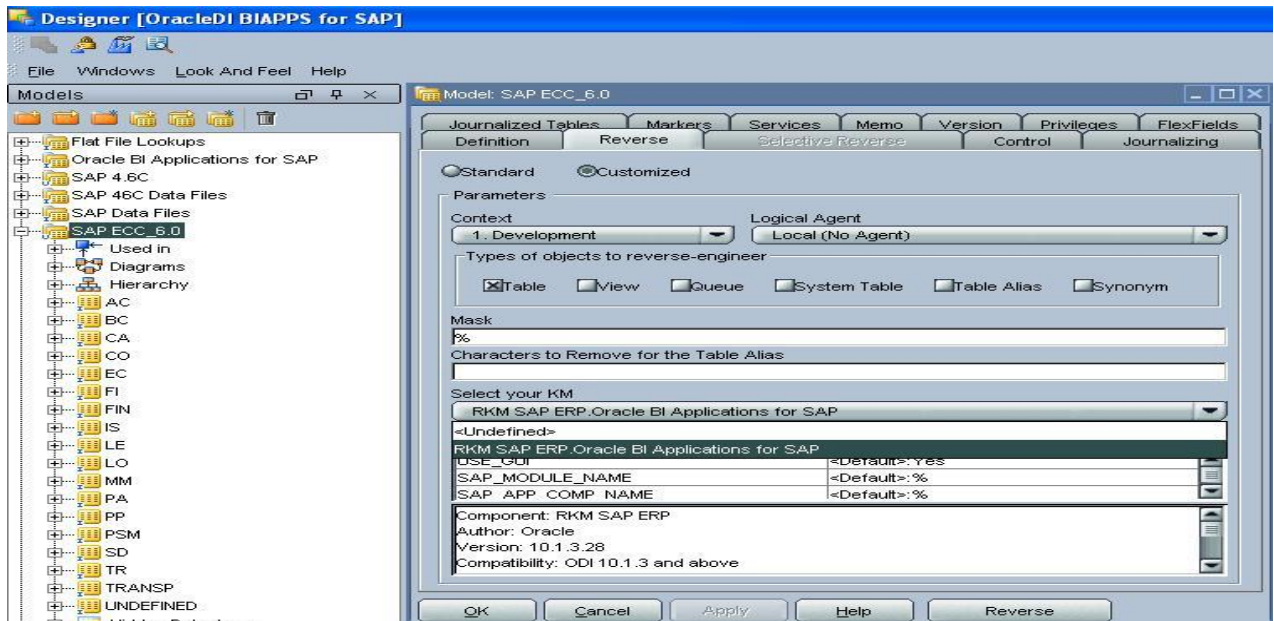
1. In ODI Designer, display the Models view, and double-click the folder that holds the table definitions (for example, SAPECC_6.0') to display the Model dialog.

Figure 194.



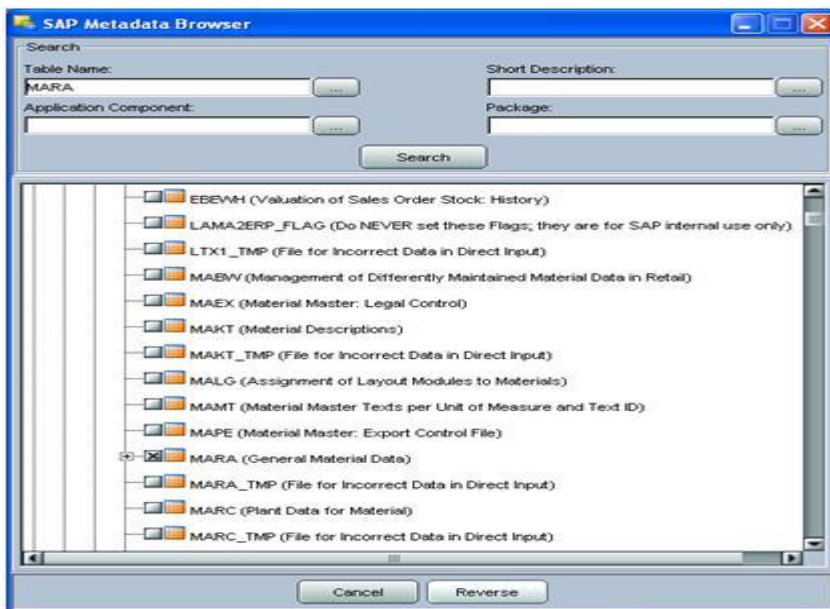
2. Display the reverse table and select Customized reverse radio button. Select Your KM as RKM_SAP_ERP.

Figure 195.



3. Two options are available for USE_GUI and Yes and No.
4. Click on the reverse button to open metadata browser. Select your table and Click on reverse.

Figure 196.



5. The table will be reversed in the SAP model.

About the Update Strategy

For loading new fact and dimension tables, design a custom process on the source side to detect the new and modified records.

The SDE process should be designed to load only the changed data (new and modified). If the data is loaded without the incremental process, the data that was previously loaded will be erroneously updated again. For example, the logic in the preconfigured SIL mappings looks up the destination tables based on the INTEGRATION_ID and DATASOURCE_NUM_ID and returns the ROW_WID if the combination exists, in which case it updates the record. If the lookup returns null, it inserts the record instead. In some cases, last update date(s) stored in target tables are also compared in addition to the columns specified above to determine insert or update. Look at the similar mappings in the preconfigured folder for more details.

ODI Steps:

1. Reverse the required Source tables by following the steps mentioned in this document earlier.
2. Create the interfaces based on the business logic that is defined by the functional team.
3. Define temporary and final interfaces based on the requirement.
4. Create SDE package to populate the staging warehouse table.
5. Create SILOS by looking up on the dimensions for all the Fact table wids that are present in the warehouse table.

Incremental Process

6. Create new Z program and Z table in SAP. The Z program will populate the incremental data in the SAP Z table for this new dimension/Fact table based on the date interval passed to it as input.
7. Include this Z table into the interfaces for incremental load as defined in the section "Example for Extracting data from SAP_ERP sources."
8. The Incremental load process work on Change capture mechanism of SAP and uses change capture tables CDHDR and CDPOS.

Creating a new Program for Delta

1. Updating Z_Config table
2. Below is the procedure for creating a new Z program for Delta for Product Dimension , Source Table is MARA & Source primary key field is MATNR.
3. Create entries corresponding to the mapped fields in (MARA-MATNR) in table ZTOBI_CONFIG.
4. ZTOBI_CONFIG helps in filtering the fields for change capture which are applicable for Delta change for a Warehouse table like W_PRODUCT_DS in this case.
5. Use transaction SE37 to change Function Module ZOBI_CONFIG_TABLE_UPDATE in change mode
6. Find code "IF NOT it_config[] IS INITIAL." In the source code. Normally it's at line no. 5346.
7. Insert the below code before line 5346 :
8. wa_config-srctable = 'MARA'.
9. wa_config-fieldname = 'MATNR'.
10. wa_config-trgttable = (Put the target table name in single quotes & in upper case , e.g 'W_PRODUCT_DS'.)
11. APPEND wa_config TO it_config.
12. CLEAR wa_config.
13. Activate and Execute.

Create Z table to capture delta in SAP

1. Use transaction SE11 to create a Z table which will capture the delta.
2. Provide the table name and click 'CREATE' (e.g. ZTOBI_INCR_MARA).
3. Provide the relevant info like ATTRIBUTES, FIELDS(MARA) & TECHNICAL SETTINGS.
4. Save & Activate

Create Z function module

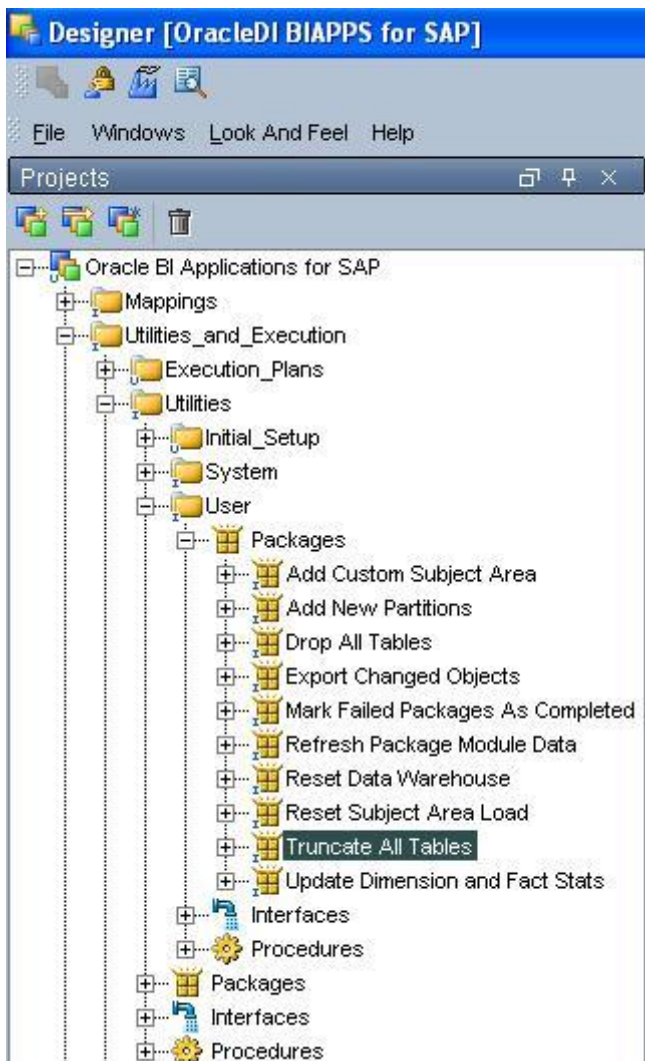
1. Use transition SE37
2. Provide the function module name like ZOBI_INCR_Product and click on "CREATE".
3. In the ATTRIBUTES tab, make the Function module Remote-enabled.
4. Provide the import parameters in the IMPORT tab. Commonly used parameters are START DATE (type STRING), END DATE (Type STRING), DELETE_FLAG (CHAR1) & INCREMENTAL (CHAR1).
5. Provide output internal table details in TABLES tab. The structure shall be similar to Z table created above.
6. Use source code tab to write code
7. Define structure (TYPES), internal tables(ITAB)& work areas (WA).
8. Indentify new inserts based on the creation date from the driving table.
9. Use the object class & tables to filter delta based on date & time range "start date& time and "end date& time" using SAP Standard Function Modules 'CHANGEDOCUMENT_READ_HEADERS' & 'CHANGEDOCUMENT_READ_POSITIONS'.

10. The field change indicator (CHNGIND) from table CDPOS will determine whether the records is a new insert 'I' or update 'U' or deleted 'D'. It should be stored along with MATNR in the code as well as the Z table.
11. Merge the Inserts & Delta and identify the latest record and update the Z table.
12. For reference Check function module ZOBI_Product
13. Include the interfaces for incremental load from SDE and SILOS in the package to create incremental flow.

About Truncating Target Tables Truncation of Tables should be done using the

'Truncate All Tables' package, which is located in the Projects\Oracle BI Applications
7.9.7.2\Utilities_and_Execution\Utilities\User\Packages folder. This utility truncates all the tables in the data warehouse prior to a forced full load.

Figure 197.



About the ETL_PROC_WID Setting

Use the ETL_PROC_WID setting in the W_PARAM_G table in custom mappings. ETL_PROC_WID is a reference key to the Run History of the ODI Master Package.

About Indices and Naming Conventions

Staging tables typically do not require any indices. Use care to determine if indices are required on staging tables. Create indices on all the columns that the E-LT will use for dimensions and facts (for example, ROW_WIDs of Dimensions and Facts, INTEGRATION_ID and DATASOURCE_NUM_ID and flags). Carefully consider which columns or combination of columns filter conditions should exist, and define indices to improve query performance. Inspect the OTB objects for guidance. Name all the newly created tables as WC_. This helps visually isolate the new tables from the out-of-the-box tables. Keep good documentation of the customizations done; this helps when upgrading your data warehouse. Once the indices are decided upon, they should be registered in the ODI Model (for more information, see To map source values to domain values:

1. Identify all the Oracle Business Analytics Warehouse table columns that use domain values.
2. For a list of columns that use domain values, see Oracle Business Analytics Warehouse Data Model Reference.
3. List all of your source values that qualify for conversion to one of the domain values.
4. Map each source value to a domain value. If any of your source system values do not map to a prepackaged domain value, and you might modify the list of domain values, then create a list of new domain values and map your orphaned source system values to your newly created domain values. You cannot modify all domain value sets. Also, you must check which metrics are affected by the modified domain value set. For more information, see the Oracle Business Analytics Warehouse Data Model Reference.
5. In ODI Designer, open the applicable extract inline view (inline views follows the SQ_* naming convention).
6. Open the Expression of the applicable column that uses domain values. Alternatively, if you want to add domain values, add them to this expression.
7. Save the changes.
8. Regenerate the scenario.

How to add an index to an existing fact or dimension table

Adding a New Dimension to the Oracle Business Analytics Warehouse

Follow this procedure to add a new dimension table to the Oracle Business Analytics Warehouse.

To add a new dimension and use it with an existing fact table:

1. In ODI Designer, log in as SUPERVISOR, and display the Models view.
2. In the Oracle BI Applications for SAP folder, create a new dimension table structure (with appropriate system columns), Use the naming convention W_<Dimension Name>_D.
3. In the Oracle BI Applications for SAP folder, create a new staging table structure (with appropriate system columns), using the naming convention W_<Dimensions Name>_DS.
4. In the definition of the fact tables related to this new dimension, insert a reference constraint to this dimension table. The dimension table must be the parent table in the reference constraint definition.

5. Display the Projects view, and select the Oracle BI Applications for SAP folder.
6. Create a new interface called SDE_SAP_<XYZ>.<Interface name>(_ Full) to populate the dimension stage. Create a custom inline view interface if necessary. This inline view interface will be the source of the main interface for loading the dimension stage. Assign the appropriate knowledge module for these interfaces. Set the appropriate value for the options in the assigned knowledge module. If required, create separate interfaces for full load and incremental load. Refer to existing interfaces as examples.
7. Create a new package called SDE_<Package_name> to contain the interfaces for loading the dimension stage. Create the steps for refresh and evaluate of variables OBI_START_PACKAGE and IS_INCREMENTAL (if full and incremental is necessary). The 'Refresh OBI_START_PACKAGE' step is always the first step. Add other variables or objects as required by the package. Invoke the interfaces created in the previous step in this package. A package can have two branches, for full or incremental load. Refer to existing SDE packages as a base sample.
8. Create new custom interfaces and packages for the SILO jobs for loading the dimension table from dimension stage. To set truncate options for the interfaces, assign a value to the KM option OBI_TRUNCATE_TABLE. A value of F is for truncate on full load, while a value of Y is for truncate always. For SDE it is usually truncate always, while for SIL it is truncate on full load.
9. Insert the new packages into the master package.

Adding a New Fact Table to the Oracle Business Analytics Warehouse

Follow this procedure to add a new fact table to the Oracle Business Analytics Warehouse.

To add a new fact table:

1. In ODI Designer, log in as SUPERVISOR, and display the Models view.
2. In the Oracle BI Applications for SAP folder, create a new fact table structure (with appropriate system columns), using the naming convention W_<Fact_Name>_F. In the constraints tree of this new fact table, insert a reference constraint to all dimension tables related to this fact. The dimension table has to be the parent table in the reference constraint definition. In the flexfields tab of this new fact table, change the value of the OBI_MODULE flexfield to the appropriate module of this new fact table such as OM,SCA,HR, FIN or CUSTOM. In addition, set the value of the OBI_SUBJECT_AREA flexfield to your customized subject area name.
3. In the Oracle BI Applications for SAP folder, create a new fact staging table structure (with appropriate system columns), using the naming convention W_<Fact Name>_FS.
4. Display the Projects view, and select the Oracle BI Applications for SAP folder.
5. Create a new interface called SDE_SAP_<XYZ>.<Interface name>(_ Full) to populate the fact stage. Create a custom inline view interface if necessary. This inline view interface will be the source of the main interface for loading the dimension stage. Assign the appropriate knowledge module for these interfaces. Set the appropriate value for the options in the assigned knowledge module. If required, create separate interfaces for full load and incremental load. Refer to existing interfaces as examples.
6. Create a new package called SDE_<Package_name> to contain the interfaces for loading the fact stage. Create the steps for refresh and evaluate of variables OBI_START_PACKAGE and IS_INCREMENTAL (if full and incremental is necessary). The 'Refresh OBI_START_PACKAGE' step is always the first step. Add other variables or objects as required by the package. Invoke the interfaces created in the previous step in this package. A package can have two branches, for full or incremental load. Refer to existing SDE packages as a base sample.

7. Create new custom interfaces and packages for the SILO jobs for loading the fact table from the fact stage. To set truncate options for the interfaces, assign a value to the KM option OBI_TRUNCATE_TABLE. A value of F is for truncate on full load, while a value of Y is for truncate always. For SDE it is usually truncate always, while for SIL it is truncate on full load.
8. Insert the new packages into the master package.

Adding a New Dimension Table for a New Fact Table in the Oracle Business Analytics Warehouse

The steps for creating a new dimension table are similar to the steps for incremental change capture.

To add a new dimension table for a new fact table:

1. In the new custom fact loading interface, drag and drop the new dimension table into the source pane of the interface.
2. Create a join between the dimension table and the fact staging table.
3. Extract the ROW_WID column from the dimension table and assign it to the corresponding column in the target fact table.

Category 3 Customizations

Category 3 customizations use the Universal adapter to load data from sources that do not have pre-packaged adapters.

This section contains the following topics:

- How to Add New Data as a Whole Row Into a Standard Dimension Table
- Configuring Extracts
- Configuring Loads
- Configuring Slowly Changing Dimensions

How to Add New Data as a Whole Row Into a Standard Dimension Table

Follow this procedure to add new data as a whole row into a standard dimension table in the Oracle Business Analytics Warehouse.

To add new data as a whole row into the standard dimension table:

1. Identify and understand the existing structure of staging tables. Refer to Oracle Business Analytics Warehouse Data Model Reference for the table structures. Non-system columns can include the null value.
2. Create a custom SDE interface to load the data into the staging table in the custom folder for this purpose. The staging table needs to be populated with incremental data (rows that have been added or changed since the last Refresh ELT process), for performance reasons.
3. Populate the INTEGRATION_ID column with the unique identifier for the record. The combination of INTEGRATION_ID and DATASOURCE_NUM_ID is unique. When importing the data, make sure that a unique identifier for the external data source is inserted in the DATASOURCE_NUM_ID column. Populate the INTEGRATION_ID column with the unique identifier for the record. The combination of INTEGRATION_ID and DATASOURCE_NUM_ID is unique. When importing the data, make sure that a unique identifier for the external data source is inserted in the DATASOURCE_NUM_ID column. The DATASOURCE_NUM_ID should be set to 51 for interfaces that source data from the SAP ERP (version SAP ECC 6.0) and DATASOURCE_NUM_ID should be set to 50 for SAP ERP (version SAP 4.6C). This is a

reserved value and is used in all standard interfaces. For example, a value of 51 can be defined for DATASOURCE_NUM_ID in the custom SDE interfaces. The standard SDE interfaces populate the INTEGRATION_ID column of the dimension staging table (used for resolving the dimension's value). The custom process must be used to populate the same column with a unique identifier from the external data source.

4. After the data is populated in the staging table, use the standard SIL interfaces to populate the dimension target tables.
5. Modify the SDE and SIL interfaces of all the related fact tables (fact tables that need to be linked to this dimension). The custom fact SDE interfaces must populate the foreign key column of the changed dimension (using a custom map table process to convert from SAP ERP ROW IDs to the external data source row IDs). The custom SIL interface should be modified to use the appropriate DATASOURCE_NUM_ID, because the standard SIL interfaces assume that the DATASOURCE_NUM_ID for the dimensions are the same as the fact table's DATASOURCE_NUM_ID. It is important to decide when the data is going to be loaded. If it is going to be loaded along with the SAP ERP source data, make sure that failure recovery is configured correctly. The preconfigured Master Packages truncate the target staging table prior to loading. Upon failure, when the Master Package restarts the task, the data is truncated and all the data is loaded again. If the data from the external source gets loaded into the same staging table, be careful with how you handle this situation, since you cannot use the truncate table functionality. The data migrating into the staging table is not incrementally loaded, and, therefore, should be cleaned up prior to attempting to load this table again. In such a case, it is recommended that you encapsulate the extract part from both the sources inside an Master Package. Note that the data from both the sources should be run at the same time, all the time. If it is decided that the data is going to be loaded at different time frequencies, then the new SDE packages need not depend on the preconfigured SDE packages and can use the Truncate Table option for failure recovery. Make sure the shared SIL process depends on the SDE processes from both sources.

Configuring Extracts

Each application has prepackaged logic to extract particular data from a particular source. This section discusses how to capture all data relevant to your reports and ad hoc queries by addressing what type of records you want and do not want to load into the Oracle Business Analytics Warehouse, and contains the following topics:

- Extracting Additional Data
- Setting Up the Delimiter for a Source File

Extracting Additional Data

You can configure extract mappings and Interfaces in the Oracle Business Analytics Warehouse to accommodate additional source data. For example, if your business divides customer information into separate tables based on region, then you would have to set up the extract interface to include data from these tables.

Extracting New Data Using an Existing Source Table

Extract interfaces generally consist of source tables, expressions used in the target columns, and a staging table. If you want to extract new data using the existing interface, you have to modify the extract interface to include the new data by performing the following tasks:

To modify an existing interface to include new data:

1. Modify the existing interface to extract information from the source, and add it to an appropriate extension column.
2. Modify the Expressions in the target table to perform any necessary transformations.
3. Save the changes.
4. Regenerate the scenario.

You have to determine which type of extension column to map the data to in the staging table. After you modified the extract interface, you would also have to modify the corresponding load interfaces (SDE and SIL) to make sure that the extension columns that you added are connected all the way from the staging table to the target data warehouse table.

Extracting Data from a New Source Table

This section is not Applicable for SAP Adaptor.

Extract interfaces (which have the SQ_* naming convention) reside in source-specific folders within the repository. Extract interfaces are used to extract data from the source system. You can configure these extract interfaces to perform the following:

- Extract data from a new source table.
- Set incremental extraction logic.

Setting Up the Delimiter for a Source File

When you load data from a Comma Separated Values (CSV) formatted source file, if the data contains a comma character (,), you must enclose the source data with a suitable enclosing character known as a delimiter that does not exist in the source data. **Note:** Alternatively, you could configure your data extraction program to enclose the data with a suitable enclosing character automatically.

For example, you might have a CSV source data file with the following data:

Months, Status

January, February, March, Active

April, May, June, Active

If you loaded this data without modification, ODI would load 'January' as the Months value, and 'February' as the Status value. The remaining data for the first record (that is, March, Active) would not be loaded. To enable ODI to load this data correctly, you might enclose the data in the Months field within the double-quotation mark enclosing character (" ") as follows:

Months, Status

"January, February, March", Active

"April, May, June", Active

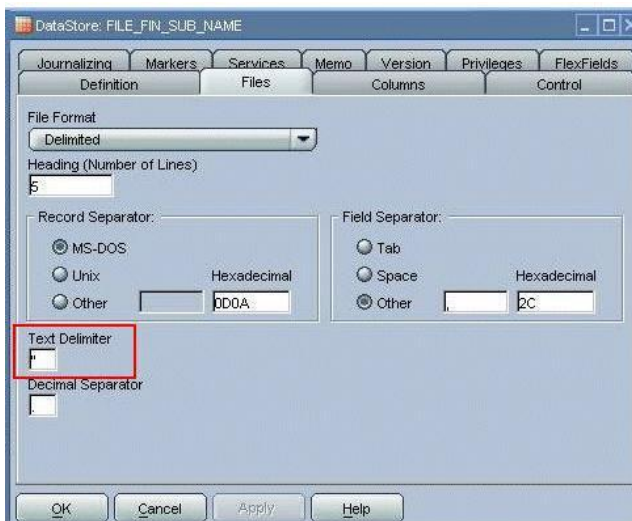
After modification, ODI would load the data correctly. In this example, for the first record ODI would load 'January, February, March' as the Months value, and 'Active' as the Status value.

To set up the delimiter for a source file:

1. Open the CSV file containing the source data.
2. Enclose the data fields with the enclosing character that you have chosen (for example, "). You must choose an enclosing character that is not present in the source data. Common enclosing characters include single quotation marks (') and double quotation marks (").

3. Save and close the CSV file.
4. In ODI Designer, display the Models view, and expand the 'Oracle BI Applications 7.9.7.2' folder. Identify the data stores that are associated with the modified CSV files. The CSV file that you modified might be associated with one or more data stores.
5. In ODI Designer, change the properties for each of these data stores to use the enclosing character, as follows:
 - a. Double-click the data source, to display the DataStore: <Name> dialog.
 - b. Display the Files tab.

Figure 198.



- c. Use the Text Delimiter field to specify the enclosing character that you used in step 2 to enclose the data.
- d. Click OK to save the changes.

You can now load data from the modified CSV file.

Configuring Loads

This section explains how to customize the way that Oracle Business Intelligence Applications loads data into the Oracle Business Analytics Warehouse. For example, you might want to delete records from the Oracle Business Analytics Warehouse that have been deleted in the source system.

Filtering and Deleting Records

Not Applicable for SAP ERP Source

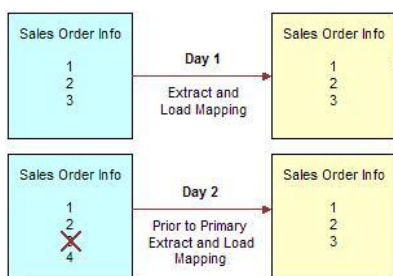
In a typical implementation, records that are deleted from your source system are not removed from the Oracle Business Analytics Warehouse. If you want to mark these records as deleted in the Oracle Business Analytics Warehouse, which were removed from the source system's database and archived in a separate database, you must enable the primary extract and delete mappings. Primary extract mappings flag records that are deleted from the Oracle Business Analytics Warehouse. Delete mappings sets the DELETE_FLG column to 'Y' for these records in the warehouse tables. When enabled, primary extract and delete mappings by default look for any records removed from the source system's database. If these mappings find that the records no longer exist in that database, the mappings mark them as deleted in the data warehouse

About Primary Extract and Delete Mappings Process

Not Applicable for SAP ERP Source

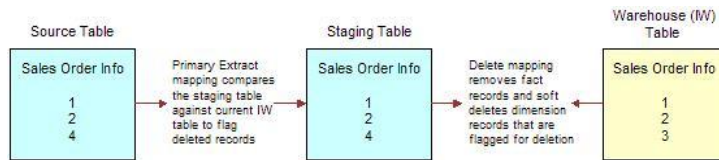
Before you decide to enable primary extract and delete sessions, it is important to understand their function within the Oracle Business Analytics Warehouse. Primary extract and delete mappings allow your analytics system to determine which records are removed from the source system by comparing primary 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 199. Extract and load mappings



The following figure 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 the following figure. When the extract and load mappings run, the new sales order is added to the warehouse.

Figure 200. Primary Extract and Delete Mappings



About Working with Primary Extract and Delete Mappings

Not Applicable for SAP ERP Source

The primary extract (*_Primary) and delete mappings (*_IdentifyDelete and *_Softdelete) serve a critical role in identifying which records have been physically deleted from the source system. However, there are some instances when you can disable or remove the primary extract and delete mappings, such as when you want to retain records in the Oracle Business Analytics Warehouse that were removed from the source systems' database and archived in a separate database. Because delete mappings use Source IDs and Key IDs to identify purged data, if you are using multiple source systems, you must modify the SQL Query statement to verify that the proper Source ID is used in the delete mapping. In addition to the primary extract and delete mappings, the configuration of the delete flag in the load mapping also determines how record deletion is handled. You can manage the extraction and deletion of data in the following ways:

- Deleting the configuration for source-archived records
- Deleting records from a particular source
- Enabling delete and primary-extract sessions
- Configuring the Record Deletion flag
- Configuring the Record Reject flag

Deleting the Configuration for Source-Archived Records

Not Applicable for SAP ERP Source

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 AST_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 About Working with Primary Extract and Delete Mappings.

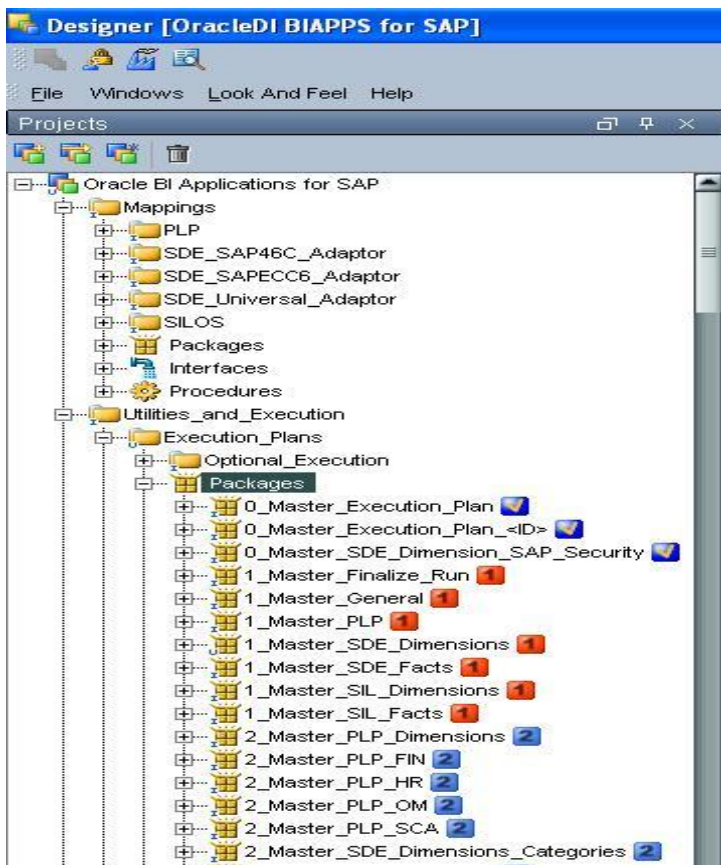
Enabling Delete and Primary Extract Sessions

Not Applicable for SAP ERP Source

If you want to mark your source-deleted records as deleted in the Oracle Business Analytics Warehouse, you need to enable the delete and primary extract tasks for your application. To enable delete and primary extract sessions:

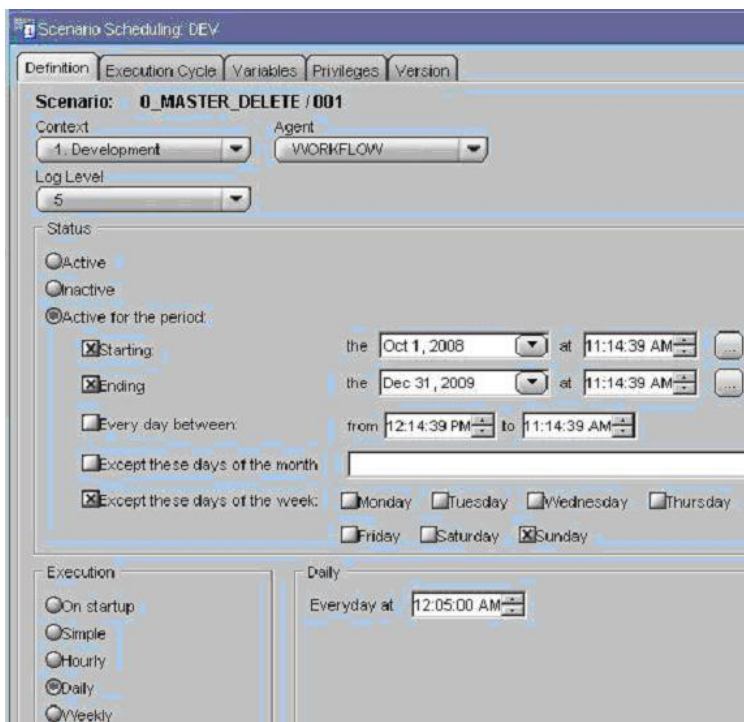
1. In ODI Designer, display the Projects view, and expand the 'Oracle BI Applications 7.9.7.2' folder.
2. Expand the \Master_Packages\Tools\Delete_Master_Packages folder.

Figure 201.



3. Double-click the package 0_Master_Delete to display the Scenario: <Name> dialog.

Figure 202.



4. Use the scheduling settings to schedule the package. The jobs for marking the deleted records are already part of the master packages. These jobs will mark the deleted records identified in by the execution of the O_Master_Delete package.

Configuring Slowly Changing Dimensions

The Oracle Business Analytics Warehouse provides Type II slowly changing dimension (SCD) functionality, which allows you to track the history of updates to dimension records. When a record in the Oracle Business Analytics Warehouse has an update, the updated information is posted into a new row and the old information is kept for historical reporting purposes. The Oracle Business Analytics Warehouse identifies and applies the slowly changing dimension logic after data has been extracted and transformed to be source-independent. You can configure Oracle Business Intelligence Applications to support both Type I SCDs, in which data is overwritten with updates, and Type II SCDs, in which the original records are maintained while a new record stores the updated data. Whether you choose to apply Type I or Type II SCD functionality to a column depends whether the column contains historically significant attributes. The table below shows the dimensions that are required using the SCD Type II update.

Table 45. *Dimensions that are required using the SCD Type II update*

Dimension	Adaptor
W_EMPLOYEE_D	SAP ERP Adapter
W_INVENTORY_PRODUCT_D	SAP ERP Adapter
W_POSITION_D	SAP ERP Adapter
W_PRODUCT_D	SAP ERP Adapter

By default, all dimensions are using Type I updates. This behavior of TYPE I or TYPE II is managed by a SIL package level Parameter for each dimension called 'TYPE2_FLG'. By default, the value of the parameter is set to 'N' out-of-the-box (except for the above mentioned tables). To turn a dimension to Type II SCD, update enabled set the value of the parameter for the particular dimension - 'TYPE2_FLG' to 'Y'.

About Identifying Historically Significant Attributes

You might want to retain a history of all the updates to a particular dimension so that you can use them in reports. These dimensions are known as historically significant attributes. For example, if a customer moves to a different region and you assign that customer a new regional salesperson and territory ID, you might want to keep records of that customer's account history with the original salesperson and territory ID. In this case, the salesperson and territory IDs are historically significant attributes. In contrast, you might have a load that populates the telephone number field. If your business does not perform data analysis on phone number history, then this information might be considered a historically insignificant attribute. Identifying attributes as significant or insignificant allows you to determine the type of SCD you require. However, before you can select the appropriate type of SCD, you must understand their differences.

About the Extract View

The extract view of any given table in the Staging Area consists of four types of records:

- New records
- Changed records with data that is historically insignificant

- Changed records having historical significance
- Changed records whose changes have no significance of any kind and are ignored altogether Of the four kinds of records, only the first three are of interest for the data mart. Of those three, brand new records and records whose changes are tracked as SCDs are both treated as new and become inserts into the Oracle Business Analytics Warehouse. Records with changes that are important but not historically tracked are overwritten in the Oracle Business Analytics Warehouse, based on the primary key.

Type I and Type II Slowly Changing Dimensions

After you have correctly identified your significant and insignificant attributes, you can configure the Oracle Business Analytics Warehouse based on the type of slowly changing dimension (SCD) that best fits your needs: Type I or Type II.

Type I Slowly Changing Dimension

A Type I SCD overwrites the column's value and is the default SCD for the Oracle Business Analytics Warehouse. Although a Type I does not maintain history, it is the simplest and fastest way to load dimension data. Type I is used when the old value of the changed dimension is not deemed important for tracking or is an historically insignificant attribute. For example, you might want to use Type I when changing incorrect values in a column. In the figure below, the State Name column for the supplier KMT is changed in the source table Suppliers, because it was incorrectly entered as California. When the data is loaded into the Oracle Business Analytics Warehouse table, no historical data is retained and the value is overwritten. If you look up supplier values for California, records for KMT do not appear; they only appear for Michigan, as they have from the beginning.

Figure 203. 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

Type II Slowly Changing Dimension

A Type II SCD creates another record and leaves the old record intact. Type II is the most common SCD because it allows you to track historically significant attributes. The old records point to all history prior to the latest change, and the new record maintains the most current information. Slowly changing dimensions work in different parts of a star schema (the fact table and the dimension table). The figure below shows how an extract table becomes a Oracle Business Analytics Warehouse dimension table (W_PRODUCT_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 MARA is loaded into the target table W_PRODUCT_D, and each record is assigned a unique primary key (ROW_WID).

Figure 204. 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 Type I SCD is applied and Mary is overwritten with Jane. Because the change in ABC's record was a Type I SCD, there was no reason to create a new customer record. In contrast, the change in XYZ's record shows a change of sales territory, an attribute that is historically significant. In this example, the Type II slowly changing dimension is required.

As shown in the figure below, instead of overwriting the Sales Territory column in the XYZ's record, a new record is added, assigning a new ROW_WID, 172, to XYZ in W_ORG_D. XYZ's original record, 102, remains and is linked to all the sales that occurred when XYZ was located in the West sales territory. However, new sales records coming in are now attributed to ROW_WID 172 in the North sales territory.

Figure 205. An example Type 2 Slowly Changing Dimension

TS_CUSTOMERS			IA_CUSTOMERS			
Customer Name	Sales Territory	Customer Contact	Customer KEY	Customer Name	Sales Territory	Customer Contact
ABC Co.	East	Jane	101	ABC Co.	East	Jane
XYZ Inc.	West	John	102	XYZ Inc.	West	John
	North		172	XYZ Inc.	North	John

Effective Dates

Effective dates specify when a record was effective. For example, if you load a new customer's address on January 10, 2003 and that customer moves locations on January 20, 2003, the address is only effective between these dates. Effective Dates are handled in the following manner:

- If the source supplies both effective dates, these dates are used in the warehouse table.
- If the source does not supply both the effective to and effective from dates, then the Type II logic creates effective dates.

If the source supplies one of the two effective dates, then you can set up the Oracle Business Analytics Warehouse to populate the missing effective dates using a wrapper mapping. This situation is discussed in this section. By default, these wrapper sessions are disabled and need to be enabled in order to be executed. For example, in the W_PRODUCT_D table previously discussed, T-RD324 moved to a new sales territory. If your source system supplied historical data on the location changes, your table might contain a record for T-RD324 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 T-RD324 has changed to configurable material product type, then a second record is inserted with an effective from date of January 1, 2002, and an effective to date of January 1, 3714.

Customizing Stored Lookups, Domain Values, and Adding Indexes

This section contains miscellaneous information that applies to all three categories of customization in Oracle Business Intelligence Applications, and contains the following topics:

- About Stored Lookups
- About Domain Values

To map source values to domain values:

5. Identify all the Oracle Business Analytics Warehouse table columns that use domain values.
 6. For a list of columns that use domain values, see Oracle Business Analytics Warehouse Data Model Reference.
 7. List all of your source values that qualify for conversion to one of the domain values.
 8. Map each source value to a domain value. If any of your source system values do not map to a prepackaged domain value, and you might modify the list of domain values, then create a list of new domain values and map your orphaned source system values to your newly created domain values. You cannot modify all domain value sets. Also, you must check which metrics are affected by the modified domain value set. For more information, see the Oracle Business Analytics Warehouse Data Model Reference.
 9. In ODI Designer, open the applicable extract inline view (inline views follows the `SQ_*` naming convention).
 10. Open the Expression of the applicable column that uses domain values. Alternatively, if you want to add domain values, add them to this expression.
 11. Save the changes.
 12. Regenerate the scenario.
- How to add an index to an existing fact or dimension table

About Stored Lookups

This section explains codes lookup and dimension keys.

Codes Lookup

Some source systems use intelligent codes that are intuitively descriptive, such as HD for hard disks, while other systems use non-intuitive codes (like numbers, or other vague descriptors), such as 16 for hard disks. While codes are an important tool with which to analyze information, the variety of codes and code descriptions used poses a problem when performing an analysis across source systems. The lack of uniformity in source system codes must be resolved to integrate data for the Oracle Business Analytics Warehouse. The code lookup in the load mapping integrates both intelligent and non-intuitive codes by performing a separate extract for codes, and inserting the codes and their description into a codes table. The codes table provides the load mapping with a resource from which it can automatically perform a lookup for code descriptions. The following components and process are used in a lookup:

W_CODE_D Table

The load control table `W_CODE_D` consolidates all codes for future reference and assigns them a category and a single language for efficient lookup capability.

Codes Mappings

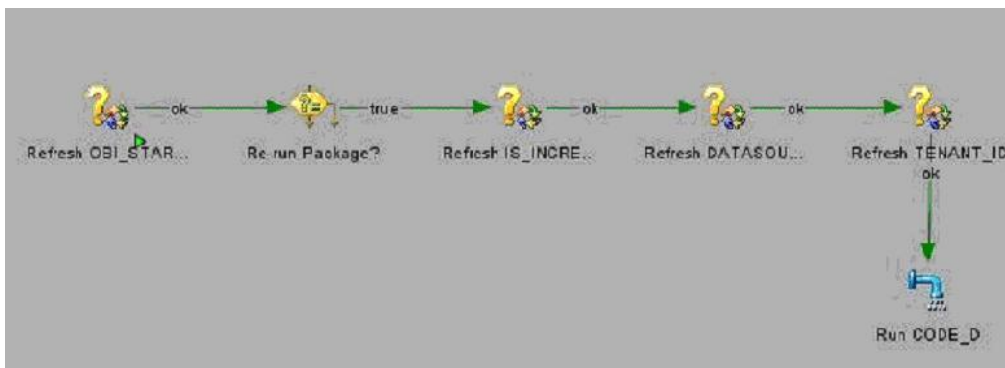
The Oracle Business Analytics Warehouse uses mappings designed to extract codes from source systems and populate the W_CODE_D table in preparation for use by the load mapping. To understand how codes mappings function, it is helpful to first understand the columns within W_CODE_D, (see table below).

Table 46. *Columns in Code Mapplet*

Column	Description
DATASOURCE_NUM_ID	Unique identifier of the source system from which data was extracted
SOURCE_CODE1	The first code in the hierarchy of the various source system codes used to identify a particular code and description combinations
SOURCE_CODE2	The second code in the hierarchy of the various source system codes used to identify a particular code and description combinations
SOURCE_CODE3	The third code in the hierarchy of the various source system codes used to identify a particular code and description combinations
SOURCE_DESC_1	Short description of the source system code
SOURCE_DESC_2	Long description for code

The naming convention for mappings designed for codes lookup is SDE_[SOURCE]_CodeDimension_[CATEGORY]. The figure below shows an example of a code package in ODI Designer.

Figure 206. *An example of a code package in ODI Designer*



About Resolving Dimension Keys

By default, dimension key resolution is performed by the Oracle Business Analytics Warehouse in the load mapping. The load interface uses prepackaged, reusable lookup transformations to provide pre-packaged dimension key resolution. This section describes how dimension keys are looked up and resolved. There are two commonly used methods for resolving dimension keys. The first method, which is the primary method used, is to perform a lookup for the dimension key. The second method is to supply the dimension key directly into the fact load mapping.

Resolving the Dimension Key Using Lookup

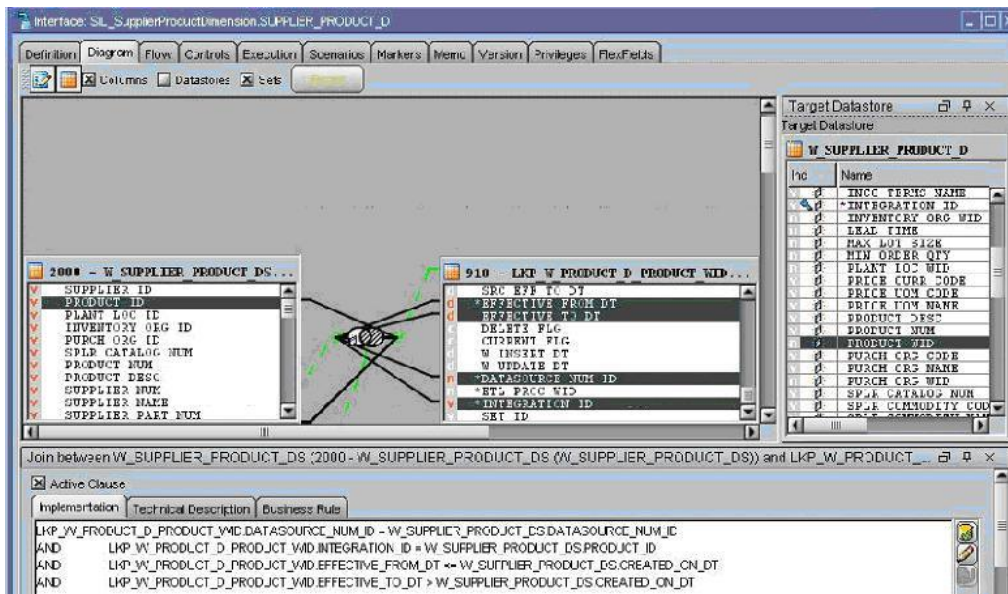
If the dimension key is not provided to the Load Interface through database joins, the load mapping performs the lookup in the dimension table. The load mapping does this using prepackaged Lookup Interfaces. To look up a dimension key, the Load Interface uses the INTEGRATION_ID, the DATASOURCE_NUM_ID, and the Lookup date, which are described in the table below.

Table 47. Columns Used in the load mapping Dimension Key Lookup

Port	Description
INTEGRATION ID	Uniquely identifies the dimension entity within its source system. Formed from the transaction in the Source Adapter of the fact table.
DATASOURCE_NUM_ID	Unique identifier of the source system instance.
Lookup Date	The primary date of the transaction; for example, receipt date, sales date, and so on.

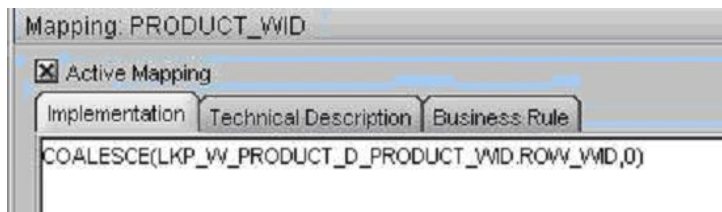
If Type II slowly changing dimensions are enabled, the load mapping uses the unique effective dates for each update of the dimension records. When a dimension key is looked up, it uses the fact's primary date to resolve the appropriate dimension key. The effective date range gives the effective period for the dimension record. The same entity can have multiple records in the dimension table with different effective periods due to Type II slowly changing dimensions. This effective date range is used to exactly identify a record in its dimension, representing the information in a historically accurate manner. In the figure below, the Supplier Product Key Lookup Interface illustrates the four columns needed for the load interface lookup. In the figure below, the INTEGRATION_ID, DATASOURCE_NUM_ID, Lookup Date (EFFECTIVE_FROM_DT and EFFECTIVE_TO_DATE) are highlighted in the Supplier Product Key Lookup.

Figure 207. INTEGRATION_ID, DATASOURCE_NUM_ID, Lookup Date (EFFECTIVE_FROM_DT and EFFECTIVE_TO_DATE) are highlighted in the Supplier Product Key Lookup



The transformation then outputs the Supplier Product Key (the dimension key) to the data warehouse table W_SUPPLIER_PRODUCT_D.

Figure 208. W_SUPPLIER_PRODUCT_D



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.

About the Domain Value Conversion Process

To best understand the domain value conversion process, consider an example of two source systems, Source System A and Source System B. Each source system stores two types of employee events, hire and rehire. Source system A uses H to denote a hire event and R to denote a rehire event, whereas source system B uses 1 to denote a hire event and 2 to denote a rehire event. When the Oracle Business Analytics Warehouse extracts data from both systems, it ports those source values through the extract package until the data reaches the W_EVENT_GRP_CODE column in the W_EVENT_TYPE_DS staging table. The load package then ports the extracted source values (H and R from source system A, and 1 and 2 from source system B) into the interface. Within the interface, source values are translated into domain values (HIR and REH) based on a set of rules that are particular to your business practices.

Preparing to Define the Rules

You must define the rules so that the ODI interface knows how to map your specific source values to the given set of domain values. Before you set up the rules you must:

1. Analyze all of your source values and how they map to the prepackaged domain values. You might find that you need to create additional domain values for particular columns. The result of this preparation work is a list of each source value and how it is mapped to a domain value.
2. Implement this logic in the ODI Interface. To set up the logic, modify the Expression transformation in the ODI Interface for each affected column. After the ODI Interface converts the source-specific values to domain values, the domain values are inserted into an Oracle Business Analytics Warehouse table. In this example, the HIR and REH values populate the W_EVENT_TYPES table (Interface SDE_ORA_EventTypeDimension_AdditionalEvents_FromFile.EVENT_TYPE_DS).

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:

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)
```

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
```

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)
```

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)
```

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)
```

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)
```

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)
```

About Extending the Domain Value Set

The Oracle Business Analytics Warehouse is also extensible in that you can create additional domain values for those columns that do not fit into the existing domain value definitions. However, before you modify the domain value set for a particular column, you first perform impact analysis on existing metrics. For example, the Oracle Business Analytics Warehouse prepackages the following two events:

- **New Hire.** This event occurs when a new person is hired.
- **New Position.** This event occurs when a position is created, but an existing employee might be hired internally. If you have an event that represents both a New Hire and a New Position, you might have to create a third event that depicts both. If you create this new event type domain value, you need to include it in the applicable metric definitions so as to account for all hires and positions.

Configuring the Domain Value Set with CSV Worksheet Files

Domain values are a set of distinct values used to calculate prepackaged metrics. These values are provided by Oracle Business Analytics Warehouse to allow you to create metric calculations independent of source system values. Oracle Business Analytics Warehouse provides CSV worksheet files to map source system values to domain values. You can add to these worksheet files if you need extra source system values and map them to domain values. You can also modify the worksheet files if you need to customize the domain values. You can use an existing domain value if you want to change the preconfigured metrics. Otherwise you can create a new domain value and create new metrics based on this domain value. The source system values that are not mapped to a domain values in the CSV worksheet files have a question mark (?) as the domain value in the Oracle Business Analytics Warehouse. These values do not affect the domain values metrics. If there are no worksheet files to map the source system values to the domain values, you need to modify the domain values using ODI (for more information, see *Configuring the Domain Value Set Using ODI Designer*. To map source values to domain values using CSV worksheet files:

1. Identify the Oracle Business Analytics Warehouse table columns that use domain values. For a list of columns that use domain values, see the Oracle Business Analytics Warehouse Data Model Reference.
2. List all of your source values that qualify for conversion to one of the domain values.
3. Map each source value to a domain value. If any of your source system values do not map to a prepackaged domain value, and you can modify the list of domain values, then create a list of new

domain values and map your orphaned source system values to your newly created domain values. You cannot modify all domain value sets. Also, you must check which metrics are affected by the modified domain value set. For more information, see the Oracle Business Analytics Warehouse Data Model Reference.

4. Open the CSV worksheet file in the <ODI_HOME>\oracledi\biapps_odi\odifiles\odidatafiles\srcfiles folder (for example, C:\ODI\oracledi\biapps_odi\odifiles\odidatafiles\lkpfiles).
5. Edit the file to map your source values to the existing domain values. Alternately, if you want to add additional domain values, add them in this worksheet file.
6. Save and close the CSV file.

Configuring the Domain Value Set Using ODI Designer

If there are no worksheet files to map the source system values to the domain values, you need to modify the values using ODI Designer. For more information on configuring the domain value set with CSV worksheet files, see Configuring the Domain Value Set with CSV Worksheet Files. Configuring the domain value set for a particular column, using ODI, entails one or both of the following activities:

- Mapping source-specific values to domain values
- Adding more domain values to the prepackaged set of values
- Regardless of which activity you choose, the configuration occurs in the Expression transformation of the applicable Source Adapter mapplet. The following procedure shows how to configure the Expression transformation to change the domain values.

To map source values to domain values:

7. Identify all the Oracle Business Analytics Warehouse table columns that use domain values.
8. For a list of columns that use domain values, see Oracle Business Analytics Warehouse Data Model Reference.
9. List all of your source values that qualify for conversion to one of the domain values.
10. Map each source value to a domain value. If any of your source system values do not map to a prepackaged domain value, and you might modify the list of domain values, then create a list of new domain values and map your orphaned source system values to your newly created domain values. You cannot modify all domain value sets. Also, you must check which metrics are affected by the modified domain value set. For more information, see the Oracle Business Analytics Warehouse Data Model Reference.
11. In ODI Designer, open the applicable extract inline view (inline views follows the SQ_* naming convention).
12. Open the Expression of the applicable column that uses domain values. Alternatively, if you want to add domain values, add them to this expression.
13. Save the changes.
14. Regenerate the scenario.

How to add an index to an existing fact or dimension table

Dimension and Fact Tables in the Oracle Business Analytics Warehouse use the following two types of index:

- E-LT Index

- E-LT Indexes are used for Unique/ Binary Tree index.
- Query Index

Query Indexes are used for Non-Unique/ Bit Map Index.

To add an index to an existing fact or dimension table:

1. In ODI Designer, display the Models view, and expand the 'Oracle BI Applications 7.9.7.2' folder.

Figure 209.



2. Expand the Fact or Dimension node as appropriate.
3. Expand the Table in which you want to create the index.
4. Right-click on the Constraints node, and select Insert Key to display the Key: New dialog.
5. Display the Description tab.

Figure 210.

Key: New

Markers Memo Version Privileges FlexFields

Description Columns Control

Table: W_AP_BALANCE_F

Name
W_AP_BALANCE_F_F8

Key or Index Type

☐ Not Unique Index

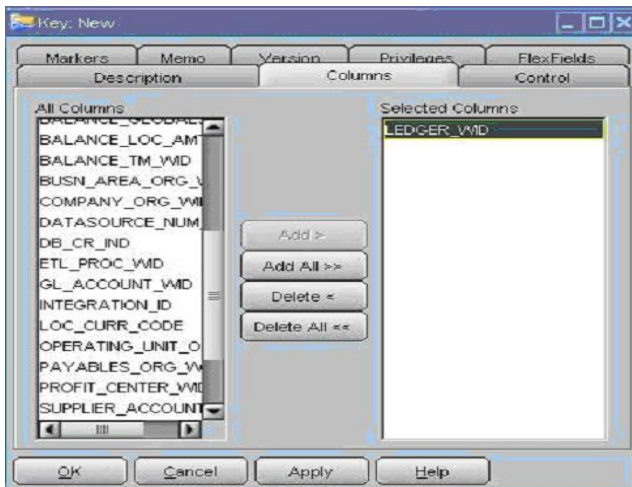
☐ Primary Key

☒ Alternate Key

OK Cancel Apply Help

6. Select the Alternate Key radio button, and update the name of the Index in the Name field.
7. Display the Column tab.

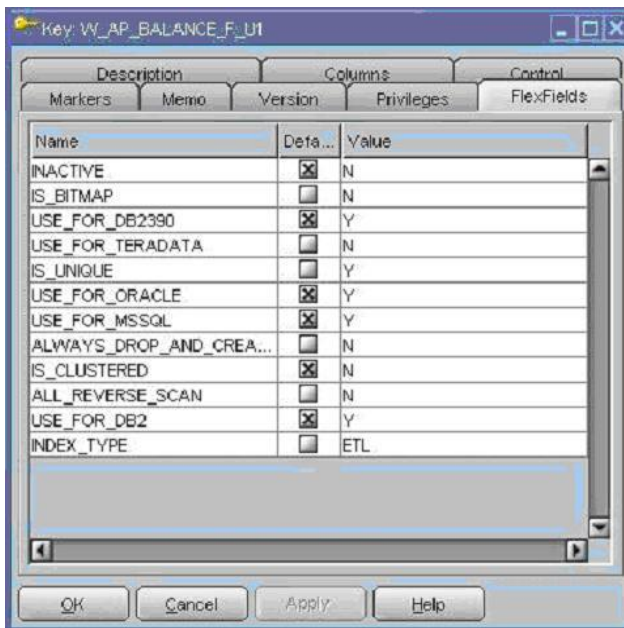
Figure 211.



8. Select the column on which you want to create the index.

9. Display the FlexFields tab.

Figure 212.



10. Use the settings to specify the index type, as follows:

- For 'Query' type indexes (the default), set the value of the IS_BITMAP parameter to 'Y' and the value of the IS_UNIQUE parameter to 'N'.
- For 'E-LT' type indexes, clear the check box for the INDEX_TYPE parameter and set the value to 'ETL'. In addition, set the value of the IS_BITMAP parameter to 'N' and the value of the IS_UNIQUE parameter to 'Y'.

11. Save the changes.

Appendix

Localizing Oracle Business Intelligence Deployments

Oracle Business Intelligence is designed to allow users to dynamically change their preferred language and locale preferences. This chapter contains the following topics on how to configure Oracle Business Intelligence Applications for deployment in one or more language environments besides English:

- Section 1, "Process of Maintaining Translation Tables for Oracle BI"
- Section 2, "About Translating Presentation Services Strings"
- Section 3, "Changing the Default Currency in Analytics Applications"

1. Process of Maintaining Translation Tables for Oracle BI

The Oracle Business Intelligence Presentation layer supports multiple translations for any column name. When working with Oracle BI Answers or rendering a dashboard, users see their local language strings in their reports. For example, English-speaking and French-speaking users would see their local language strings in their reports. There are two kinds of application strings requiring translation in Oracle Business Intelligence:

- Metadata

Metadata strings are analytics-created objects in the Oracle Business Intelligence repository such as subject areas, metrics, and dimensions.

- Presentation Services

Presentation Services objects are end-user created objects such as reports, dashboards, and pages. Translations for Presentation Services strings are stored in the XML caption files. For more information on accessing these strings and changing the translations, see Oracle Business Intelligence Presentation Services Administration Guide.

This process includes the following tasks:

- Section B.1.1, "Upgrading Oracle Business Intelligence Seed Data for Non-English Locales"
- Section B.1.2, "Externalizing Customer Metadata Strings"
- Section B.1.3, "Adding Custom Translations to the W_LOCALIZED_STRING_G Table"

1.1 Upgrading Oracle Business Intelligence Seed Data for Non-English Locales

If Oracle Business Intelligence data in your deployment is to be viewed in a language other than English, you must also import Locale seed data into a data warehouse table called `W_LOCALIZED_STRING_G`. This process must be performed once for each language the application users might select to run their web client.

During the Oracle Business Intelligence installation, a directory named `ORACLE_HOME\biapps\seeddata` was created, which contains a sub directory for each language. Within each language sub directory is a `.dat` file (the data to be imported) and an `.inp` file (the WHERE clause governing the import).

1.1.1 Importing Locale Seed Data Into The Translation Table (`W_LOCALIZED_STRING_G`)

If the primary language being used is not English, you may have to import additional locale seed data (depending on the number of languages you use) as shown in the following procedure. Note: This procedure requires the use of the `dataimp` utility, which can only be used on 32-bit operating systems.

Note: This procedure should be performed only by a BI Administrator.

To verify creation of Translation Table (`W_LOCALIZED_STRING_G`) and corresponding indexes:

- a. Verify that the Business Analytics Warehouse contains the `W_LOCALIZED_STRING_G` table.
- b. Lookup the definitions of the indexes in DAC and create them manually in the Business Analytics Warehouse. The names of the indexes are as follows:
 - `W_LOCAL_STRING_G_U1`
 - `W_LOCAL_STRING_G_P1`
 - `W_LOCAL_STRING_G_M1`
 - `W_LOCAL_STRING_G_M2`

Note: It is better to add these indexes to `W_LOCALIZED_STRING_G` prior to importing the locale seed data in the next section, in order to safeguard against inadvertently duplicating the data in the table.

To import Locale seed data into the Translation Table (`W_LOCALIZED_STRING_G`)

- a. Open a command window and navigate to `ORACLE_HOME\biapps\seeddata\bin` directory.
- b. Run the import command in step 3 after replacing these connection parameters with the values appropriate to your database environment:
 - `UserName`
 - `Password`

- ODBCDataSource
- DatabaseOwner

c. 3. Run the import command:

```
ORACLE_HOME\biapps\seeddata\Bin\dataimp /u $UserName /p $Password /c
"$ODBCDataSource" /d $DatabaseOwner /f ORACLE_HOME\biapps\seeddata\l_
<XX>\analytics_seed_<XXX>.dat /w y /q 100 /h Log /x f /i ORACLE_
HOME\biapps\seeddata\l_<XX>\metadata_upgrade_<XXX>_<DBPlatform>.inp /l
metadata_upgrade_<XXX>.log
```

Note: Replace the XX with the Oracle Business Intelligence two-letter language code (fr, it) and the XXX with the Siebel Systems three-letter code (FRA, ITA).

- d. When you have finished importing the Locale seed data into the Translation Table (W_LOCALIZED_STRING_G), configure the initialization block in the Oracle BI Repository using the Oracle BI Administration Tool to connect to the database where this table resides.

Note: Unicode connectivity can be used to access databases that do not support Unicode.

1.2 Externalizing Customer Metadata Strings

Metadata Strings are loaded by the Oracle BI Server from a database table. In the case of Oracle Business Intelligence applications, this table is W_LOCALIZED_STRING_G in the data warehouse. The initialization block 'Externalize Metadata Strings' loads the strings for the Server. It is recommended that you run a test to make sure that this initialization block runs successfully. An example of the translation table is shown in Table B–1.

Table B–1 *Example of W_LOCALIZED_STRING_G Translation Table*

MSG_NUM	MSG_TEXT	LANG_ID
CN_Customer_Satisfaction	Customer Satisfaction	ENU DEU
CN_Customer_Satisfaction	Kundenzufriedenheit	PTB
CN_Customer_Satisfaction	Satisfação do cliente	

By default, the Oracle Business Intelligence repository is configured to run in English only. To deploy in any other language, you must externalize the metadata strings, as described in the following procedure.

To externalize metadata strings in the Oracle Business Intelligence repository

- a. Stop the Oracle BI Server.
- b. Using the Oracle BI Administration Tool in offline mode, open EnterpriseBusinessAnalytics.rpd.
- c. Select the entire Presentation layer and right-click the mouse to display the menu.
 - From the pop-up menu, select Externalize Display Names. (A check mark appears next to this option the next time you right-click on the Presentation layer.)
 - Unselect the Presentation layer.

Note: When Externalize Display Names is checked, all metadata strings are read from the W_LOCALIZED_STRING_G table in the data warehouse.

- d. In the Physical layer, select the Externalized Metadata Strings database icon. Expand the tree.
- e. Double-click Internal System Connection Pool. In the Connection Pool dialog General tab, the field Data source name should point to the data warehouse.
- f. Click OK and exit the Oracle BI Administration Tool.
- g. Restart the Oracle BI Server.

1.3 Adding Custom Translations to the W_LOCALIZED_STRING_G Table

When you add custom objects to the metadata and choose to externalize these objects (by right-clicking the object and checking the Externalize Display Name option), the Oracle BI Server looks for the translations (including those for the native language) in the W_LOCALIZED_STRING_G table.

If you do not externalize the display names, you do not need to perform the following procedures.

Note: The custom Presentation layer objects show up only in the native language of the metadata (the language in which you added these new objects).

1.3.1 Adding String Translations for Analytics Metadata

The following procedure describes how to add string translations for Oracle Business Intelligence metadata to the W_LOCALIZED_STRING_G table. This task occurs in any database administration tool, and in the Oracle BI Administration Tool.

To add string translations for Analytics metadata

- a. Open a database administration tool and connect to your data warehouse database.
- b. Query for the table named W_LOCALIZED_STRING_G and add a new record to the table, as defined below in steps 4 to 8.
- c. Obtain the Message Key from the Oracle BI Administration Tool as follows:
 - In the Oracle BI Administration Tool, right-click on the new Presentation layer metadata object and select Properties from the menu.
 - The Message key is displayed in the dialog under Custom Display Name. The Message key is the part that starts with CN_.

For example, double-click the Pipeline catalog directory in the Presentation layer. The Custom Display name is Valueof(NQ_SESSION.CN_Pipeline). CN_Pipeline is the Message Key.

- d. Enter your deployment language in the new record.
- e. Enter the Message Type required (for example, Metadata, FINS_Metadata).
- f. Select the Message Level AnalyticsNew, then do the following:
 - In the Message Text column, add the translation of the object.
 - Check the flags (set to Yes) for the Translate and Active columns.
 - Set the Error Message # column to 0.
- g. Enter the required Message Facility (for example, HMF, FIN).
- h. Repeat Step 3 through Step 7 for each new metadata object string.
- i. Exit the database administration tool, then restart the Oracle BI Server.

2 About Translating Presentation Services Strings

The translations for such Presentation Services objects as report and page names are stored in the xxxCaptions.xml files available in the ORACLE_HOME\biapps\catalog\res\web\|_<Language Abbreviation>\Captions directories. In multiple language deployment mode, if you add any additional Presentation Services objects, such as reports and new dashboard pages, you also need to add the appropriate translations. Add these translations using the Catalog Manager tool. For more information on using this utility, see Oracle Business Intelligence Presentation Services Administration Guide.

3 Changing the Default Currency in Analytics Applications

In Oracle Business Intelligence Applications, you may see a dollar sign used as the default symbol when amounts of money are displayed. In order to change this behavior, you must edit the currencies.xml file using the following procedure. The currencies.xml file is located in the following directories:

- Windows:

ORACLE_HOME\bifoundation\web\display\currencies.xml

- UNIX:

ORACLE_HOME/bifoundation/web/display/currencies.xml

To change the default currency in Analytics Applications

- In a text editor, open the currencies.xml file.
- Look for the currency tag for the warehouse default (tag="int:wrhs"):

```
<Currency tag="int:wrhs" type="international" symbol="$" format="$#" digits="2"
displayMessage="kmsgCurrencySiebelWarehouse">

<negative tag="minus" format="-$#" />

</Currency>
```

- Replace the symbol, format, digits and negative information in the warehouse default with the information from the currency tag you want to use as the default.

For example, if you want the Japanese Yen to be the default, replace the contents of the warehouse default currency tag with the values from the Japanese currency tag (tag="loc:ja-JP"):

```
<Currency tag="loc:ja-JP" type="local" symbol="¥" locale="ja-JP" format="$#" digits="0">

<negative tag="minus" format="-$#" />

</Currency>
```

When you are finished, the default warehouse currency tag for Japanese should look like the following example:

```
<Currency tag="int:wrhs" type="international" symbol="¥" format="$#" digits="0"
displayMessage="kmsgCurrencySiebelWarehouse">

<negative tag="minus" format="-$#" />

</Currency>
```

- Save and close the currencies.xml file.