Siebel Customer-Centric Enterprise Warehouse Installation and Configuration Guide

Version 7.8.3

September 2005

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404

Printed in the United States of America

PRODUCT MODULES AND OPTIONS. This guide contains descriptions of modules that are optional and for which you may not have purchased a license. Siebel's Sample Database also includes data related to these optional modules. As a result, your software implementation may differ from descriptions in this guide. To find out more about the modules your organization has purchased, see your corporate purchasing agent or your Siebel sales representative.

Proprietary Information

Siebel Systems, Inc. considers information included in this documentation and in Siebel Online Help to be Confidential Information. Your access to and use of this Confidential Information are subject to the terms and conditions of: (1) the applicable Siebel Systems software license agreement, which has been executed and with which you agree to comply; and (2) the proprietary and restricted rights notices included in this documentation.

Contents

Chapter 6: Configuring the Siebel Business Analytics Repository for Siebel Customer-Centric Enterprise Warehouse

Chapter 7: Deploying Multiple Siebel Customer-Centric Enterprise Warehouse Applications

Chapter 8: Configuring Common Components of the Siebel Customer-Centric Enterprise Warehouse

Chapter 12: Checklist for Configuring Siebel Customer-Centric Enterprise Warehouse Applications

Chapter 13: Configuring Siebel Enterprise Contact Center Analytics

Chapter 15: Configuring Siebel Enterprise Workforce Analytics

Chapter 16: Configuring Siebel Financial Analytics

Chapter 17: Configuring Siebel Strategic Sourcing Analytics

Chapter 18: Configuring Siebel Supply Chain Analytics

Index

1 What's New in This Release

What's New in Siebel Customer-Centric Enterprise Warehouse Installation and Configuration Guide

The Siebel Customer-Centric Enterprise Warehouse Installation and Configuration Guide replaces the Siebel Customer-Centric Enterprise Warehouse Implementation Guide.

Table 1 lists changes described in this version of the documentation to support Release 7.8.3 of the software.

Table 1. What's New in Siebel Customer-Centric Enterprise Warehouse Installation and Configuration Guide, Version 7.8.3

Topic	Description
Chapter 6, "Configuring the Siebel Business Analytics Repository for Siebel Customer-Centric Enterprise Warehouse"	Added chapter. This chapter explains how to configure the Siebel Business Analytics repository for the Siebel Customer-Centric Enterprise Warehouse applications.
Chapter 7, "Deploying Multiple Siebel Customer-Centric Enterprise Warehouse Applications"	Added chapter. This chapter provides instructions for deploying multiple applications for the Siebel Customer-Centric Enterprise Warehouse.
Chapter 9, "Configuring Siebel Customer-Centric Enterprise Warehouse for Oracle 11i"	Added chapter. This chapter provides configuration information that is specific to Oracle 11i applications.
Chapter 16, "Configuring Siebel Financial Analytics"	Added chapter. This chapter combines the previous Siebel Financial Analytics applications of Siebel General Ledger Analytics, Siebel Payables Analytics, Siebel Receivables Analytics, and Siebel Profitability Analytics.
Process of Configuring the Table Analyze Utility on page 68	Added a process that explains how to configure the Table Analyzer Utility to analyze tables.
Setting Up the Representative Activities Table on page 238	Added a procedure that explains how to set up the Representative Activities table.
Setting Up the Contact Representative Snapshot Table on page 240	Added a procedure that explains how to set up the Contact Representative Snapshot table.
Setting Up the Benchmarks and Targets Table on page 241	Added a procedure that explains how to set up the Benchmarks and Targets table.
Process of Configuring Siebel Enterprise Sales Analytics for SAP R/3 on page 248	Added a process that explains how to configure Siebel Enterprise Sales Analytics for SAP R/3.

Table 1. What’s New in Siebel Customer-Centric Enterprise Warehouse Installation and Configuration Guide, Version 7.8.3

Topic	Description
Process of Aggregating Siebel Enterprise Sales Analytics Tables on page 268	Added a process that explains how to aggregate Siebel Enterprise Sales Analytics tables.
Domain Values and CSV Worksheet Files for Siebel Enterprise Sales Analytics on page 287	Added a section that lists the CSV worksheet files and the domain values for Siebel Enterprise Sales Analytics.
Process of Configuring Workforce Payroll for Oracle 11i on page 301	Added a process that explains how to configure Workforce Payroll for Oracle 11i.
Aggregating the Payroll Table for Siebel Enterprise Workforce Analytics on page 303	Added a procedure that explains how to set aggregate the Payroll table for Siebel Enterprise Workforce Analytics.
Domain Values and CSV Worksheet Files for Siebel Enterprise Workforce Analytics on page 304	Added a section that lists the CSV worksheet files and the domain values for Siebel Enterprise Workforce Analytics.
Mapping Siebel General Ledger Analytics Account Numbers to Group Account Numbers on page 312	Added a section that explains how to map Siebel General Ledger Analytics Accounts to Group Account Numbers.
Configuring the Customer Costs Lines and Product Costs Lines Tables for Siebel Profitability Analytics on page 324	Added a procedure that explains how to configure Siebel Profitability Analytics for Oracle 11i.
Configuring the Group Account Number Categorization for Siebel General Ledger Analytics on page 330	Added a section that explains how to configure the Financial Statement Item Categorization for SAP R/3.
Configuring the Customer Costs Lines and Product Costs Lines Tables for Siebel Profitability Analytics on page 337	Added a procedure that explains how to configure Siebel Profitability Analytics for SAP R/3.
About PeopleSoft Trees in Siebel Financial Analytics on page 339	Added a section that describes PeopleSoft Trees and explains how they are used in Siebel Financial Analytics.
Process of Configuring Siebel Financial Analytics for Post-Load Processing on page 345	Added a process that explains how to configure Siebel Financial Analytics for post-load processing.
Process of Configuring Siebel Financial Analytics for Post-Load Processing on page 345	Added a process that explains how to configure Siebel Financial Analytics for post-load processing.

Table 1. What’s New in Siebel Customer-Centric Enterprise Warehouse Installation and Configuration Guide, Version 7.8.3

Topic	Description
Process of Configuring Siebel Strategic Sourcing Analytics for SAP R/3 on page 355	Added a process that explains how to configure Siebel Strategic Sourcing Analytics for SAP R/3.
Configuring Expenses for Post-Load Processing on page 363	Added a process that explains how to aggregate Siebel Strategic Sourcing Analytics tables.
Domain Values and CSV Worksheet Files for Siebel Strategic Sourcing Analytics on page 375	Added a section that lists the CSV worksheet files and the domain values for Siebel Strategic Sourcing Analytics.
Process of Aggregating Siebel Supply Chain Analytics Tables on page 388	Added a process that explains how to aggregate Siebel Supply Chain Analytics tables.

Additional Changes

This version of the documentation also contains the following general changes:

The following terms were changed throughout the book:

- Siebel Enterprise Contact Center Analytics replaces Service.
- Siebel General Ledger Analytics replaces General Ledger.
- Siebel Payables Analytics replaces Payables.
- Siebel Receivables Analytics replaces Receivables.
- Siebel Profitability Analytics replaces Profitability.
- Siebel Supply Chain Analytics replaces Inventory.
- Siebel Strategic Sourcing Analytics replaces Sourcing.
- Siebel Enterprise Workforce Analytics replaces Workforce Operations and Workforce Retention.
- Siebel Enterprise Sales Analytics replaces Sales.

2 Overview of Installing and Configuring the Siebel Customer-Centric Enterprise Warehouse

To implement the Siebel Customer-Centric Enterprise Warehouse follow the process of installing, populating, and configuring as discussed in the subsequent chapters.

This chapter contains the following topics:

- Installing the Siebel Customer-Centric Enterprise Warehouse on page 19
- Initializing and Populating the Siebel Customer-Centric Enterprise Warehouse on page 20
- Configuring the Siebel Customer-Centric Enterprise Warehouse on page 20
- Additional Resources on page 22

Installing the Siebel Customer-Centric Enterprise Warehouse

To install the Siebel Customer-Centric Enterprise Warehouse, perform the following tasks:

1. **Determine your Analytics system configuration**. This includes determining the expected rate of growth for your data warehouse. For more information, see the following section in Chapter 4, "Installing the Siebel Customer-Centric Enterprise Warehouse Environment":

   Process of Defining Your Installation Configuration on page 31

2. **Set up system infrastructure.** This includes servers, databases, users, and so on, in your development environment. This step includes installing PowerCenter and patches. For more information, see the following sections in Chapter 4, "Installing the Siebel Customer-Centric Enterprise Warehouse Environment":

   Process of Installing PowerCenter and the Siebel Customer-Centric Enterprise Warehouse on page 39

   Process of Creating the Development Repository on page 42

3. **Extract, Transform, and Load (ETL).** Populate the Siebel Customer-Centric Enterprise Warehouse repository with the ETL objects required for your data warehouse. For more information, see Initializing and Populating the Siebel Customer-Centric Enterprise Warehouse on page 20.

4. **Set up the data warehouse tables.** Set up the Siebel Customer-Centric Enterprise Warehouse tables. For more information, see Configuring the Siebel Customer-Centric Enterprise Warehouse on page 20.

For more information on installing the Siebel Customer-Centric Enterprise Warehouse, see Chapter 4, "Installing the Siebel Customer-Centric Enterprise Warehouse Environment" This chapter covers the tasks to be completed before and then during installing the Siebel Customer-Centric Enterprise Warehouse.

Initializing and Populating the Siebel Customer-Centric Enterprise Warehouse

After you have installed the Siebel Customer-Centric Enterprise Warehouse, the next step is to load the data and see if it loads properly.

To populate the Siebel Customer-Centric Enterprise Warehouse, perform the following tasks:

1. **Initialize the Siebel Customer-Centric Enterprise Warehouse.** This task includes loading prepackaged data provided by Siebel Customer-Centric Enterprise Warehouse.
2. **Load data.** This task includes running the workflows to populate the data warehouse with source data.

For more information on initializing and populating the Siebel Customer-Centric Enterprise Warehouse, see Chapter 5, “Initializing and Populating the Siebel Customer-Centric Enterprise Warehouse.” This chapter outlines tasks needed to prepare your data warehouse to perform your initial data capture.

Configuring the Siebel Customer-Centric Enterprise Warehouse

When the data warehouse is loaded, you must then verify that the data is loaded according to your expectations. For example, Siebel Customer-Centric Enterprise Warehouse loads only booked orders for the Sales Order Lines table IA_SALES_ORDLNS; it filters out other information, such as nonbooked orders. However, you may want nonbooked orders in the Sales Order Lines table as well. Therefore, you would need to configure a mapping to reset the loading logic for this table.

To configure the Siebel Customer-Centric Enterprise Warehouse, perform the following tasks:

1. **Perform Gap Analysis.** This task includes comparing what is actually stored in the data warehouse against what you expect to be stored.
2. **Configure Repository Objects.** This task includes changing the way in which data is loaded to better meet your business requirements. It also includes resetting the run-time for any of the sessions, worklets, or workflows.
3. **Reload and Validate Data.** After your objects are configured, you must perform a check on the source data to verify that no data is missing. If data is missing, your reports may be inaccurate. Therefore, you must run the extract mappings and then check the staging tables to verify that the columns are populated.

Table 2 describes the chapters in this guide that provide more detail on the configuration phase of the Siebel Customer-Centric Enterprise Warehouse implementation.

Table 2. Chapters Discussing the Configuration Phase

Chapter	Description
Chapter 3, "Planning Your Warehouse Configuration Project"	This chapter provides the methodology for comparing prepackaged the Siebel Customer-Centric Enterprise Warehouse repository objects with your business organization's needs.
Chapter 4, "Installing the Siebel Customer-Centric Enterprise Warehouse Environment"	This chapter describes the tasks you must complete before and then during installing the Siebel Customer-Centric Enterprise Warehouse.
Chapter 5, "Initializing and Populating the Siebel Customer-Centric Enterprise Warehouse"	This chapter outlines the tasks you must do to prepare your data warehouse before performing your initial data capture.
Chapter 6, "Configuring the Siebel Business Analytics Repository for Siebel Customer-Centric Enterprise Warehouse"	This chapter describes how to configure the Siebel Customer-Centric Enterprise Warehouse repository.
Chapter 7, "Deploying Multiple Siebel Customer-Centric Enterprise Warehouse Applications"	This chapter provides instructions for deploying multiple applications for the Siebel Customer-Centric Enterprise Warehouse.
Chapter 8, "Configuring Common Components of the Siebel Customer-Centric Enterprise Warehouse"	This chapter provides instructions for modifying common components of the Siebel Customer-Centric Enterprise Warehouse.
Chapter 9, "Configuring Siebel Customer-Centric Enterprise Warehouse for Oracle 11i"	This chapter provides configuration procedures for the Oracle 11i that span multiple applications.
Chapter 10, "Storing, Extracting, and Loading Additional Data"	This chapter discusses the methodology for storing additional data in the data warehouse. In addition, it gives general procedures for extracting and loading new data.
Chapter 11, "Integrating Additional Data"	This chapter provides procedural information for creating and modifying the ETL components as well as creating and modifying the PLP components to populate aggregate tables.
Chapter 12, "Checklist for Configuring Siebel Customer-Centric Enterprise Warehouse Applications"	After the Siebel Business Analytics application is installed, you may need to configure certain objects for particular sources to meet your business needs.
Chapter 13, "Configuring Siebel Enterprise Contact Center Analytics"	This chapter provides configuration information about Siebel Enterprise Contact Center Analytics.

Table 2. Chapters Discussing the Configuration Phase

Chapter	Description
Chapter 14, "Configuring Siebel Enterprise Sales Analytics"	This chapter provides configuration information about the Siebel Enterprise Sales Analytics application.
Chapter 15, "Configuring Siebel Enterprise Workforce Analytics"	This chapter provides configuration information about Siebel Enterprise Workforce Analytics.
Chapter 16, "Configuring Siebel Financial Analytics"	This chapter provides configuration information about the Financial Applications. The Financial applications consist of Siebel General Ledger Analytics, Siebel Payables Analytics, Siebel Receivables Analytics, and Siebel Profitability Analytics.
Chapter 17, "Configuring Siebel Strategic Sourcing Analytics"	This chapter provides configuration information about Siebel Strategic Sourcing Analytics.
Chapter 18, "Configuring Siebel Supply Chain Analytics"	This chapter provides configuration information about Siebel Supply Chain Analytics.

Additional Resources

The following documentation contains information that may be relevant to your use of Siebel Business Analytics.

- For more information about the system requirements and supported platforms, see the *System Requirements and Supported Platforms for Siebel Enterprise Analytic Applications*.
- For a list of domain values, see *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.
- For more information about the installation and configuration tasks related to Siebel Business Analytics, see *Siebel Analytics Platform Installation and Configuration Guide*.
- For more information about the Siebel Data Warehouse, see *Siebel Analytics Applications Installation and Administration Guide*.
- For more information about PowerCenter and PowerMart installation and requirements, see the PowerCenter/PowerMart *Installation QuickStart Guide* and the PowerCenter/PowerMart *Installation and Configuration Guide*.

3 Planning Your Warehouse Configuration Project

This chapter provides guidelines on how to assess which objects need to be configured. After you complete your configuration project, do not forget that every time a business rule changes, or changes occur in your source system, you may need to configure this project again.

This chapter discusses important topics that are applicable to the planning of your configuration project. It contains the following topics:

Data Warehouse Configuration Stages

There are four stages where you can configure the handling of data to meet your needs. Configuration can happen in any of the following stages:

- **Extract Mapping.** For more information, see Configurable Objects in the Extract Mapping on page 24.
- **Load Mapping.** For more information, see Configurable Objects in the Load Mapping on page 25.
- **Post-Load Processing Mapping.** For more information, see Configurable Objects in the Post-Load Processing Mapping on page 25.
- **Reporting.** For more information, see Configurable Objects in the Reporting Area on page 26.

Where to configure an object depends on what you are trying to accomplish. Each stage performs different tasks when sourcing, transforming, loading, and reporting data. Generally, data goes through an extract mapping, load mapping, and front-end calculation process. Sometimes data also goes through the post-load stage, where the data is transformed before populating an aggregate table. Figure 1 illustrates these stages.

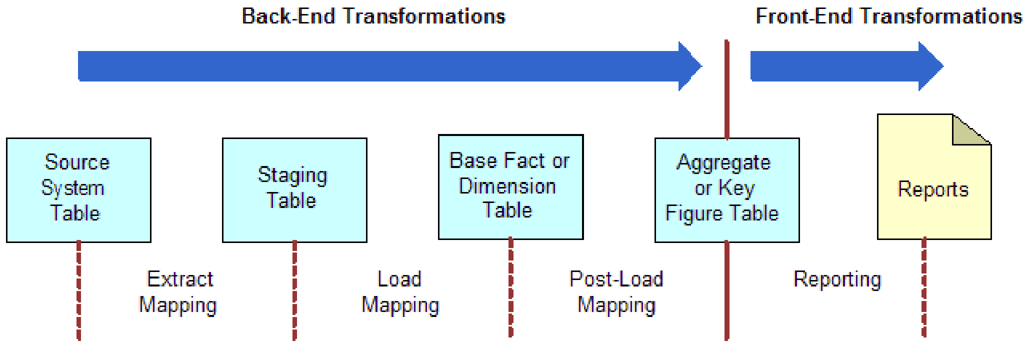


Figure 1. Stages for Configuration

Each of these stages contains various objects that are configurable. The sections that follow describe some of the common, configurable objects in each stage.

Configurable Objects in the Extract Mapping

Extract mappings are useful in extracting data from a source system and staging the data in a staging table. Within each extract mapping (with the exception of Universal Source extract mappings), you find the following objects—Business Component mapplet, Expression transformation, and staging table. A *mapplet* is a reusable object that contains a set of transformations, and it allows you to reuse the transformation instructions in multiple mappings.

You can configure the Business Component mapplet to do any of the following tasks:

- Extract additional data from new or existing source tables.
- Set up an incremental extract.
- Filter the type of records extracted.

NOTE: Universal Source extract mappings do not contain Business Components. These mappings have a flat file source which requires that most transformations be performed prior to the extract mapping.

You can configure the Expression transformation in the extract mapping to perform the following tasks:

- Set the Key ID for the records.
- Set the Source ID for the records.

The staging table is the target table for the extract mapping, and stores the extracted information.

Configurable Objects in the Load Mapping

Load mappings are useful in transforming and porting data from the staging table to a base fact or dimension table. Within each load mapping, you find the following objects—Source Qualifier transformation, Source Adapter mapplet, Analytic Data Interface (ADI) mapplet, and target table or tables.

You can configure the Source Qualifier transformation to filter records being loaded into the Siebel Customer-Centric Enterprise Warehouse target table.

The Source Adapter mapplet in the load mapping is where you usually find the source-dependent transformations. This is the last, and preferred, stage at which you can perform source-dependent transformations.

You can configure the Source Adapter mapplet to do any of the following tasks:

- Transform source data so that it becomes source-independent. For example, it maps some source values to the Siebel Business Analytics domain values.
- Provide values for dimension class table types, such as `COMPANY` as the business location type for the `MPLT_SAS_BUSN_LOCS_COMPANY` data.
- Load exchange rates and currency codes, instead of performing lookups to retrieve them.
- Set the dimension IDs in fact loads so that it can resolve the dimension keys.
- Set the Type 2 flags.
- Convert positive values to negative values.

The ADI mapplet in the load mapping must not be configured at the mapping level. Instead, changes must be made at the session level, using an SQL statement. The types of SQL statements include redirecting lookups to dimension, code, exchange rate, and custom-built dimension tables. The target tables of a load mapping can vary. If the table is a dimension load mapping, and Type 2 functionality has been enabled, you usually have two instances of the `IA_*` target table:

- One instance for inserting new records
- One instance for updating records

You also have an `OD_*` load control table. If the table is a fact load mapping, you usually have one `IA_*` target table and an `OD_*` load control table.

Configurable Objects in the Post-Load Processing Mapping

Post-Load Processing mappings are useful in transforming data from a base fact table and loading it into an aggregate table. For information about aggregate tables, see the *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*. Because Post-Load Processing mappings vary widely, the only common items that are configurable are:

- Time period for which data is stored or aggregated
- Calculations for metrics contained within the aggregate tables

Configurable Objects in the Reporting Area

If you have purchased Siebel Business Analytics, which contain the Siebel Customer-Centric Enterprise Warehouse and the Siebel Analytics Server schema and metadata, then you can configure metrics, attributes, reports, hierarchies, and schema definitions in the Siebel Analytics Server repository. This is the last stage at which you can perform configuration tasks.

About Scoping Your Configuration Project

After you have completed the initialization process as described in the installation chapters, the Siebel Customer-Centric Enterprise Warehouse is ready to be populated using the default configuration. Although the default configuration produces data in the data warehouse, which is used to populate your front-end analytic environment, it may not accurately reflect your business models. Your company may have unique business rules and requirements that depart from the Siebel Customer-Centric Enterprise Warehouse's prepackaged assumptions.

For example, your definition of a sales order may not be the same as the definition in the Siebel Customer-Centric Enterprise Warehouse. In addition to storing sales orders as defined by the Siebel Customer-Centric Enterprise Warehouse, the Sales Order table in your source system may also store sales inquiries, sales estimates, or other customized data to suit your specific needs. If your definition of sales orders is either more general or more specific than the Siebel Customer-Centric Enterprise Warehouse definition, you must configure the Siebel Customer-Centric Enterprise Warehouse to store the data you require. The research required to determine which areas you need to configure is called *gap analysis*. For more information, see Gap Analysis on page 26.

Gap Analysis

Gap analysis identifies the difference, or gaps, between what a product does by default and what your company needs the analytic solution to do. Gap analysis must be performed both at the time of installation, and as you prepare to do the configuration. This chapter focuses on gap analysis in terms of configuration.

TIP: It is recommended that you perform gap analysis, beginning with the front-end, and then work your way down to the Siebel Customer-Centric Enterprise Warehouse. This approach saves you time and effort. This chapter provides gap analysis only for the Siebel Customer-Centric Enterprise Warehouse.

Preparing for Gap Analysis

To best prepare for gap analysis, you must understand the following components of your analytic solution:

- **Your source system.** You can do this by understanding:
  - How you use the default features of your source system. Use your source system's standard documentation.

  - How you have customized your source system. Use any documentation that describes modifications to the source system’s default configuration. These documents may provide information on the effect that customizations to your source system have on how data is stored in the Siebel Customer-Centric Enterprise Warehouse.

- **The custom columns.** The custom columns you need that fall outside of the Siebel Customer-Centric Enterprise Warehouse data model.
- **The Siebel Customer-Centric Enterprise Warehouse.** You can do this by understanding:
  - The architecture of the product
  - Each data warehouse table column’s origin in your transaction source system
  - Each column’s business meaning and expected values
- **Your business rules.** How your business rules differ from the ones assumed by the Siebel Customer-Centric Enterprise Warehouse.

Guidelines for Populating the Siebel Customer-Centric Enterprise Warehouse

You are ready to embark on gap analysis for the back end as soon as you have populated the Siebel Customer-Centric Enterprise Warehouse with the data. The following list provides recommended steps for populating the Siebel Customer-Centric Enterprise Warehouse:

1. Complete the installation and initialization processes described in the previous installation chapters of this guide.
2. Use the default configuration to complete one entire extract and load cycle to populate the Siebel Customer-Centric Enterprise Warehouse with your data.

After the data warehouse is populated, you are ready to begin gap analysis.

Beginning Gap Analysis

The most efficient way to approach gap analysis is to start with the front-end reports. If you do not have easy access to the front-end reports, you can perform the gap analysis strictly from the back end. To understand where gap analysis ends in the front end, and begins in the back end, you must understand the entire gap-analysis process, as shown in Figure 2, and outlined as follows:

1. **Begin with your front-end reports.** Do they meet your needs? If your requirements do not exceed what is available in the reports, you do not have any gaps to analyze. Otherwise, you need to identify information that is not available in the default reports. To do so, proceed to the next step.
2. **Continue with your front-end solution.** Is the information you are seeking already available in, or can it be derived from, the front-end metadata?

3. **Conclude with the data warehouse data model.** Is the information you need already being extracted from the source system, or do the extracts need configuration?

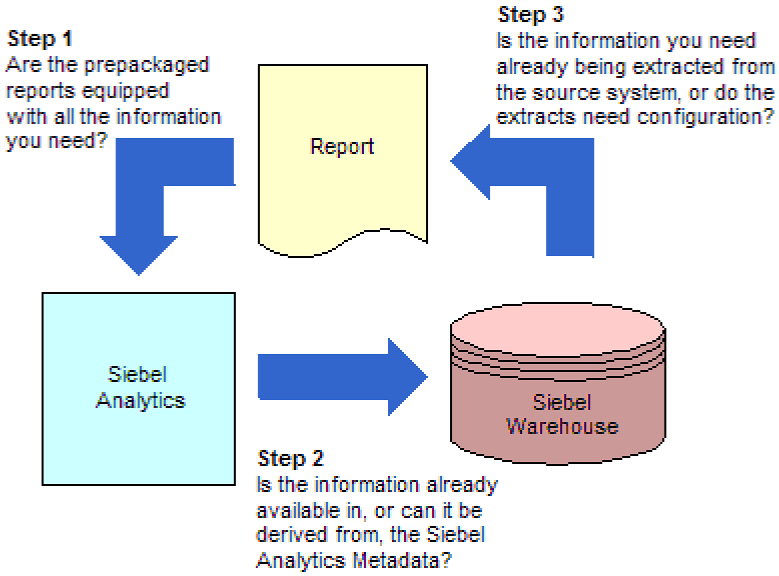

Step 1
Are the prepackaged reports equipped with all the information you need?
Report
Step 3
Is the information you need already being extracted from the source system, or do the extracts need configuration?
Siebel Analytics
Siebel Warehouse
Step 2
Is the information already available in, or can it be derived from, the Siebel Analytics Metadata?

Figure 2. Simple Approach to Gap Analysis

Resolving Gaps in the Data Warehouse Data Model

You can redefine, add, or delete functionality in the data warehouse model to best meet your business requirements. The following sections provide a brief overview of general back-end configuration information that spans all applications. They also provide configuration information specific to each application.

There are two high-level possibilities for configuration at the back end:

- The information you need is already extracted, but is not transformed and loaded as you require. To reconfigure, modify the appropriate mappings to transform and load additional data into the extension columns provided in the target table.
- The information you need is not being extracted from your source system. There are many ways to address this issue—the method you choose depends on how much information is required. You can modify an existing extract mapping to extract the data, or create a new extract mapping.

Roles Involved in the Gap-Analysis Process

Gap Analysis requires the expertise of a variety of analysts, experts, and specialists. The following is a list of roles commonly involved in gap analysis, which may be filled by more than one individual, according to your company’s needs:

- **Source specialist.** The source specialist is often a consultant who has helped you implement the source system or who works for the manufacturer. A source specialist knows your company’s requirements of the source system, as well as customizations that must be included in the Siebel Customer-Centric Enterprise Warehouse.
- **Business analyst.** The business analyst understands the intricacies of the business process and your reporting requirements. For example, the Customer Relationship Management business analyst understands how a sales order and an invoice are defined, and the information that is sourced from them to produce the necessary reports.
- **Front-end specialist.** The front-end specialist participates in designing reports and customizing the user interface. The business analysts use these reports created by the front-end specialist.
- **Professional Services Consultants (PSCs).** Siebel PSCs, or other third-party consultants, are responsible for making sure that your implementation of the Siebel Customer-Centric Enterprise Warehouse runs smoothly. They provide expertise in the Siebel Customer-Centric Enterprise Warehouse, data warehousing concepts, and various source systems.
- **Data warehouse expert.** The data warehouse expert is familiar with your storage requirements, performance levels, and database access.

Configuration Guidelines

Although the applications are configurable, it is not recommended that you modify the data model itself. Before performing any customization work, review the following best practices.

Guideline on Choosing Where to Transform an Object

When you choose an object to edit, such as a mapping, it is recommended that you change the transformation as close to the source as possible. For example, if you are editing a delete mapping to handle archived records, it is better to edit the Source Qualifier’s SQL statement rather than add a filter transformation after the Source Qualifier.

Guidelines on Documenting Changes

It is critical to document your work as you add and change objects in the data warehouse, so that the modifications are available when upgrading to new versions of the Siebel Customer-Centric Enterprise Warehouse. Create a change log to record the object added or modified, the objects it impacts, and a brief description of your task. (See Table 3 for a sample change log.) For example, if you add source tables to a Business Component, record the additional tables in the log as well as their purpose and impact. When upgrading, you can make the decision to add the tables again, or you can identify if the upgrade itself has made the additional tables unnecessary.

Table 3. Sample Change Log

Entry Number	Date Entered	Entered By	Object Type	Description of Change	Status
1	01/03/03	Sara Chin	Session	Modifying the S_M_S_SALES_ORDLNS_LOAD session so that non-booked orders are loaded into the Sales Order Lines table (IA_SALES_ORDLNS). Do this by removing the booking flag filter condition in the following two filter transformations: FIL_SALES_BKGLNS and FIL_SALES_BKGLNS1.	In progress. Assigned to Oscar Tapa.
2	01/04/03	Mike Thomas	Expression transformation	Reset the AP Balance ID for the EXP_SAI_AP_XACTS Expression transformation from GL_ACCOUNT_ID ||'~'|| VENDOR_SITE_ID ||'~'|| ORGANIZATION_ID to GL_ACCOUNT_ID ||'~'|| VENDOR_SITE_ID.	Completed by Joe Kennedy.

4 Installing the Siebel Customer-Centric Enterprise Warehouse Environment

This chapter describes the tasks you must complete before and then during installing Siebel Business Analytics Applications (Enterprise) and Siebel Customer-Centric Enterprise Warehouse.

NOTE: Siebel Business Analytics Platform must be installed before the Siebel Business Analytics Applications (Enterprise). For information on installing Siebel Analytics Platform, see the *Siebel Analytics Platform Installation and Configuration Guide*.

This chapter contains the following topics:

- Process of Defining Your Installation Configuration on page 31
- Process of Installing PowerCenter and the Siebel Customer-Centric Enterprise Warehouse on page 39
- Process of Creating the Development Repository on page 42
- Moving the Required Siebel Customer-Centric Enterprise Warehouse Files on page 50

Process of Defining Your Installation Configuration

Before installing any software, you must first define your installation configuration and related infrastructure requirements. To define your installation, perform the following tasks:

1. Determining Configuration Requirements on page 31
2. Gathering PowerCenter Information on page 32
3. Gathering Information About Your Data Warehouse and Staging Tables on page 36
4. Determining Source Requirements on page 37
5. Creating the Database Tables on page 38

Determining Configuration Requirements

This task is a step in the Process of Defining Your Installation Configuration on page 31.

To determine configuration requirements, you must identify the Siebel Business Analytics components to be installed and plan their configuration in your environment. Some factors that determine your installation configuration are:

- Hardware costs
- Operating system used to run Siebel Business Analytics
- Number of rows being extracted from the source

- Number of Siebel Business Analytics users

The configuration must specify:

- **Connection information for source machines.** Specify information about the network addresses of the source system from which data needs to be extracted, any associated constraints, ODBC connect strings, and so on.
- **Physical location of the server.** Identify the specific machines and their IP addresses, disk sizes, CPUs, memory, operating system, and so on.
- **Physical location of the databases.** Specify the database to be configured on the servers. The databases are used to house the data warehouse, staging tables, control tables, and so on.

  You must also make sure of the following in relation to the physical location of the databases:

  - It is recommended that you locate the staging area and the data warehouse in a single database with one user ID. This configuration makes it easier to create outer joins between the staging area and data warehouse tables, without needing multiple synonyms.
  - If you do create a separate user for the staging tables and the data warehouse, your staging area user must have select privileges on all the data warehouse objects. Create the appropriate grant and synonym creation scripts necessary for this purpose. To minimize input or output contention and improve performance, it is recommended that you create indexes in a separate tablespace.

Gathering PowerCenter Information

This task is a step in the Process of Defining Your Installation Configuration on page 31.

While you are preparing to install the Siebel Customer-Centric Enterprise Warehouse, you must gather information for your databases and repositories, such as passwords, IDs, and so on. Siebel Customer-Centric Enterprise Warehouse provide variables with predefined designations to act as placeholders for the information needed throughout this guide. The variables begin with the dollar symbol ($), followed by a name that represents the placeholder role. For example, the PowerCenter Server requires a password for the UNIX user ID: `$PC_SVR_UNIX_PASSWORD`.

Table 4 lists the variable names for installing the PowerCenter. You can use this table as a worksheet, and record the values applicable to your installation.

Table 4. Variable Names with Explanations for Installing PowerCenter

Variable	Explanation	Your Value
PowerCenter Server Parameters for Windows NT		
$PC_SVR_NT_USERID	The NT user ID with administrative access to PowerCenter and, if you are extracting from SAP, to PowerConnect.	
$PC_SVR_NT_PASSWORD	The password for the user ID to use to install PowerCenter.	

Table 4. Variable Names with Explanations for Installing PowerCenter

Variable	Explanation	Your Value
$PC_SVR_NT_DOMAIN	The NT domain on which PowerCenter is to be installed.	
PowerCenter Server Parameters for UNIX		
$PC_SVR_UNIX_USERID	The user ID that has administrative access to PowerCenter and, if you are extracting from SAP, administrative access to PowerConnect.	
$PC_SVR_UNIX_PASSWORD	The password to the UNIX user ID.	
PowerCenter Client Parameters		
$PC_CLIENT_USERID	The user ID that has administrative access to PowerCenter and, if you are extracting from SAP, to PowerConnect.	
$PC_CLIENT_PASSWORD	The password for the user ID (to be used to install PowerCenter).	
$PC_CLIENT_DOMAIN	The NT domain on which PowerCenter is to be installed.	
PowerCenter Repository Server Parameters		
$PC_REP_SVR_DOMAIN	The domain for the user account that runs the Repository Server service (must have the Logon as a service right).	
$PC_REP_SVR_USERNAME	The username for the user account that runs the Repository Server service.	
$PC_REP_SVR_PASSWORD	The password for the user account that runs the Repository Server service.	
$PC_REP_SVR_PORT_NUMBER	The port number the Repository Server uses to connect to the repository client applications. The default port number is 5001.	
$PC_REP_SVR_PORT_MAX	The maximum port number the Repository Server can assign to the Repository Agent process. The default value is 65535.	
$PC_REP_SVR_PORT_MIN	The minimum port number the Repository Server can assign to the Repository Agent process. The default value is 5002.	
PowerCenter Repository Parameters		
$PC_TEMP_REPO_NAME	The name you want to assign to your first, or temporary PowerCenter Repository (IA_TEMP).	

Table 4. Variable Names with Explanations for Installing PowerCenter

Variable	Explanation	Your Value
$PC_TEMP_REPO_USERID	The database user ID to connect to the database housing the first, or temporary PowerCenter Repository (IA_TEMP).	
$PC_TEMP_REPO_PASSWORD	The database password to connect to the database housing the first, or temporary PowerCenter Repository (IA_TEMP).	
$PC_TEMP_REPO_SERVER	The server housing the first, or temporary PowerCenter Repository (IA_TEMP).	
$PC_TEMP_REPO_DATABASE	The database where your first, or temporary PowerCenter Repository (for IA_TEMP) resides.	
$PC_TEMP_CODE_PAGE	The code page specific to the characters and language your system is using.	
$PC_CONTENT_INSTALL_DIR	The install directory you choose when you first install the PowerCenter content from the installation CD (when you run SETUP.EXE). By default, the content files are installed on C:\Program Files\Informatica PowerCenter 7.1.2.	
$PC_DEV_REPO_NAME	The name you want to assign to your development repository (IA_DEV).	
$PC_DEV_REPO_USERID	The user ID to connect to the database housing the PowerCenter development repository for IA_DEV.	
$PC_DEV_REPO_PASSWORD	The password to connect to the database housing the PowerCenter development repository for IA_DEV.	
$PC_DEV_REPO_SERVER	The server housing the PowerCenter development repository for IA_DEV.	
$PC_DEV_REPO_DATABASE	The database where your development repository (for IA_TEMP) resides.	
$PC_DEV_CODE_PAGE	The code page specific to the characters and language your system is using.	
$PC_QA_REPO_NAME	The name that you want to assign to your QA repository (IA_QA).	
$PC_QA_REPO_USERID	The user ID to connect to the database housing the PowerCenter QA repository (IA_QA).	

Table 4. Variable Names with Explanations for Installing PowerCenter

Variable	Explanation	Your Value
$PC_QA_REPO_PASSWORD	The password to connect to the database housing the PowerCenter QA repository (IA_QA).	
$PC_QA_REPO_SERVER	The server hosting the PowerCenter QA repository (IA_QA).	
$PC_QA_REPO_DATABASE	The database where your QA repository (for IA_QA) resides.	
$PC_QA_CODE_PAGE	The code page specific to the characters and language your system is using.	
$PC_PROD_REPO_NAME	The name you want to assign to your production repository (IA_PROD).	
$PC_PROD_REPO_USERID	The user ID to connect to the database housing the PowerCenter production repository (for IA_PROD).	
$PC_PROD_REPO_PASSWORD	The password to connect to the database housing the PowerCenter production repository (for IA_PROD).	
$PC_PROD_REPO_SERVER	The server housing the PowerCenter production repository (for IA_PROD).	
$PC_PROD_REPO_DATABASE	The database where your production repository for IA resides (IA_PROD).	
$PC_PROD_CODE_PAGE	The code page specific to the characters and language your system is using.	
PowerCenter Server License Information		
Product Key	You are asked to enter the Platform Key during your installation of PowerCenter. This information is provided with your installation CDs.	
Options Key	This key varies with each option you have purchased with PowerCenter, for example Team Based Development, Server Grid, and so on. This information is provided with your installation CDs.	
Database Key	This key varies with each RDBMS, so the Oracle key is different from the key you enter if you are running Sybase. This information is provided with your installation CDs.	

Table 4. Variable Names with Explanations for Installing PowerCenter

Variable	Explanation	Your Value
Siebel Business Analytics and Siebel Customer-Centric Enterprise Warehouse License Information		
Siebel Business Analytics and Siebel Customer-Centric Enterprise Warehouse Key	You are asked to provide the license XML file during your installation of Siebel Business Analytics and Siebel Customer-Centric Enterprise Warehouse. This file is provided with your installation CDs.	

Gathering Information About Your Data Warehouse and Staging Tables

This task is a step in the Process of Defining Your Installation Configuration on page 31.

You must gather information for the data warehouse and the staging tables.

Table 5 lists information about the data warehouse and staging tables. You can use this table as a worksheet and record the values applicable to your installation.

Table 5. Information About Your Data Warehouse and Staging Tables

Variable	Explanation	Your Value
STAGING AREA		
$IA_SA_USERID	The database user ID to connect to the database housing the staging tables. It is recommended that this ID be the same user as for the data warehouse, $IA_DW_USERID. If this ID is different from the data warehouse user, this user must have select privileges on all the data warehouse objects. Create the appropriate grant and synonym creation scripts necessary for this purpose.	
$IA_SA_PASSWORD	The password to connect to the database housing the staging tables. This password must be the same as the $IA_DW_PASSWORD password.	
$IA_SA_DW_SERVER	The server housing the data warehouse and staging area databases (preferably the same for both).	
$IA_SA_DATABASE	The database where the staging area and data warehouse tables reside (must be the same as $IA_DW_DATABASE).	

Table 5. Information About Your Data Warehouse and Staging Tables

Variable	Explanation	Your Value
DATA WAREHOUSE		
$IA_DW_USERID	The database user ID to connect to the database housing the data warehouse tables. It is recommended that this user ID is the same as the user ID for the data warehouse, $IA_SA_USERID.	
$IA_DW_PASSWORD	The password to connect to the database housing the data warehouse tables. It is recommended that this password is the same as $IA_SA_PASSWORD.	
$IA_SA_DW_SERVER	The server housing the data warehouse and staging area databases (preferably the same for both).	
$IA_DW_DATABASE	The database where the data warehouse tables reside (must be the same as $IA_SA_DATABASE).	

Determining Source Requirements

This task is a step in the Process of Defining Your Installation Configuration on page 31.

Some source systems and database platforms require additional steps, as detailed in this section. To install PowerConnect for specific source systems, see the relevant PowerConnect *User and Administrator Guide*.

NOTE: This section applies to customers using universal business adapters or SAP R/3.

Using the Universal Business Adapter

For source systems without prepackaged business adapters, you must provide flat file extracts. The flat files must conform to a template (located in the directory `$Siebel\ETL\SrcFiles`). Siebel Customer-Centric Enterprise Warehouse use the Universal Interface document to process and load the data into the data warehouse.

Using the SAP R/3 Source System

This section describes how to customize your Siebel Customer-Centric Enterprise Warehouse environment for a SAP R/3 source system.

Creating a Profile

The SAP R/3 administrator must create a profile in the R/3 system that allows you to use the integration features. This profile must include authorization for the objects and related activities listed in Table 6.

Table 6. SAP R/3 Integration Features

Integration Feature	Authorization Object	Activity
Import definitions, install programs	[S_DEVELOP]	All activities Also must set Development Object ID to PROG
Extract data	[S_TABU_DIS]	READ
Run file mode sessions	[S_DATASET]	WRITE
Submit background job	[S_PROGRAM]	BTCSUBMIT, SUBMIT
Release background job	[S_BTCH_JOB]	DELE, LIST, PLAN, SHOW Also must set Job Operation to RELE
Run stream mode sessions	[S_CPIC]	All activities

Generating ABAP Code

If you are extracting from SAP R/3, you must generate the ABAP code from the extract mappings. The mappings used to extract data from SAP R/3 all follow this naming convention, M_S_*_EXTRACT. For the mappings whose names fall into this category, you generate the ABAP codes. For more information on generating the ABAP codes, see the *PowerConnect for SAP R/3 User and Administrator Guide*.

Creating the Database Tables

This task is a step in the Process of Defining Your Installation Configuration on page 31.

Contact your system administrator to create your database requirements. You need six table spaces on your database for the following uses:

- Siebel Customer-Centric Enterprise Warehouse
- Staging Area
- Siebel Customer-Centric Enterprise Warehouse Repository
- Indexes
- Temporary Tables
- Schema Control Tables

Process of Installing PowerCenter and the Siebel Customer-Centric Enterprise Warehouse

This section describes how to install PowerCenter and the Siebel Customer-Centric Enterprise Warehouse. PowerCenter is included on its own CD, whereas the Siebel Customer-Centric Enterprise Warehouse is included on the Siebel Business Analytics Applications (Enterprise) CD. To install the required products, perform the following tasks:

1. Installing the Java SDK on page 39
2. Setting the Environment Variable for the Java SDK on page 39
3. Installing PowerCenter on page 40
4. Installing the Siebel Business Analytics Platform on page 40
5. Installing Siebel Customer-Centric Enterprise Warehouse and Enterprise Applications on page 40
6. Moving the Shell Repository File on page 41

Installing the Java SDK

This task is a step in the Process of Installing PowerCenter and the Siebel Customer-Centric Enterprise Warehouse on page 39.

This section provides instructions for installing the Java SDK. You need to install the Java SDK on the machine where the PowerCenter Server is installed.

NOTE: You need to install Java SDK version 1.4.2.x. Later versions of Java SDK are not supported.

To install the Java SDK 1.4.2

- Download the Java SDK version 1.4.2.x from the Sun Web site, and install it on the machines where the PowerCenter is installed.

  Make sure there are no spaces in the directory path. For example, on Windows:

  ```
  D:\j2sdk142
  ```

Setting the Environment Variable for the Java SDK

This task is a step in the Process of Installing PowerCenter and the Siebel Customer-Centric Enterprise Warehouse on page 39.

This section provides instructions for setting the required environment variable for the Java SDK.

To set the MY_JAVA_HOME environment variable for the Java SDK

1. In the Windows System Properties window, click the Environment Variables button.

2 Add a new variable to the System variables called MY_JAVA_HOME, and add the path to the Java SDK installation directory.

  Include the bin folder in your path. For example, on Windows:

```
MY_JAVA_HOME=C:\j2sdk142\bin
```

Installing PowerCenter

This task is a step in the Process of Installing PowerCenter and the Siebel Customer-Centric Enterprise Warehouse on page 39.

Before installing any other product, you must install your basic PowerCenter platform. For more information about PowerCenter installation and requirements, see the *Siebel Analytics Applications Installation and Administration Guide*.

Installing the Siebel Business Analytics Platform

This task is a step in the Process of Installing PowerCenter and the Siebel Customer-Centric Enterprise Warehouse on page 39.

You must install Siebel Business Analytics Platform before you install the Siebel Customer-Centric Enterprise Warehouse. For the supported version of Siebel Business Analytics for Siebel Customer-Centric Enterprise Warehouse, see the *System Requirements and Supported Platforms for Siebel Enterprise Analytic Applications*. For more information about installing Siebel Business Analytics Platform, see the *Siebel Analytics Platform Installation and Configuration Guide*.

Installing Siebel Customer-Centric Enterprise Warehouse and Enterprise Applications

This task is a step in the Process of Installing PowerCenter and the Siebel Customer-Centric Enterprise Warehouse on page 39.

The Siebel Customer-Centric Enterprise Warehouse software uses a standard Windows installation program (setup.exe) for installation. This task copies the Repository, Web Catalog, and ETL folders and files to your machine.

For more information on installing Siebel Relationship Management Warehouse, see *Siebel Analytics Applications Installation and Administration Guide*.

To install Siebel Customer-Centric Enterprise Warehouse and Enterprise Applications

1 Access the installation files, and then run the program setup.exe.

  The installation wizard window appears.

2 The installation wizard window prompts you through each screen, as shown in the following table.

To continue to the next screen, click Next. To return to a previous screen, click Back.

Screen	Action
Welcome	Click Next.
License Agreement	Select "I accept...." and click Next.
License File	The License file is an XML file corresponding to the installation type you are licensed for. This file is mailed to you, or installed on your corporate server. Select the appropriate XML file in the License File screen. Click Next.
Choose RMW or CEW	Click CEW and click Next.
Installation Directory	To accept the default installation (to the C:\ drive), click Next. To change the default (recommended), click browse and establish the installation path, then click Next.
Application Selection	Select the application or applications you want to install and click Next.
Summary Information (preinstallation)	A list of all the features you have chosen, and the directory where they are to be installed. Read this information to confirm it is correct. Click Next.
Installing	Placeholder screen that appears while the installer installs all the features you have selected. Click Next when done.
Summary Information (postinstallation)	Acknowledgement that the installation wizard has finished installing the Siebel Enterprise Analytic Applications. Click Next.
Finish	Click Finish.

Moving the Shell Repository File

This task is a step in the Process of Installing PowerCenter and the Siebel Customer-Centric Enterprise Warehouse on page 39.

The shell repository is a placeholder for the PowerCenter Repository Server. This shell repository is an empty repository that contains the folder structure, and the relational and application connections. When you have installed your required Siebel Applications components, place the Shell.rep file in the backup directory to restore the latest Siebel Business Analytics repository.

For information on how to restore the latest Siebel Customer-Centric Enterprise Warehouse repository, see Creating and Configuring an Empty Siebel Business Analytics Repository on page 43.

To move the shell repository file

1. Navigate to the `$Siebel\CommonFiles\ETL` folder.
2. Copy the Shell.rep file to the `$pmrepserver\RepositoryServer\bin\Backup` folder

   where `$pmrepserver` is the path of the PowerCenter Repository Server installation folder.

Process of Creating the Development Repository

The following section guides you through restoring the prepackaged Siebel Business Analytics repository and creating a development repository with just the contents applicable to your data warehouse.

To create a development repository, perform the following tasks:

- Creating and Configuring an Empty Siebel Business Analytics Repository on page 43
- Registering Packages for the SAP R/3 Repository on page 44
- Modifying the PowerCenter Server Configuration on page 44
- Starting the PowerCenter Server on page 45
- Importing Application Workflows into the Repository on page 45
- Backing Up the Development Repository on page 47
- Moving the Required PowerCenter Server Data Files on page 47
- Creating the Data Warehouse Schema Control Tables on page 47
- Deploying Stored Procedures on page 49
- Creating the Seed Data on page 49

Related Topic

- Modifying the Rollback Segments on page 42

Modifying the Rollback Segments

The PowerCenter repository population process performs a single, logged transaction. Because this single transaction is extremely large, you must configure your rollback segments to return your database to its original state if the install transaction fails for any reason.

For information on modifying rollback segments, contact your database administrator, or read the documentation for your specific database.

Creating and Configuring an Empty Siebel Business Analytics Repository

This task is a step in the Process of Creating the Development Repository on page 42.

This task creates a repository containing configuration folders for the different source systems that Siebel Business Analytics supports. The Siebel Business Analytics has source-independent and application-independent folders, as well as the Post-Load Process folder. After you create the repository, you may want to delete the folders and the connections that are not applicable to you.

You can also configure your server information in the following procedure.

To create and configure an empty Siebel Business Analytics repository

1. Start the Repository Server Administration Console.
2. Register the Repository Server.

   a. In the left pane, select the PowerCenter Repository Servers node, which appears under Console Root.

   b. On the Action menu, click New Server Registration.

   c. Enter the host name (the machine where the repository resides).

   d. Accept the default port number 5001 or enter the appropriate port number. Click OK.

      The Repository Server host name appears in the right pane under Hostname.

   e. Right-click the server name, and click Connect.

   f. In the Connection Information section, enter the Administrator password. Click OK.

3. Right-click on Repositories, and click New Repository.
4. In the General tab, enter a name for your new repository, and click Do not create any content.
5. Click the Database Connection tab and enter the following:

   - Database type
   - Code page
   - Connect string
   - Database user
   - Database password

6. Click the Licenses tab, and add your license key or keys, and Click Update.
7. Click OK.
8. Right-click on your new repository name, and click All Tasks, and Restore.

   The Restore Repository dialog box appears.

9. Select shell.rep, and click OK.

   Your repository is created.

Registering Packages for the SAP R/3 Repository

This task is a step in the Process of Creating the Development Repository on page 42.

If you are sourcing from SAP R/3, then you must register the pmsapplg.xml plug-in on the repository.

To register a package for the SAP R/3 repository

1. Start the Repository Server Administration Console.
2. Connect to your Repository Server.
3. Click on Available Packages.
4. Right-click on pmsapplg.xml plug-in, and click Register.
5. In the Register Plugin dialog box, select your SAP R/3 Repository from the Repository drop-down list.
6. Select Update existing plugin registration check box.
7. Enter your user name and password.
8. Click OK.

Modifying the PowerCenter Server Configuration

This task is a step in the Process of Creating the Development Repository on page 42.

Follow this procedure to modifying the PowerCenter Server configuration.

To modify the PowerCenter Server configuration

1. Start the Workflow Manager.
2. Connect to your Repository Server.
3. Right-click on Server, and click Edit.
4. In the Server Editor, edit the Host Name or IP Address.
5. Click Resolve Server.

   The Resolved IP Address box is unavailable, and displays the correct IP address of the server machine.
6. Click Advanced.
7. Enter the directory path to $pmserver

   where $pmserver is the path of the PowerCenter Server installation directory.
8. Click OK.

Starting the PowerCenter Server

At this point in the installation and configuration process, you can start the PowerCenter Server.

Follow this procedure to start the PowerCenter Server.

To start the PowerCenter Server

1. In the Control Panel, open the Services window.
2. Double-click Informatica.
3. The Informatica Properties dialog box opens.
4. In the General tab, in the Startup Type drop-down list, select Automatic.
5. In the General tab, in the Server Status area, click Start.
6. Close the Services window.

Importing Application Workflows into the Repository

This task is a step in the Process of Creating the Development Repository on page 42.

You must import individual workflows for every application purchased within the Siebel Customer-Centric Enterprise Warehouse. Each application involves importing the application-specific XML file using the PowerCenter Repository Manager or the command line. For example, for the Siebel Supply Chain Analytics applications there are two XML files—SupplyChain_Application_Oracle11i_Import_Control.xml and SupplyChain_Application_Oracle11i.xml. Use the SupplyChain_Application_Oracle11i.xml file to import the object metadata into the repository.

NOTE: The process of importing the object metadata into the repository could use a large amount of memory and can be time-consuming on the machine that runs the import. For this reason, do not leave the process, using the PowerCenter Repository Manager, unattended. If you want to use an unattended process, please use the command line approach described in the following section.

To import the object metadata into the repository using the Repository Manager

1. Open Repository Manager and connect to the repository.
2. Open the source-specific configuration folder.

   For example, open the Configuration for PeopleSoft 8.4 folder.
3. On the Repository menu, click Import Objects.

   The Import Wizard appears.
4. Click Browse, and open your XML file in the `$Siebel\ETL\Applications` folder

   where `$Siebel` is the path of the Siebel Customer-Centric Enterprise Warehouse installation directory.

5 Click Next, click Add All, and then click Next.

NOTE: Click the Add All and not the Add option.

6 In the Match Folders screen, match the folders in your XML file to the folders in your destination repository, and click Next.

7 Specify the rules to resolve object conflicts and click Next.

A Conflict Resolution Wizard window appears to resolve all conflicts.

8 Click Import.

9 Repeat the procedure for each application and XML file.

To import the object metadata into the repository using the command line

1 Navigate to the $Siebel\ETL\Applications folder, and open the application-specific Import Control XML file using Microsoft WordPad or Notepad.

2 Replace all occurrences of the TARGETREPOSITORYNAME with the repository name you entered in Creating and Configuring an Empty Siebel Business Analytics Repository on page 43.

For example, you see the following line in the XML file:

```
TARGETREPOSITORYNAME="<Put your Target Repository Name here>"/>
```

If your repository name is MY_REPOSITORY, then change the line to the following:

```
TARGETREPOSITORYNAME="MY_REPOSITORY"/>
```

Replace all other occurrences of TARGETREPOSITORYNAME with MY_REPOSITORY in the file.

3 Start a command window and navigate to the $pmclient folder where $pmclient is the path of the PowerCenter Client installation directory.

4 Type pmrep, and press Enter.

The pmrep command prompt appears.

5 Connect to the repository using the following command:

```
Connect -r <repository_name> -n <repository_user_name> -x <repository_password> -h
<repository_server_host_name> -o <repository_server_port_number>
```

A message confirming the connection with the repository appears.

6 Use the following command to import the metadata for an application:

```
ObjectImport -i <application_xml_file> -c <application_import_control_xml_file>
```

The application_xml_file is the application-specific XML file found in the $Siebel\ETL\Applications folder. The application_import_control_xml_file is the application-specific Import Control XML file found in the $Siebel\ETL\Applications folder.

7 Repeat Step 1, Step 2, and Step 6 for each application.

Backing Up the Development Repository

This task is a step in the Process of Creating the Development Repository on page 42.

The final step in creating your development repository is to back up the repository.

To back up the development repository

1. In the Administration Console, expand the Repositories node.
2. Select your repository.
3. On the Action menu, click All Tasks, and click Backup.

   The Backup Repository dialog box appears.
4. Enter the repository user name, password, and file name for the repository backup file.
5. (Optional) If you want to overwrite an existing repository backup file, choose to replace the existing file.
6. Click OK after completing all changes.

Moving the Required PowerCenter Server Data Files

This task is a step in the Process of Creating the Development Repository on page 42.

The PowerCenter data files you must move are currently located on the machine where you installed Siebel Customer-Centric Enterprise Warehouse.

For Windows systems, copy the folders as described in the following procedure. For UNIX systems, use FTP to access these files, and be sure you use binary mode.

To move the required PowerCenter Server data files

1. Navigate to the $Siebel\ETL\SrcFiles folder.
2. Move all the files to the `$pmserver\SrcFiles` folder.
3. Navigate to the `$Siebel\ETL\LkpFiles` folder.
4. Move all the files to the `$pmserver\LkpFiles` folder.

Creating the Data Warehouse Schema Control Tables

This task is a step in the Process of Creating the Development Repository on page 42.

This section describes how to create schema control tables so that the Siebel Customer-Centric Enterprise Warehouse can operate successfully.

If you already have a Siebel Relationship Management Warehouse (RMW), the following procedure modifies the existing date tables and adds columns required for the Siebel Customer-Centric Enterprise Warehouse. If you do not have a RMW, this step creates the necessary date tables.

If you already have a RMW, you need to drop the following tables before creating the data warehouse schema control tables:

- W_DAY_D
- W_MONTH_D
- W_QTR_D
- W_WEEK_D

The data in these tables have smart keys, so the link between the fact tables and the preceding dimension tables are not broken after dropping, recreating, and reloading the data into these tables.

The W_WEEK_D table in the RMW is used to store fiscal weeks. In the Siebel Customer-Centric Enterprise Warehouse, the W_WEEK_D table stores calendar weeks, and the W_FSC_WEEK_D table stores fiscal weeks. You need to change the Siebel Customer-Centric Enterprise Warehouse Repository where applicable.

NOTE: If you added extra columns to the these RMW tables as part of your customization, you need to add the columns again after you create Siebel Customer-Centric Enterprise Warehouse schema control tables.

To create the data warehouse schema control tables

1. Copy the `ddlimp` utility from the `$Siebel\Bin` folder.
2. Copy the schema control files from the `$Siebel\ETL\Applications` folder:
   - ddlsme_warehouse.ctl
   - ddlsme_staging.ctl
   - ddlsme_control.ctl
   - ddlsme_temp.ctl
3. Create a role in the database called `SSE_ROLE` with privileges to create objects.

   NOTE: In Oracle this equates to `CONNECT` and `RESOURCE` privileges.
4. Create an `ODBC DSN` entry in the database.
5. Open the command prompt window, and change the directory to the folder where the preceding files are copied.
6. Create the data model, including all the required tables for the Siebel Customer-Centric Enterprise Warehouse, by using the following command:

   ```
   ddlimp /s N /u <schema_user>  /p <schema_password> /c <ODBC connection string> /G
   "SSE_ROLE" /f ddlsme_warehouse.ctl /b "" /K "" /X "" /W N
   ```

   NOTE: For the Oracle ODBC connection using the Siebel Merant ODBC driver is the preferred option.

7 Repeat Step 6 for `ddlsme_staging.ctl`, `ddlsme_control.ctl`, and `ddlsme_temp.ctl`.

To populate the fields, you must run the Common Initialization Workflow. For more information on common initialization workflows, see About the Initialization Workflow on page 56.

Deploying Stored Procedures

This task is a step in the Process of Creating the Development Repository on page 42.

This section applies only to Siebel Supply Chain Analytics for Oracle 11i and the Siebel Enterprise Workforce Analytics applications for PeopleSoft 8.4.

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

You can deploy stored procedures by copying the stored procedure files from your Siebel Customer-Centric Enterprise Warehouse installation and deploying them to the target data warehouse.

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

To deploy stored procedures

1 Navigate to the `$Siebel\ETL\Stored_Procedure_Scripts` folder.

- For Oracle 11i, copy the source codes `Compute_Bounds_Ora11i.sql` and `PROC_PA_TASK_HIER.SQL` into the target data warehouse schema.
- For PeopleSoft 8.4, copy the source code `build_posn_sets.sql` into the target data warehouse schema.

2 Compile the stored procedures in the target data warehouse database.

NOTE: If you have problems deploying the stored procedures, see your database reference guide, or contact your database administrator.

Creating the Seed Data

This task is a step in the Process of Creating the Development Repository on page 42.

To create the seed data you must run some SQL statements on your data warehouse. These SQL statements insert a row with zero (0) as the primary key in the Dimension and Fact tables.

To create the seed data

1 Navigate to the `$Siebel\Etl\Applications` folder.

2 Open the `insertSQL.txt` file.

3 Run the SQL statements found in `insertSQL.txt` on your data warehouse.

4 Commit the changes to your data warehouse.

Moving the Required Siebel Customer-Centric Enterprise Warehouse Files

You need to move the `SiebelBusinessAnalytics.rpd` and `SiebelBusinessAnalytics.webcat` files for Siebel Customer-Centric Enterprise Warehouse to the Siebel Business Analytics Platform installation folder.

To move the Siebel Business Analytics repository file

1. Navigate to the `$Siebel\Repository` folder.
2. Move the `SiebelBusinessAnalytics.rpd` file to the `$SAHome\SiebelAnalytics\Repository` folder.

To move the Siebel Business Analytics Web Catalog file

1. Navigate to the `$Siebel\WebCatalog` folder.
2. Move the `SiebelBusinessAnalytics.webcat` file to the `$SAHome\SiebelAnalyticsData\Web\Catalog` folder.

   For more information on configuring Siebel Business Analytics Web Catalog, see *Siebel Analytics Web Administration Guide*.

5 Initializing and Populating the Siebel Customer-Centric Enterprise Warehouse

This chapter outlines the tasks you must perform to prepare your data warehouse before performing your initial data capture.

It contains the following topics:

Loading and Refreshing the Siebel Customer-Centric Enterprise Warehouse

This section describes the workflows to run when loading and refreshing the Siebel Customer-Centric Enterprise Warehouse.

Loading the Siebel Customer-Centric Enterprise Warehouse

1. **The initialization workflow.** For more information on the initialization workflow, see About the Initialization Workflow on page 56 and Initialization Workflow Files on page 57.
2. **The source-specific initialize workflow, if applicable to your module**. For more information on the source-specific initialize workflow, see About Working with Workflows on page 55
3. **The source-specific main workflow.** For more information on the source-specific main workflow, see About the Source-Specific Main Workflows on page 60.
4. **The post-load workflow.** The post-load processing initial workflow or the post-load processing workflow, whichever is in your configuration folder. For more information on the post-load processing workflow, see About Working with Post-Load Processing Workflows on page 65.

Refreshing the Siebel Customer-Centric Enterprise Warehouse

1. **The source-specific main workflow.** or more information on source-specific main workflow, see About the Source-Specific Main Workflows on page 60.
2. **The post-load workflow.** The post-load processing incremental workflow or the post-load processing workflow, whichever is in your configuration folder. For more information on the post-load processing workflow, see About Working with Post-Load Processing Workflows on page 65.

Database Types for Data Warehouse Objects

PowerCenter supports database independence. It is the database type that you specify in the Workflow Manager connection that determines the database type that PowerCenter Server uses. A target table that is displayed as the default, DB2, can actually be in SQL Server, DB2, or Oracle, depending on the Workflow Manager connection information.

Sources and targets found in the Siebel Customer-Centric Enterprise Warehouse folder are usually defined with the database type DB2. The source-specific folders in the Designer navigator window also contain sources and targets defined for databases other than what you actually run. However, you must specify your database type in Workflow Manager. The only limitation is that a single Source Qualifier needs to read from all the tables (with the same database type) that are defined in a mapping.

Modifying the Repository Connections

Your server configuration requires that the following database connections be configured to reflect the actual database type that your system is running:

- **Source.** IA_[SOURCE ABBREVIATION]_SOURCE
- **Staging Area.** IA_[SOURCE ABBREVIATION]_STAGE

- **Data Warehouse.** `IA_[SOURCE ABBREVIATION]_WAREHOUSE`

  NOTE: If you have more than one source, you must modify the connection for each flat file or database.

Configuring these database connections also helps to make sure that Siebel Customer-Centric Enterprise Warehouse is installed correctly. Configure all three repository connections listed in the preceding paragraph for each of your source types.

Each source type requires a relational database connection, or an application database connection, or both. Relational and application connections have been defined for the following:

- Oracle applications with an Oracle database connection
- PeopleSoft applications with an Oracle database connection
- SAP R/3 with a source connection
- Warehouse with an Oracle database connection
- Staging with an Oracle database connection

If you want to use these connections, you can edit their properties. Otherwise, you must create a new connection. For a list of source types and their connection configuration, see Table 7.

Table 7. Source Types and the Database Connection Configuration

Source Type	Database Connection	Connection Type
Common	IA_WAREHOUSE	Relational
PeopleSoft 8.4	IA_PSFT8_SOURCE IA_PSFT8_STAGE IA_PSFT8_WAREHOUSE	Relational and application
Oracle 11i	IA_ORA11i_SOURCE IA_ORA11i_STAGE IA_ORA11i_WAREHOUSE	Relational
SAP R/3	IA_SAP_SOURCE IA_SAP_STAGE IA_SAP_WAREHOUSE	Application

To configure the relational database connections

1. Start the Workflow Manager.
2. On the Connections menu, click Relational....
3. Connect to your repository.

4. Click your target database in the Relational Connection Browser.

   NOTE: If you want to change the database type, you must:

   a. Click Edit....

   b. Click your database type, and click OK.

5. Click New....

6. In the Connection Object Definition box, type the appropriate Database Name, User Name, Password, Connect String, and Code Page, and click OK.

7. Repeat this procedure for each connection you want to configure:

   - Source
   - Staging area
   - Data warehouse

8. Click Close.

To configure the application database connections

1. Start Workflow Manager.

2. Connect to your repository.

3. On the Connections menu, click Application....

4. Click your target database in the Application Connection Browser.

5. Click New....

6. In the Connection Object Definition box, type the appropriate Database Name, User Name, Password, Connect String, and Code Page for your source connection.

7. Click OK, and then click Close.

To replace relational database connections

1. Start Workflow Manager.

2. Connect to your repository.

3. On the Connections menu, and click Replace Connections.

4. Click Add a New Connection Pair.

5. Enter your source database connection in the From field, and your new source database connection in the To field.

6. Click Replace.

NOTE: You cannot replace or copy an application database connection. You have to delete the original application connection, and create a new connection with the same name. For example, if you want to extract from a PeopleSoft DB2 database, you have to delete the existing PeopleSoft Oracle application connection called, IA_PSFT8_SOURCE. Create a new application connection using PeopleSoft DB2, with the same name, IA_PST8_SOURCE.

Optimizing Database Performance

After setting up your database connections, but before loading any data, you may want to optimize your repository's database performance. To perform optimization, request your Database Administrator to run the appropriate, update-statistics operation for your RDBMS on all tables and indexes.

The optimization procedure updates information about the distribution of key values, and improves the performance of operations (for example, starting the workflow or opening mappings). It is also recommended that you run an update, statistics operation on both your source and target databases.

When the repository has been restored, and you are satisfied with your setup, you are then ready to run the Siebel Customer-Centric Enterprise Warehouse initialization workflows. For more information about initializing workflows, see About Working with Workflows on page 55.

About Working with Workflows

Not all the data required by the Siebel Business Analytics is extracted from your source. To set up the Siebel Customer-Centric Enterprise Warehouse correctly, you may want to supplement your source data with other information. For such additional data, the Siebel Business Analytics provide `*.csv` and `*.txt` files that are used to populate dimension tables. These files are placed in your `$Siebel\ETL\Siebel Warehouse Support File\SrcFiles` directory as an automatic part of the installation. The load workflows that you see listed in the Workflow Manager directory use these prepackaged files.

Some of these prepackaged files require no interaction from you. They are already packaged with the data that has to be loaded. Several prepackaged files are required by the Siebel Business Analytics to properly manage your data, and are initialized when you run the common initialization workflows. However, some of the prepackaged files do require further information that is specific to your data warehouse, so you can edit these files as described in this chapter.

Table 8 lists the Siebel Customer-Centric Enterprise Warehouse Workflow Names.

Table 8. Workflow Names for Siebel Customer-Centric Enterprise Warehouse

Functional Workflow Name	Logical Workflow Name	Folder Name
Initialization Workflow	INITIALIZE	Siebel
Source-Specific Initialize Workflow	[SOURCE]_[SUBJECT]_Application_Initialize	Source Application
Source-Specific Main Workflow	[SOURCE]_[SUBJECT]_Application	Source Application
Post-Load Processing Workflow	[PLP]_[SUBJECT]_Application	Post Load Processing
Post-Load Processing Initial Workflow	[PLP]_[SUBJECT]_Application_INIT	Post Load Processing
Post-Load Processing Incremental Workflow	[PLP]_[SUBJECT]_Application_INCR	Post Load Processing

About the Initialization Workflow

Running the common initialization workflow, INITIALIZE, verifies that both the PowerCenter and the Siebel Customer-Centric Enterprise Warehouse are installed successfully. Running the INITIALIZE workflow also initializes the dimension tables, and verifies that your date tables are configured correctly.

The common initialization workflow must be run only once, but it is run before any of the other workflows. The INITIALIZE workflow contains prepackaged files that are required by your system, regardless of your source type, to use the Siebel Business Analytics installation.

The common initialization workflow contains four worklets and one stand-alone workflow:

- Z_INITIAL
- Z_LOAD_DATES_TIME
- Z_LOAD_DATES_AGGR
- Z_LOAD_DATES INIT
- S_M_Z_PARAMETERS_FILE_LOAD

These worklets, nested within the initialization workflows, write results to the target tables specified in Table 9. A *worklet* is an object that represents a set of tasks. It allows you to reuse a set of workflow instructions in several workflows. Table 9 lists the common initialization worklets and the target tables.

Table 9. Common Initialization Worklets and Target Tables

Workflow or Worklet Name	Task Name	Target Table Name
Z_INITIAL	S_M_Z_INITIAL_LOAD_1	IA_DATES
		IA_CAL_MONTHS
		IA_CAL_WEEKS
		IA_CAL_QTRS
		IA_CAL_YEARS
		IA_FSC_MONTHS
		IA_FSC_WEEKS
		IA_FSC_QTRS
		IA_FSC_YEARS
Z_LOAD_DATES_TIME	S_M_Z_TIME_OF_DAY_LOAD	IA_TIME_OF_DAY
	S_M_Z_TIME_OF_DAY_AGGR	IA_HOUR_OF_DAY
	S_M_Z_DATES_LOAD	IA_DATES
Z_LOAD_DATES_AGGR	S_M_Z_DATES_AGGR	IA_CAL_MONTHS

Table 9. Common Initialization Worklets and Target Tables

Workflow or Worklet Name	Task Name	Target Table Name
		IA_CAL_WEEKS
		IA_CAL_QTRS
		IA_CAL_YEARS
	S_M_Z_DATES_FSC_AGGR	IA_FSC_MONTHS
		IA_FSC_WEEKS
		IA_FSC_QTRS
		IA_FSC_YEARS
	S_M_Z_DATES_FSC_AGGR	IA_DATES
	S_M_Z_SYSDATE_CREATION	TZ_DATES_GENERIC
Z_LOAD_DATES	SIL_DayDimension	W_DAY_D
	SIL_DayDimension_GenerateRow1	W_DUAL_G
	SIL_DayDimension_GenerateRow2	W_DUAL_G
	SIL_DayDimension_GenerateRow3	W_DUAL_G
	SIL_DayDimension_GenerateRow4	W_DUAL_G
	SIL_DayDimension_GenerateRow5	W_DUAL_G
	SIL_DayDimension_GenerateRow6	W_DUAL_G
	SIL_DayDimension_GenerateRow7	W_DUAL_G
	SIL_DayDimension_GenerateSeed	W_DUAL_G
	SIL_DayDimension_Unspecified	W_DAY_D
	SIL_FiscalMonthDimension	W_FSC_MONTH_D
	SIL_FiscalWeekDimension	W_FSC_WEEK_D
	SIL_MonthDimension	W_MONTH_D
	SIL_QuarterDimension	W_QTR_D
	SIL_WeekDimension	W_WEEK_D
	SIL_FiscalQuarterDimension	W_FSC_QTR_D
	SIL_FiscalYearDimension	W_FSC_YEAR_D
	SIL_YearDimension	W_YEAR_D
	SIL_DayDimension_CleanSeed	W_DUAL_G

Initialization Workflow Files

The following files are found in the common initialization workflow, INITIALIZE:

- timespan.txt
- dates.txt

You modify the timespan.txt and the dates.txt files. These files reflect your data warehouse time span. Do not modify the other files in the INITIALIZE workflow, because the system requires those default values during installation.

Timespan.txt File

Required by the INITIALIZE workflow. Open the timespan.txt, edit the $$START_DATE, and the $$END_DATE.

Time.txt File

Required by the Z_LOAD_DATES_TIME worklet. The time.txt file contains references to the time_am.csv, and time_pm.csv files. Do not modify the file.

Time_am and Time_pm.csv Files

Required by the Z_LOAD_DATES_TIME worklet. The time CSV files contain either the A.M. or the P.M. hours of the 24-hour day. Do not modify these files. Each of the two comma-separated files contains a 12-hour segment of the day, broken down by minute and second, and mapped to a description such as night, dawn, and morning for the A.M. hours, or afternoon, evening, and night for the P.M. hours.

Dates.txt File

Required by the INITIALIZE workflow, the dates.txt file contains pointers to each of the Dates_xxxx_xxxx CSV files. You must configure this file to match your data warehouse requirement for a time horizon.

Example of Dates.txt File Use

For example, imagine that today is March 3, 2004 and you want to create a data mart that contains your data from January 2004 through December 2012. In your dates.txt file, you reference the dates_xxxx_xxxx CSV files (see the next paragraph) that define your desired time horizon by removing the entries for the files that are not needed. In this example, you must reference two files (dates_2000_2009.csv and dates_2010_2019.csv), so you remove the entries for the other Dates_xxxx_xxxx CSV files that do not match your time horizon.

Dates_XXXX_XXXX.csv Files

Required by the Z_LOAD_DATES_TIME worklet. The date CSV files are used as described under the heading, Dates.txt. Do not modify the Dates_xxxx_xxxx.csv files.

Example of Using the Dates_XXXX_XXXX.csv Files

In this example, the Xs are a placeholder for the decade start and end dates (such as 1970–1979). These files are divided by decades. For example, the 1970s decade is 1970-1979. The files are optional, because you choose the appropriate decade files, depending on the time horizon you want to represent in your data warehouse. (For example, if you want to represent 1990 through 2009, you keep the Dates_1990_1999.csv and the Dates_2000_2009.csv files in your Dates.txt file, thus defining your time horizon.) Each comma-separated file contains every day of the decade, mapping it to weeks, months, quarters, and years. Each file also contains other descriptive information for each year, such as day of week, end or beginning of month flags, and so on.

Loading Your Fiscal Calendar

The INITIALIZE Workflow contains worklets to load information about your company's fiscal calendar, holidays, and other date-related concepts. If you want to load your Fiscal Calendar, you must configure the INIALIZE workflow before you can run the Fiscal Calendar.

The dates dimension (IA_DATES) contains fields that you can configure to store fiscal calendar information. By default, the fields that contain the fiscal information are populated with calendar information. Therefore, if your fiscal calendar differs from the standard calendar, you must configure one of two prepackaged CSV files (fiscal_months.csv, and fiscal_week.csv). Your fiscal calendar must be set up such that fiscal weeks roll up to fiscal months, which roll up to fiscal quarters, which roll up to fiscal dates. By default, the Z_UPDATE_FISCAL_DATES_BY_WEEKS workflow is disabled, and you load the fiscal calendar by running the Z_UPDATE_FISCAL_DATES_BY_MONTHS.

The explanations that follow help you to determine your system needs, and to choose an appropriate option, based on how much information you want to specify.

Configuring Your Fiscal Month

If you want to specify the beginning date of each fiscal month within a fiscal year, you may do so in the fiscal_months.csv file. The Z_UPDATE_FISCAL_DATES_BY_MONTHS worklet uses the fiscal_months.csv file as source information to calculate the appropriate fiscal week information, within each month that is specified. Run the worklet to update each record in the IA_DATES table for the period that you specify.

Configuring Your Fiscal Week

If your fiscal week information cannot be described using the methods in the preceding sections, you may use the fiscal_week.csv file. You can specify the start date for each week, and indicate which fiscal week, month and year each week is in, with the fiscal_week.csv file. After the file has been modified, enable the Z_UPDATE_FISCAL_DATES_BY_WEEKS worklet, and disable the Z_UPDATE_FISCAL_DATES_BY_MONTHS worklet in the INITIALE workflow. The Z_UPDATE_FISCAL_DATES_BY_WEEKS worklet calculates only information within the parameters that you specified, such as the day in the fiscal week.

Using the Fiscal_months.csv and the Fiscal_weeks.csv Files

There are two fiscal calendar files, one of which you have to modify. The two fiscal calendar files are used by the Z_UPDATE_FISCAL_DATES_BY_MONTHS or Z_UPDATE_FISCAL_DATES_BY_WEEKS worklets. The contents of these files must represent your company fiscal month or week. You can edit the files by entering the correct start and end dates for your company's fiscal month, week, or year.

To modify the fiscal_weeks.csv file

1. Open the `fiscal_weeks.csv` file in the `$Siebel\ETL\SrcFiles` folder.
2. Following the format specified in the spreadsheet, start with row six and enter the number of your fiscal year in column A, month in column B, and week in column C.

   Enter the start date of the fiscal week for a given year in column D.
3. Repeat the numbering process for each week in the fiscal year.

   Be sure to identify the correct month for each week.
4. Select File > Save, and close the file.

Follow the preceding procedure to edit the `fiscal_months.csv` file used by the Z_UPDATE_FISCAL_DATES_BY_MONTHS worklet to configure your fiscal calendar.

About the Source-Specific Main Workflows

This section allows you to prepare your source-specific main workflow so that it meets your implementation requirements. In this source-specific main workflow, the Siebel Business Analytics provide you with the logic to extract all the data from a particular source system. Depending on your source system, the workflow contains worklets for each Siebel Business Analytics product.

You may want to remove some worklets that are not part of your implementation, depending on the module or product family you want to use. After configuring your main workflow, run it to capture your first, complete extraction of all the rows in your source database. The result provides you with a populated target schema that you can use for further configuration.

Most of the source-specific main workflows that you see in the Navigation window of the Workflow Manager contain nested worklets.

These usually include:

- Prepare worklet
- Fact extract worklet
- Dimension extract worklet
- Dimensions load worklet
- Facts load worklet
- Execution finished worklet

About Choosing Which Rows To Keep in the Data Warehouse

The purpose of the primary extract sessions is to track any physical deletions that occur in the source system. Primary extract sessions do not perform a full data extract; they extract only the KEY_ID and SOURCE_ID for facts. The delete sessions delete the rows that are in the data warehouse.

If rows are physically removed from your source system, you must make a choice about retaining the rows in your data warehouse:

- If you want to retain the rows in the data warehouse even though the rows are removed from the source system, then keep the default primary extract sessions, and the corresponding delete session disabled.
- If you do not want to retain the rows in the data warehouse after they are removed from the source system, then enable the default primary extract sessions, and the corresponding delete session.
- If your source system archives rows, you may want to set a parameter to search for archive dates, and then execute the delete session only on rows that have been archived and are no longer needed in the warehouse. See the discussion on deletion configuration for source-archived records, in About Working with Primary Extract and Delete Mappings on page 126.
- The PRE_D sessions for aggregate tables are disabled. If you enable the primary extract sessions and the corresponding delete sessions, you should also enable the corresponding PRE_D sessions.

About Modifying Session Parameters for Initial and Incremental Loads

Parameters are designed to regulate the behavior of workflows and tasks, and are set using a source file. Siebel Customer-Centric Enterprise Warehouse supplies this file. However, you must modify the appropriate CSV parameters source file to set up your ETL process frequency. For a list of parameter source files for every source application folder see Table 10.

Table 10. Source Application Folders and Parameter Source Files

Source Application Folder	Parameter Source File
Configuration for Oracle Applications v11i	file_parameters_ora11i.csv
Configuration for PeopleSoft 8.4	file_parameters_psft84.csv
Configuration for SAP R/3	file_parameters_sapr3.csv
Configuration for Universal Source	file_parameters_univ.csv
Configuration for Post Load Processing	file_parameters_plp.csv

Table 11 is an example of the `file_parameters_psft84.csv` parameter file. It shows a number of important columns and example sessions for the `file_parameters_psft8.csv` parameter file.

Table 11. Example of the file_parameters_psft84.csv Parameter File

PARM_TYPE	PARM_CODE	PARM_NVALUE_1	PARM_SVALUE_1	PARM_SVALUE_3	PARM_DVALUE_1	Source_ID
PSFT84	S_M_P_AP_XACTS_VCHR_ACCTG_LINE_GL_REFERENCE_EXTRACT:LAST_EXTRACT_DATE	0	D	MPLT_BCP_AP_XACTS_VCHR_ACCTG_LINE_GL_REFERENCE_EXTRACT	200400101000000	PSFT84
PSFT84	S_M_P_AR_XACTS_BI_ACCT_ENTRY_GL_REFERENCE_EXTRACT:LAST_EXTRACT_DATE	0	D	MPLT_BCP_AR_XACTS_BI_ACCT_ENTRY_GL_REFERENCE_EXTRACT	20040101000000	PSFT84
PSFT84	S_M_P_AR_XACTS_ITEM_DST_GL_REFERENCE_EXTRACT:LAST_EXTRACT_DATE	0	D	MPLT_BCP_AR_XACTS_ITEM_DST_GL_REFERENCE_EXTRACT	20040101000000	PSFT84
PSFT84	S_M_P_GL_COGS_CM_ACCTG_LINE_GL_REFERENCE_EXTRACT:LAST_EXTRACT_DATE	0	D	MPLT_BCP_GL_COGS_CM_ACCTG_LINE_GL_REFERENCE_EXTRACT	20040101000000	PSFT84

The following are the important columns in the parameter file:

- **PARM_TYPE.** The `PARM_TYPE` column represents the source system you are working on. For example, PeopleSoft 8.4 has a value of `PSFT80`.
- **PARM_CODE.** The `PARM_CODE` column contains the name of the session concatenated with the session parameter name. You have to configure this value. For example, in both the initial and incremental extracts mappings in Table 11, you use the `LAST_EXTRACT_DATE` parameter.
- **PARM_NVALUE_1.** The `PARM_NVALUE_1` column represents the number of days you want to use as your extraction window. For your initial loads, the value is 0. For incremental loads with a window of 4 days from the current system date, the value is 4. For more information on configuring these dates, see Configuring the Date Parameters for Parameter Files on page 63.
- **PARM_SVALUE_1.** The `PARM_SVALUE_1` column represents the type of parameterization being done. For example, when configuring dates the value is D.
- **PARM_SVALUE_3.** The `PARM_SVALUE_3` column represents the mapplet name. For Siebel Business Analytics, most of the extraction logic is hidden in the business component mapplet. The mapplet name corresponds to the PARM_CODE column.
- **PARM_DVALUE_1.** The `PARM_DVALUE_1` column represents the date that is used for initial runs. The value from this column is used only when PARM_NVALUE_1 has a non-zero value.

- **SOURCE_ID.** The SOURCE_ID always represents the source system you are working on. For example, PeopleSoft 8.4 has a value of PSFT84.

Configuring the Date Parameters for Parameter Files

If you use Table 11 for the initial run, the PARM_DVALUE_1 parameter has a value of January 01, 2004. If you extract changed data from 2004, then you are extracting the entire data set. If you have older data, change the value of PARM_DVALUE_1 in the format yyyymmddhh24miss.

If you want incremental runs with a two-day window for the first two sessions, you change only the PARM_NVALUE_1 column for the corresponding sessions. If you want incremental runs with a three-day and four-day window for the third and the fourth sessions, you also change the PARM_NVALUE_1 column for the corresponding sessions. When PARM_NVALUE_1 has a nonzero value, PARM_DVALUE_1 is not used. For an example of how your parameter file (file_parameters_psft8.csv) would appear for incremental loads, see Table 12.

Table 12. Example of the file_parameters_psft8.csv Parameter File for an Incremental load

PARAM_TYPE	PARAM_CODE	PARAM_NVALUE_1	PARAM_SVALUE_1	PARAM_SVALUE_3	PARAM_DVALUE_1	SOURCE_ID
PSFT84	S_M_P_AP_XACTS_VCHR_ACCTG_LINE_GL_REFERENCE_EXTRACT:LAST_EXTRACT_DATE	2	D	MPLT_BCP_AP_XACTS_VCHR_ACCTG_LINE_GL_REFERENCE_EXTRACT	20040101000000	PSFT84
PSFT84	S_M_P_AR_XACTS_BI_ACCT_ENTRY_GL_REFERENCE_EXTRACT:LAST_EXTRACT_DATE	2	D	MPLT_BCP_AR_XACTS_BI_ACCT_ENTRY_GL_REFERENCE_EXTRACT	20040101000000	PSFT84
PSFT84	S_M_P_AR_XACTS_ITEM_DST_GL_REFERENCE_EXTRACT:LAST_EXTRACT_DATE	3	D	MPLT_BCP_AR_XACTS_ITEM_DST_GL_REFERENCE_EXTRACT	20040101000000	PSFT84
PSFT84	S_M_P_GL_COGS_CM_ACCTG_LINE_GL_REFERENCE_EXTRACT:LAST_EXTRACT_DATE	4	D	MPLT_BCP_GL_COGS_CM_ACCTG_LINE_GL_REFERENCE_EXTRACT	20040101000000	PSFT84

To modify the CSV parameter file

1 Navigate to the appropriate CSV parameter file.

For example, for Oracle 11i navigate to $pmserver\SrcFiles\file_parameters_ora11i.csv. For a list of parameter files, see Table 10.

2. Copy your CSV file to the SRCFiles folder under the PowerCenter Server folder.
3. Open the file using Microsoft WordPad or Notepad.
4. Search for the column labeled `PARM_DVALUE_1`.
5. Change the default date (197001010000) to the date you require.
6. Change your new date in every session you want to run incrementally.
7. Save your file.

Configuring the Database Parameter for the Source System

You must configure your database parameter for your source system in the `M_Z_SESS_PAR_FILE_GENERATE` mapping in the Siebel Applications folder.

NOTE: If you are using the Configuration for Oracle 11i and the Configuration for SAP R/3 folders, then you do not have to configure the database parameter. For SAP R/3, the extracts are done using ABAP codes, and the initial and incremental filters do not depend on any specific back-end database. For Oracle 11i, you use the ORACLE RDBMS.

To configure the database parameter for the source system

1. Start the Designer.
2. Connect to your repository.
3. Open the Siebel Applications folder.
4. Open the `M_Z_SESS_PAR_FILE_GENERATE` mapping with Mapping Designer.
5. Double-click the EXP_PARAMETERS Expression transformation to open the Edit Transformation box.
6. Click the Ports tab.
7. Set the expression for the port DATABASE_NAME_VAR with the database value you are using for your source system:
   - If the source system database is Oracle, set the value to ORACLE.
   - If DB2, set the value to DB2.
   - If MS SQL SERVER, set the value to MSSQL.
8. Validate the mapping and save the repository.

Configuring the Database Parameter for Post-Load Processing

You must configure your database parameter for your target data warehouse database system in the M_PLP_SESS_PAR_FILE_GENERATE mapping in the Configuration for Post Load Processing folder.

To configure the database parameter for the target data warehouse database system

1. Start the Designer.
2. Connect to your repository.
3. Open the Configuration for Post Load Processing folder.
4. Open the M_PLP_SESS_PAR_FILE_GENERATE mapping with Mapping Designer.
5. Double-click the EXP_PARAMETERS Expression transformation to open the Edit Transformation box.
6. Click the Ports tab.
7. Set the expression for the port DATABASE_NAME_VAR with the database value you are using for your target data warehouse database system:
   - If the source system database is Oracle, set the value to ORACLE.
   - If DB2, set the value to DB2.
   - If MS SQL SERVER, set the value to MSSQL.
   - If Teradata, set the value to TERADATA
8. Validate the mapping and save the repository.

About Working with Post-Load Processing Workflows

Regardless of which workflow you use to execute your data captures, there is one feature, post-load processing, which requires execution for your initial data load to operate successfully. This section outlines steps for working with the post-load processing feature.

The post-load processing workflows search for an indicator file before they start running. This indicator file is created by the Execution Finished Worklet, and is called, file_plp_<suite name>.ind.

TIP: If you want the post-load processing workflow to start as soon as your fact loads complete, schedule these two workflows at the same time. The post-load processing workflow starts as soon as the indicator file appears in its directory.

NOTE: The Siebel Enterprise Workforce Analytics applications have a post-load processing initial workflow and a post-load processing incremental workflow. Run the initialization workflow the first time you load your data warehouse, and run the incremental workflow to refresh your data warehouse.

Restarting the Extract, Transform, and Load Process

This section provides guidelines on what to do in the event of an Extract, Transform, and Load (ETL) failure. For example, the ETL process fails if a required file is missing or the PowerCenter Server is shut down because of memory problems. You can restart the ETL process in various workflows and worklets—Initialize, Prepare, Extract Facts, Extract Dimensions, Load Dimensions, Load Facts, and Post-Load Processing.

NOTE: This section assumes that your data warehouse is already in production and you have taken care of any possible data issues.

Scenario for a Workflow in Suspended Mode After a Task Fails

You can resume a workflow after investigating and resolving the issue that causes the suspended workflow. For example, a workflow fails because of a missing source file. In this scenario, you can provide the missing file and resume the workflow.

However, some workflow failures require changes to the session properties. You need to make the required changes to the session properties, save the changes in the repository, and resume the suspended workflow. If the workflow fails with your new changes, abort the workflow, and restart the workflow from the task.

Scenario for a Workflow In Failed Mode and the Shut Down of the PowerCenter Server

You can restart the PowerCenter Server, and restart the workflow from the task, after investigating and resolving the issue that causes the workflow to fail. For example, a workflow fails and the PowerCenter Server is shut down because of memory problems.

To resume a workflow

1. Start the Workflow Monitor.
2. Connect to your repository.
3. Right-click on the suspended workflow, and click Resume Workflow.

To restart a workflow from a task

1. Start the Workflow Monitor.
2. Connect to your repository.
3. Expand the failed workflow, and select the failed task.
4. Right-click on the task, and click Restart Workflow From Task.

   If you have multiple session failures in the same level, you need to click Restart Task for each session. Click Restart Workflow From Task for the last session.

About Improving the Query Performance of Siebel Customer-Centric Enterprise Warehouse

After the initial ETL load you can run a script to create indexes and improve the query performance. These indexes are helpful when running queries against the Siebel Customer-Centric Enterprise Warehouse. These indexes can be created after the full ETL load as they are not required during the ETL load.

The Siebel Business Analytics Installer creates three folders—DB2UDB, Oracle, and MSSQL. In each folder there is a subfolder called Query Performance. In each Query Performance folder there is a file called create_indexes_<db_platform>.sql. After the initial ETL load, run the script to create all additional indexes to enhance the query performance of your Siebel Customer-Centric Enterprise Warehouse. You can enhance the script by adding parallel statements for database servers that have multiple processes. Also, the Database Administrator can split the script into multiple scripts and run them in parallel.

After creating these indexes the Database Administrator runs the appropriate, update-statistics operation for the customer RDBMS to update the statistics on the index statements.

For Oracle, the script creates a bitmap index on every foreign key in a fact table. These bitmap indexes enhance the performance during the query time but they are slow to update during data inserts or updates. It is recommended that you drop these indexes before the ETL incremental run and recreate the indexes after the run. You can create a PowerCenter mapping to call a drop index script and another mapping to call a create index script.

It is possible that the bitmap index on a foreign key could prevent the optimizer from using the multicolumn index. This happens when there is a filter on some dimension, which the optimizer assumes would reduce the number of selected fact records. If this reduction does not happen, then the foreign key index can slow the query. The Database Administrator can delete or deactivate the index.

You can build more aggregates on the large fact tables to reduce the number of indexes and increase the report performance. You need to compare the time taken for updating the indexes on a fact table with the time required to update the aggregate table during the ETL incremental run.

For more information on the general guidelines for setting up the Siebel Data Warehouse, see *Siebel Analytics Applications Installation and Administration Guide*.

About Analyzing Data Warehouse Tables for Optimal ETL Performance

The Siebel Customer-Centric Enterprise Warehouse loads data into the data warehouse tables through a series of extraction, transformation and loading (ETL) steps. The set of data warehouse tables depends on the application and the data source adapter that the you have purchased.

For the ETL process to work efficiently, you need to analyze the target database tables and compute statistics on them during the ETL process. This is a critical step to avoid significant delays in the ETL process.

The preconfigured workflows you see in the Siebel Customer-Centric Enterprise Warehouse depends on the application and the data source adapter that you have purchased. Each workflow contains a set of worklets, and each worklet contains a set of sessions, each of which loads one or more target tables. These target database tables need to be analyzed for the database to use the most efficient plan for running a SQL statement that is generated by the ETL process.

About the Table Analyze Utility

Most databases have a feature to analyze and gather statistics for a table. By analyzing a table, one can store the key distribution, index distribution, average row size, and a lot more other useful information related to the content of the table.

The Siebel Customer-Centric Enterprise Warehouse uses the Table Analyze Utility to analyze tables after they are loaded. After the data is loaded into a table, the Table Analyze Utility uses an analyze table command for the specific table in the correct database. If the next session uses the previously analyzed table as a source and pushes SQL to the database using this table as a joined table, then the database knows that statistics are available for the table and it can produce the most effective execution plan immediately. The Table Analyze Utility improves the overall ETL performance.

Process of Configuring the Table Analyze Utility

To configure the Table Analyze Utility, perform the following tasks:

- Setting the Environment Variable for the JDBC drivers on page 68
- Setting the Database Connection Information for the Table Analyze Utility on page 69
- Setting the Program Parameters for the Table Analyze Utility on page 70
- Moving the Table Analyze Utility Files on page 71
- Setting the Environment Variable for the OctopusUtils.jar File on page 71
- Creating the Encrypted Password File for the Table Analyze Utility on page 72
- Configuring the Table Analyze Utility for UNIX on page 72
- Running the Table Analyze Utility on page 73

Related Topic

- About Creating Command Tasks on page 74

Setting the Environment Variable for the JDBC drivers

This task is a step in the Process of Configuring the Table Analyze Utility on page 68.

You need to add the path of your target database's JDBC drivers to the CLASSPATH environment variable. Depending on what target warehouse database you use, the JDBC drivers needs to be installed on the PowerCenter Server machine:

- **Oracle.** If you are using an Oracle database (other than 8.x), find the directory where Oracle is installed. The JDBC driver is named `ojdbc14.jar` in the `jdbc\lib` directory.

  If you are using Oracle 8.x, the JDBC driver file is named `classes12.zip`.

- **DB2**. If you are using a DB2 database, find the directory where DB2 is installed. The JDBC driver is named `db2java.zip` in the Java subdirectory.
- **MSSQL**. If you are using an MSSQL database, download the SQL Server JDBC drivers from Microsoft's Web site. The JDBC drivers are named `msbase.jar`, `mssqlserver.jar`, and `msutil.jar`.
- **Teradata**. If you are using a Teradata database, find the directory where Teradata is installed. The JDBC drivers are named `terajdbc4.jar`, `log4j.jar`, and `tdgssjava.jar`. Depending on the Teradata JDBC version, you may not have `log4j.jar` and `tdgssjava.jar`.

To set the CLASSPATH environment variable for the JDBC drivers

- In the Windows System Properties window, add to the CLASSPATH environment variable the path for the JDBC driver directory and the full file name.

Setting the Database Connection Information for the Table Analyze Utility

This task is a step in the Process of Configuring the Table Analyze Utility on page 68.

You need to set the database connection information before you execute the Table Analyze Utility program.

To set the database connection information

1. On the server where you installed the Siebel Business Analytics, navigate to the `$Siebel\Etl\AnalyzeUtility` folder.
2. Open the `database.properties` file using Microsoft WordPad or Notepad.
3. Edit fields in the database.properties file depending on your database type, as shown in the following table.

Field	Oracle (OCI8)	Oracle (Thin)	DB2	MSSQL	Teradata
DatabaseUsername	A database user with permission to read and write.				
JDBCTableowner	Name of the table owner.				
JDBCDBType	Oracle (OCI8)	Oracle (Thin)	DB2	MSSQL	Teradata
JDBCConnString	TNS name	Service name or SID	Connection string	Database name	Database name

Field	Oracle (OCI8)	Oracle (Thin)	DB2	MSSQL	Teradata
JDBCHost	None	Server host	None	Server host	Server host
JDBCPort	None	Server port	None	Server port	Server port

Setting the Program Parameters for the Table Analyze Utility

This task is a step in the Process of Configuring the Table Analyze Utility on page 68.

You need to set the program parameters information before you execute the Table Analyze Utility program.

To set the program parameters

1. On the server where you installed the Siebel Business Analytics, navigate to the `$Siebel\Etl\AnalyzeUtility` folder.
2. Open the database.properties file using Microsoft WordPad or Notepad.
3. Edit the program parameters, as shown in the following table.

Parameter	Description
logDirectory	The log directory. The file AnalazerLog is created in this folder. The program creates a new folder if one does not exist. If the folder is left empty no log files are created. If you are providing a log directory location, you need to add two back slashes instead of the one back slash. For example, on Windows: `C:\\Program Files\\Informatica\\PowerCenterServer7.1.2 \\Server\\TgtFiles\\AnalyzeLogs`
customSQL	The location of the customsql.xml. This file is used to customize statistics for the database tables. If you don't set this parameter, the customsql.xml is not read, and the default analyze syntax is used. If you are providing a custom statistics XML file location, you need to add two back slashes instead of the one back slash. For example, on Windows: `C:\\Program Files\\Informatica\\PiowerCenter Server 7.1.2 \\Server\\SrcFiles\\mycustomsql.xml`

Parameter	Description
ConcurrencyLevel	This parameter identifies how many tables are analyzed simultaneously. Note: parallelism is achieved only when you make multiple calls to utility. A set of tables specified in a call runs sequentially. **NOTE:** For DB2, check the concurrency level with the Database Administrator. If deadlocks occur, set the value to 1.
GatherStats	The value for this parameter is Y or N. You can set the value to N to avoid analyzing tables without removing the call to the program from the session. Otherwise set the parameter value to Y.
NumberOfRetries	This parameter identifies how many attempts is made to analyze a table before failing. If this parameter is not set, the default value of 10 is used.
Timeout	Time (in milliseconds) before the analysis call is considered to fail. No timeout is used if this parameter is not set.

Moving the Table Analyze Utility Files

This task is a step in the Process of Configuring the Table Analyze Utility on page 68.

You need to copy the Table Analyze files to the PowerCenter Server folder.

To move the Table Analyze Utility files

1. On the server where you installed the Siebel Business Analytics, navigate to the `$Siebel\Etl\AnalyzeUtility` folder.
2. Copy the following files to the `$pmrepserver\Server\SRCFiles` folder:
   - `OcotpusUtils.jar`
   - `database.properties`
   - `customsql.xml`

Setting the Environment Variable for the OctopusUtils.jar File

This task is a step in the Process of Configuring the Table Analyze Utility on page 68.

To create an encrypted password file, you would need to add the OctopusUtils.jar to the CLASSPATH environment variable.

To set the CLASSPATH environment variable for the OctopusUtils.jar

- In the Windows System Properties window, add to the CLASSPATH environment variable the path for the file `OctopusUtils.jar`, including the full file name.

  For example:

  ```
  $pmrepserver\Server\SRCFiles\OctopusUtils.jar
  ```

Creating the Encrypted Password File for the Table Analyze Utility

This task is a step in the Process of Configuring the Table Analyze Utility on page 68.

The password for the database user name is stored in an encrypted format in a separate file.

To create an encrypted password file

1. At the command prompt, navigate to the `$pmrepserver\Server\SRCFiles` folder.
2. Enter the following:
   - For Windows:

     ```
     %MY_JAVA_HOME%\java -cp %CLASSPATH% com.siebel.oc.utils.PasswordFile
     database.psw <your password>
     ```

   - For Unix:

     ```
     $MY_JAVA_HOME/java -cp $CLASSPATH com.siebel.oc.utils.PasswordFile database.psw
     <your password>
     ```

   This creates an encrypted password file called database.psw in the `$pmrepserver\Server\SRCFiles`.

NOTE: You need to modify the call in the analyze session if you use a file name different to `database.psw`. You can have multiple sets of database properties, password files, and `customsql.xml` files.

Configuring the Table Analyze Utility for UNIX

This task is a step in the Process of Configuring the Table Analyze Utility on page 68.

There is an additional configuring step when the PowerCenter server is installed on UNIX.

Configuring the Table Analyze Utility for UNIX

1. Open the Workflow Designer and connect to the repository.
2. Right click on SIEBEL_DW_SERVER and select Edit menu.
3. Click the Advanced button.

4 Navigate to the $PMSuccessEmailUser attribute name.

5 Change the attribute value:

```
%MY_JAVA_HOME%/java -cp %CLASSPATH% AnalyzeTables
```

to:

```
$MY_JAVA_HOME/java -cp $CLASSPATH AnalyzeTables
```

6 Click OK.

Running the Table Analyze Utility

This task is a step in the Process of Configuring the Table Analyze Utility on page 68.

Tables are run sequentially when using the Table Analyze Utility for a list of tables. Running multiple Table Analyze Utilities results in multiple tables being analyzed at the same time (if ConcurencyLevel is higher than 1).

The Table Analyze Utility uses the W_ETL_STAT_UTIL data warehouse table. There is a record created in the W_ETL_STAT_UTIL table for each analyzed table. If a table is analyzed daily, a new record is created for each day. When the Table Analyze Utility examines multiple tables, a record for each table is created, when all the previous tables are successfully analyzed.

If you want to cancel a request, find the analyzed table name and set the ABORT_FLG to Y in the TABLE_NAME column. The Table Analyze Utility cancels the table analysis and skips the remaining tables.

Table Analyze Utility returns an error code on completion. Table 13 lists the error codes and their descriptions.

Table 13. Error Codes and Descriptions

Code	Description
0	Success
401	Request aborted
402	Connection can't be established
403	SQL error
404	Encryption error
405	I/O error
406	Incorrect arguments
407	Requested table not found
419	Utility internal error
420	Unknown error

NOTE: Truncate the W_ETL_RUN_ANAL table for improved performance.

To run the Table Analyze Utility

- At the command prompt, enter the following:
  - Windows

    ```
    %MY_JAVA_HOME%\java -cp %CLASSPATH% AnalyzeTables -d PropertiesFile -p
    PasswordFile table1 table2 …
    ```

  - Unix

    ```
    $MY_JAVA_HOME/java -cp $CLASSPATH AnalyzeTables -d PropertiesFile -p PasswordFile
    table1 table2 …
    ```

  where the PropertiesFile is the name and path of the properties file. The PasswordFile is the name and path of the encrypted password file.

About Creating Command Tasks

Whenever the Siebel Customer-Centric Enterprise Warehouse session loads data successfully, it runs post (or pre) session command task, by calling a reusable standalone command task. This command task is called ANALYZE.

You can gather statistics for any of the target tables that are loaded by your sessions, by creating a similar stand alone command task by calling the reusable ANALYZE command task. For example, you could create a post session command task for any of your sessions by reusing the ANALYZE task and adding the target table name to need to be analyze.

Use your post session command task if you have to create a new database.properties file for any of the following reasons:

- Your target table resides on a different schema with different connection information.
- You decide to gather (or not to gather) analyzed logs for a specific table.
- You decide to use a different syntax for a table and you have a new copy of `customsql.xml` file.

You must have a different concurrency level set for the each specific session.

6 Configuring the Siebel Business Analytics Repository for Siebel Customer-Centric Enterprise Warehouse

This chapter describes how to configure the Siebel Business Analytics repository for the Siebel Customer-Centric Enterprise Warehouse.

It contains the following topics:

Configuring the Siebel Business Analytics Repository Connections

You need to configure the Siebel Business Analytics repository variables and connection pools to connect to your database.

Siebel Business Analytics repository uses three connection pools to the Physical layer:

- **Siebel Data Warehouse Connection Pool.** The Siebel Data Warehouse Connection Pool is the main connection pool in the Siebel Business Analytics 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.
- **Siebel Data Warehouse DBAuth Connection Pool.** The Siebel Data Warehouse DBAuth Connection Pool is used if database authentication is required.

- **Siebel Data Warehouse Repository Initblocks Connection Pool.** You need to configure the Siebel Data Warehouse Repository Initblocks Connection Pool to connect to the your physical data warehouse. The connection is used by the repository level initialization blocks. Repository level initialization blocks cannot be configured to use the dynamic data source name.

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

To configure the Siebel Business Analytics Data Warehouse database connection

1. Open the SiebelBusinessAnalytics.rpd in the `$SAHome\SiebelAnalytics\Repository` folder

   where `$SAHome` is the path of Siebel Business Analytics Server installation folder.

2. In the Physical pane, double-click the Siebel Business Analytics Data Warehouse.
3. In the Database list, click your database type.
4. Save the repository.
5. Click Yes to Check Global Consistency.
6. Click OK when the Warnings are displayed.

To configure the Siebel Business Analytics repository connection pools

1. Open the SiebelBusinessAnalytics.rpd in the `$SAHome\SiebelAnalytics\Repository` folder.
2. In the Physical pane
   a. Double-click the Siebel Data Warehouse Connection Pool.
   b. Type the database source name in the Data source name box.
   c. Type your database User ID in the User name box.
   d. Type your password in the Password box.
3. Repeat Step a through Step d for the Siebel Enterprise DBAuth Connection Pool and Siebel Data Warehouse Repository Initblocks Connection Pool connection pools.
4. Save the repository.
5. Click Yes to Check Global Consistency.
6. Click OK when the Warnings are displayed.

To configure the Siebel Business Analytics repository variables

1. Open the SiebelBusinessAnalytics.rpd in the `$SAHome\SiebelAnalytics\Repository` folder.

2. On the Manage menu, click Variables.
3. In the Variables Manager window, click Static.
4. Edit the `OLAP_DSN`, `OLAP_USER`, and `OLAPTBO` variables, and close the Variables Manager window.
5. On the Manage menu, click Security
6. In the Security Manager window, 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.

Customizing Time Dimensions

The time dimension in the Siebel Business Analytics repository for Siebel Customer-Centric Enterprise Warehouse is a standard or canonical time dimension that links to the important time role in each star schema.

Secondary dates are shown to the end users by a detailed presentation folder. The detailed presentation folder is typically called the Details folder.

About the Date Specific Metrics

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, if the Invoice fact table has three metrics called Invoice Amount, Fulfill Amount, and Paid Amount, then each of these metrics need to be reported by the corresponding date—Invoice Date, Fulfill Date, and Payment Date.

In Table 14, 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 14. Invoice Fact Table Example

Date	Invoice Amount	Fulfill Amount	Payment Amount
January	4000	5000	4500

To implement data specific metrics

1. Open the SiebelBusinessAnalytics.rpd with the Admin Tool.
2. Right-click on Siebel Business Analytics Data Warehouse in the Physical layer, and create a new physical alias for the fact table.
3. Create Joins for the physical alias which are the 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 Siebel Customer-Centric Enterprise Warehouse

Siebel Business Analytics repository provides a framework to add Period Ago metrics. The repository for Siebel Customer-Centric Enterprise Warehouse is preconfigured with premapped period ago metrics, however you can map other metrics by using the following procedure.

To set up additional time series metrics

1. Open the SiebelBusinessAnalytics.rpd with the Admin Tool.
2. Right-click on Siebel Business Analytics Data 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 SiebelBusinessAnalytics.rpd file.

Setting Up Additional Dimension Tables for Siebel Customer-Centric Enterprise Warehouse

Siebel Customer-Centric Enterprise Warehouse is preconfigured to map dimension tables required for analysis. The physical layer in the Siebel Business Analytics 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 (for example, SAP R/3) you are using.

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

2. Open the SiebelBusinessAnalytics.rpd with the Admin Tool.
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 SiebelBusinessAnalytics.rpd file.

Merging Siebel Business Analytics Repositories

You can merge the Siebel Business Analytics repositories for Siebel Relationship Management Warehouse and for Siebel Customer-Centric Enterprise Warehouse into one Siebel Business Analytics repository.

The merged repository can have conflicts for the following variables:

- CURRENT_MONTH
- CURRENT_QUARTER
- CURRENT_WEEK
- CURRENT_YEAR
- OLAPTBO
- OLAP_DSN
- OLAP_USER

You must decide which of these common variables to use. You need to rename or delete the variables you no longer require.

To merge Siebel Business Analytics repositories

1. Open the admin tool and create a new repository.
2. Save the repository with no content.

   For example, name the new repository file Original.rpd.
3. On the File menu, click Merge, and open the Original.rpd.

   The Merge repositories dialog box is displayed.
4. In the Modified repository, click Select, and open the SiebelAnalytics.rpd (Siebel Relationship Management Warehouse).
5. Click Merge.
6. In the Modified repository, click Select, and open the SiebelAnalytics.rpd (Siebel Customer-Centric Enterprise Warehouse).
7. Resolve any variable conflicts, and click Merge.
8. Save and close your merged repository file.

About Siebel Business Analytics Repository Variables

The Siebel Business Analytics repository is preconfigured with variables that are used for both reporting and internal usage.

Table 15 lists the Siebel Business Analytics repository date variables and their descriptions.

Table 15. Siebel Business Analytics Repository Date Variables

Variable Name	Description
BUILD	Holds the internal build number information for the Siebel Business Analytics Repository.
CAL_MONTH_YEAR_AGO	Returns the value of Previous Year Month in the YYYY/MM format.
CURRENT_DAY	Returns the value of Current Date in the MM/DD/YYYY format
CURRENT_FSCL_MONTH	Returns the value of Current Fiscal Month in the YYYY/MM format.
CURRENT_FSCL_QUARTER	Returns the value of Current Fiscal 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 Fiscal Quarter in YYYY Q n format.

Table 15. Siebel Business Analytics Repository Date Variables

Variable Name	Description
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 Fiscal Quarter in YYYY Q n format.
PREVIOUS_FSCL_MONTH	Returns the value of Previous Fiscal Month in the YYYY/MM format.
PREVIOUS_FSCL_QUARTER	Returns the value of Previous Fiscal 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 Fiscal 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.
NEXT_FSCL_MONTH	Returns the value of Next Fiscal Month in the YYYY / MM format.
NEXT_FSCL_QUARTER	Returns the value of Next Fiscal Quarter in the YYYY Q n.
NEXT_FSCL_WEEK	Returns the value of Next Fiscal Week in the YYYY Weeknn format.
NEXT_FSCL_YEAR	Returns the value of Next Fiscal Year in the FYYYYY format.
NEXT_MONTH	Returns the value of Next Month in the YYYY / MM format.
NEXT_QUARTER	Returns the value of Next Fiscal 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.
YEAR_AGO_DAY	Returns the value of year ago date in the mm/dd/yyyy format.
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.
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.

Table 15. Siebel Business Analytics Repository Date Variables

Variable Name	Description
IS_CME_ORDER_NUM	Set to 1 if the Order is a CME Order. This variable should not be changed.
CURRENT_BALANCE_DK_AP	Returns the value of the last date key for the available Accounts Payable balance. It is used in Accounts Payable Account Balance Computation.
CURRENT_BALANCE_DK_AR	Returns the value of the last date key for the available Accounts Receivables balance. It is used in Accounts Receivable Account Balance Computation.
CURRENT_BALANCE_DK_GL	Returns the value of the last date key for the available General Ledger balance. It is used in General Ledger Account Balance Computation.

Web Catalog Configuration Variables

The Web Catalog configuration variables are used to control the number of years displayed in the global filters for the year and fiscal year variables.

Table 16 lists the Web Catalog configuration variables and their descriptions in the Siebel Business Analytics repository.

Table 16. Web Catalog Configuration Variables

Variable Name	Description
FILTER_CAL_FROM_YEAR	You need to set this variable to the earliest year for the Year global filter.
FILTER_CAL_TO_YEAR	You need to set this variable to the earliest year for the Year global filter.
FILTER_FSCL_FROM_YEAR	You need to set this variable to the earliest year for the Year global filter.
FILTER_FSCL_TO_YEAR	You need to set this variable to the earliest year for the Year global filter.

About User Authentication

You need to configure the user authentication of the Siebel Customer-Centric Enterprise Warehouse Repository depending on your requirements. The Siebel Customer-Centric Enterprise Warehouse supports various authentication modes, for example, Repository authentication, Database authentication, and LDAP.

For more information about configuring user authentication, see *Siebel Analytics Server Administration Guide*.

About the Security or Visibility Configuration

The Siebel Customer-Centric Enterprise 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 *Siebel Analytics Server Administration Guide*.

Table 17 lists the groups in the Siebel Customer-Centric Enterprise Warehouse repository.

Table 17. 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 Siebel Payables Analytics.
AP Manager	This user group is able to view high-level application content for Siebel Payables Analytics.
AR Analyst	This user group is able to view application content for Siebel Receivables Analytics.
AR Manager	This user group is able to view high-level application content for Siebel Receivables Analytics.
CFO	This user group is able to view most of the Siebel Financial Analytics application content.
Contact Center and Agent Performance Analyst	This user group is able to view Siebel Enterprise Contact Center and Agent Performance application content.
Contact Center and Agent Performance User	This user group is able to view a subset of Siebel Enterprise Contact Center and Agent Performance application content.
Contact Center Sales Analyst	This user group is able to view Siebel Enterprise Contact Center Sales Analytics application content.
Contact Center Sales User	This user group is able to view a subset of Siebel Enterprise Contact Center Sales Analytics application content.
Controller	This user group is able to view application content for Siebel General Ledger Analytics and Siebel Profitability Analytics.

Table 17. Repository User Groups

Repository User Group	Description
Customer Service Analyst	This user group is able to view Customer Service for Siebel Enterprise Contact Center Analytics application content.
Customer Service User	This user group is able to view a subset of Customer Service for Siebel Enterprise Contact Center Analytics application content.
Enterprise Contact Center User	This user group is able to view Siebel Enterprise Contact Center Analytics application content.
Financial Analyst	This user group is able to view Siebel Financial Analytics application content.
Human Resources Analyst	This user group is able to view Siebel Enterprise Workforce Analytics application content.
Human Resources Vice President	This user group is able to view high-level application content for Siebel Enterprise Workforce Analytics application.
Inventory Analyst	This user group is able to view application content for Siebel Supply Chain Analytics application.
Inventory Manager	This user group is able to view high-level application content for Siebel Supply Chain Analytics application.
Primary Owner-Based Security	Used for securing owner-based data elements that come from the Siebel Customer-Centric Enterprise Warehouse transactional system.
Primary Position-Based Security	Used for securing position-based data elements that come from the Siebel Customer-Centric Enterprise Warehouse transactional system.
Purchasing Buyer	This user group is able to view Siebel Strategic Sourcing Analytics application content pertaining to purchasing.
Sales Executive Analytics	This user group is able to view high-level application content for Siebel Enterprise Sales Analytics application.
Sales Manager	This user group is able to view most of the high-level application content for Siebel Enterprise Sales Analytics application.
Sales Manager Analytics	This user group is able to view most of the high-level application content for Siebel Enterprise Sales Analytics application.
Sales Operations Analytics	This user group is able to view operational application content for Siebel Enterprise Sales Analytics application.
Sales Representative Analytics	This user group is able to view low-level application content for Siebel Enterprise Sales Analytics application.

Table 17. Repository User Groups

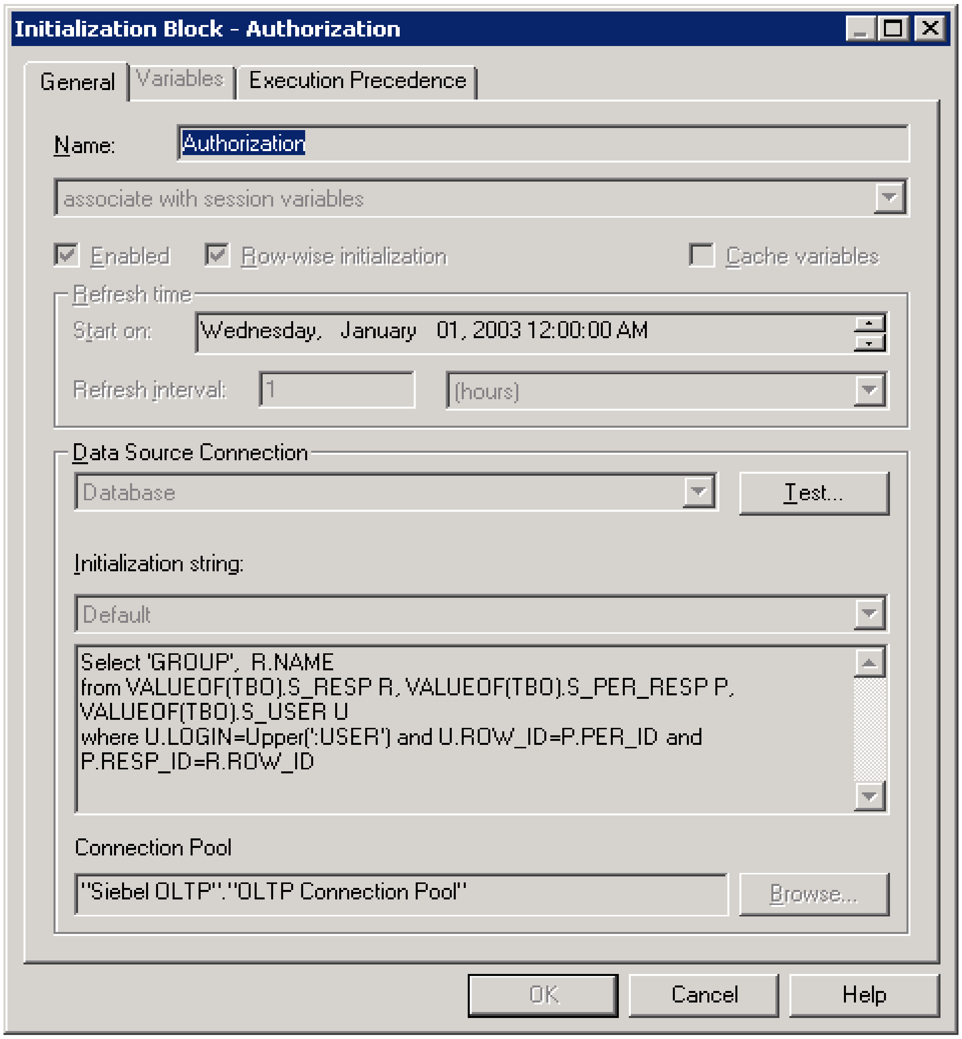
Repository User Group	Description
Sales Rev and Fulfill Analyst	This user group is able to view the content for Siebel Enterprise Sales Analytics Revenue and Fulfillment application.
Sales Rev and Fulfill Exec	This user group is able to view the high-level application content for Siebel Enterprise Sales Analytics Revenue and Fulfillment application.
Sales Rev and Fulfill Mgr	This user group is able to view most of the high-level application content for Siebel Enterprise Sales Analytics Revenue and Fulfillment application.
Sales Rev and Fulfill Rep	This user group is able to view low-level application content for Siebel Enterprise Sales Analytics Revenue and Fulfillment application.
Sales Revenue Analyst	This user group is able to view the content for Siebel Enterprise Sales Analytics Revenue application.
Sales Revenue Exec	This user group is able to view the high-level application content for Siebel Enterprise Sales Analytics Revenue application.
Sales Revenue Mgr	This user group is able to view most of the high-level application content for Siebel Enterprise Sales Analytics Revenue application.
Sales Revenue Rep	This user group is able to view low-level application content for Siebel Enterprise Sales Analytics Revenue application.
Service Delivery and Costs Analyst	This user group is able to view Service Delivery and Costs for Siebel Enterprise Contact Center Analytics application content.
Service Delivery and Costs User	This user group is able to view a subset of Service Delivery and Costs for Siebel Enterprise Contact Center Analytics application content.
Supplier Performance Analyst	This user group is able to view Siebel Strategic Sourcing Analytics application content pertaining to supplier performance.
Supplier Performance Manager	This user group is able to view high-level content for Siebel Strategic Sourcing Analytics application pertaining to supplier performance.
Supply Chain Executive	This user group is able to view Siebel Supply Chain Analytics and Siebel Strategic Sourcing Analytics application content.

About the Group Variable

The Group variable determines the membership of a user in the various security groups. If you are using Siebel Customer-Centric Enterprise Warehouse with Siebel Relationship Management Warehouse, you need to associate users to the Group variable as part of the repository configuration.

For more information about configuring the Group variable, see *Siebel Analytics Web Administration Guide*.

Figure 3 shows an example of an initialization block that associates a user to a Group membership.



Initialization Block - Authorization

General | Variables | Execution Precedence

Name: Authorization

associate with session variables

☑ Enabled ☑ Row-wise initialization ☐ Cache variables

Refresh time

Start on: Wednesday, January 01, 2003 12:00:00 AM

Refresh interval: 1 (hours)

Data Source Connection

Database

Initialization string:

Default

```
Select 'GROUP', R.NAME
from VALUEOF(TBO).S_RESP R, VALUEOF(TBO).S_PER_RESP P,
VALUEOF(TBO).S_USER U
where U.LOGIN=Upper(':USER') and U.ROW_ID=P.PER_ID and
P.RESP_ID=R.ROW_ID
```

Connection Pool

"Siebel OLTP"."OLTP Connection Pool"

Figure 3. Group Initialization Block Example

About the Period Ago Keys for Siebel Customer-Centric Enterprise 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. Siebel Customer-Centric Enterprise Warehouse is preconfigured with a set of fields in the W_DAY_D table. These fields are:

- MONTH_AGO_KEY
- QUARTER_AGO_KEY
- TRIMESTER_AGO_KEY
- WEEK_AGO_KEY
- YEAR_AGO_KEY

These fields are used in joins to Siebel Customer-Centric Enterprise Warehouse fact tables to achieve the period ago metrics. The surrogate keys in Siebel Customer-Centric Enterprise Warehouse fact tables uses are different to the surrogate keys that the Siebel Relationship Management Warehouse uses. The joins in Siebel Customer-Centric Enterprise Warehouse uses the Period Ago fields in the W_DAY_D table.

About Configuring Usage Tracking for Siebel Customer-Centric Enterprise Warehouse

Siebel Customer-Centric Enterprise Warehouse supports the accumulation of usage tracking statistics. The Siebel Business Analytics repository for Siebel Customer-Centric Enterprise 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 *Siebel Analytics Server Administration Guide*.

About the Incremental Deployment of the Siebel Business Analytics Repository

Siebel Customer-Centric Enterprise Warehouse consist of various families of applications, for example, Siebel Supply Chain Analytics, Siebel Enterprise Sales Analytics, Siebel Financial Analytics, and so on. You can purchase these applications at different times. You can customize functionality and incrementally add new application or applications.

This section describes the procedure for deploying multiple applications. You can repeat the procedure to add applications incrementally.

Figure 4 shows a single Siebel Customer-Centric Enterprise Warehouse application environment. A license key is used to extract the Siebel Business Analytics repository. You can modify the Siebel Business Analytics repository to suit your business needs.

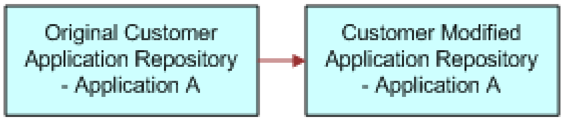

Original Customer Application Repository - Application A
Customer Modified Application Repository - Application A

Figure 4. Single Repository Environment

When you purchase another Siebel Customer-Centric Enterprise Warehouse application, you need to use the combined license key to extract both Siebel Business Analytics application repositories. Use the Siebel Analytics Administration merge utility to perform a three-way merge of the original repository, the modified repository, and the combined repository. For more information on merging repositories, see Merging Siebel Business Analytics Repositories on page 79.

The merged repository preserves your modifications from the original Siebel Business Analytics repository and appends the information with the new Siebel Business Analytics repository, as shown in Figure 5.

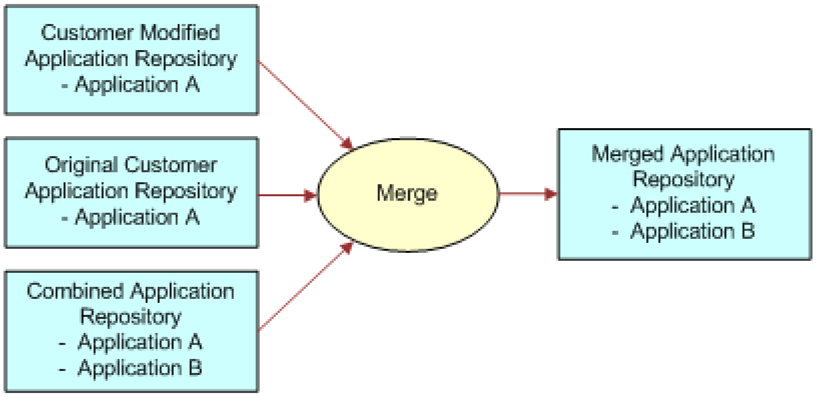

Customer Modified Application Repository - Application A
Original Customer Application Repository - Application A
Combined Application Repository
- Application A
- Application B
Merge
Merged Application Repository
- Application A
- Application B

Figure 5. Merging Repositories

You can repeat this merging procedure to add more Siebel Customer-Centric Enterprise Warehouse applications to the Siebel Business Analytics repository.

7 Deploying Multiple Siebel Customer-Centric Enterprise Warehouse Applications

This chapter provides instructions for deploying multiple applications for the Siebel Customer-Centric Enterprise Warehouse.

It contains the following topics:

About Building Multi-Application Workflows for the Siebel Business Analytics

The Siebel Customer-Centric Enterprise Warehouse is structured into different applications and a customer may buy one or more of these applications. By default, the workflows execute one application at a time. If you buy more than one application you can run all of these applications together as one workflow. Before creating a multi-application workflow, you need to import all the individual application XML files into the same repository. For more information on importing XML files into a repository, see Importing Application Workflows into the Repository on page 45.

For more information on how to create a master workflow that allows you to run all your applications together, see Process of Building Multi-Application Workflows for the Siebel Business Analytics on page 89.

Process of Building Multi-Application Workflows for the Siebel Business Analytics

This section describes the process of building a multi-application workflow for the Siebel Business Analytics.

To build a multi-application workflow for the Siebel Business Analytics, perform the following tasks:

- Creating the Multi-Application Workflow Shell on page 90
- Creating the Post-Load Processing Initial Workflow on page 91
- Creating the Post-Load Processing Incremental Workflow on page 91
- Editing the Nonreusable Worklets on page 92

Creating the Multi-Application Workflow Shell

This task is a step in the Process of Building Multi-Application Workflows for the Siebel Business Analytics on page 89.

This task creates a multi-application workflow shell.

To create a multi-application workflow shell

1. Open PowerCenter Workflow Manager and connect to your repository.
2. Open your source-specific folder.
3. On the Tools menu, click Workflow Designer.
4. On the Workflows menu, select Create.

   Create the multi-application workflow and add the following nonreusable worklets:

   - Prepare
   - Extract_Facts
   - Extract_Dimensions
   - Load_Dimensions
   - Load_Facts
5. Link the nonreusable worklets together.
6. Double-click on all the workflow links and add $Status = SUCCEEDED OR $Status = DISABLED into the link.
7. Include the reusable worklet S_EXECUTION_FINISHED at the end of the workflow.
8. Edit the workflow and select the Suspend on Error check box.
9. Double-click on all your worklets and select the Fail Parent if this task fails check box.

   The multi-application workflow shell is now created. You need to insert the appropriate application specific reusable worklets into the nonreusable worklets. For more information on editing the reusable worklets, see Editing the Nonreusable Worklets on page 92.

Creating the Post-Load Processing Initial Workflow

This task is a step in the Process of Building Multi-Application Workflows for the Siebel Business Analytics on page 89.

This task creates the post-load processing initial workflow.

To create the Post-Load Processing Initial workflow

1. Open PowerCenter Workflow Manager and connect to your repository.
2. Open the Configuration for Post Load Processing folder.
3. On the Tools menu, select Workflow Designer.
4. On the Workflow menu, click Create, and create a workflow called PLP_INIT.
5. Include the appropriate post-load processing worklets based on your application combinations.

   The following table lists the order of the reusable worklets for the Post-Load Processing Initial Workflow.

Worklet Order	Reusable Worklet Details
1	PLP_EVENT_WAIT
2	PLP_PREPARE
3, 4, and so on.	For more information on which reusable worklets to use for the Post-Load Processing Initial Workflow, see Post-Load Processing Initial Worklet on page 109.

6. Edit the PLP_INIT workflow and select the Suspend on Error check box.
7. Double-click on all the workflow links and add $Status = SUCCEEDED OR $Status = DISABLED into the link.
8. Double-click on all your worklets and select the Fail Parent if this task fails check box.

Creating the Post-Load Processing Incremental Workflow

This task is a step in the Process of Building Multi-Application Workflows for the Siebel Business Analytics on page 89.

This task creates the post-load processing incremental workflow.

To create the Post-Load Processing Incremental workflow

1. Open PowerCenter Workflow Manager and connect to your repository.
2. Open the Configuration for Post Load Processing folder.

3 On the Tools menu, select Workflow Designer.

4 On the Workflows menu, select Create, and create a workflow called PLP_INCR.

5 Include the appropriate post-load processing worklets based on your application combinations.

The following table lists the order of the reusable worklets for the Post-Load Processing Incremental Workflow.

Worklet Order	Reusable Worklet Details
1	PLP_EVENT_WAIT
2	PLP_PREPARE
3, 4, and so on.	For more information on which reusable worklets to use for the Post-Load Processing Incremental Workflow, see Post-Load Processing Incremental Worklet on page 110.

6 Edit the PLP_INCR workflow and select the Suspend on Error check box.

7 Double-click on all the workflow links and add $Status = SUCCEEDED OR $Status = DISABLED into the link.

8 Double-click on all your worklets and select the Fail Parent if this task fails check box.

Editing the Nonreusable Worklets

This task is a step in the Process of Building Multi-Application Workflows for the Siebel Business Analytics on page 89.

This task edits the nonreusable worklets in your multi-application workflow and replaces them with reusable worklets.

To edit the nonreusable worklets

1 Right-click on each of the nonreusable worklets in your multi-application workflow, and select Open Worklet.

2 On the Tasks menu, click Insert Worklet.

You can edit the nonreusable shell worklets and add the appropriate reusable worklets, as shown in the following table.

Shell Worklet	Source System	Reusable Worklets Details
Prepare	All	S_PREPARE
Extract_Facts	SAP	SAP Facts Extract Worklet on page 96
	Oracle 11i	Oracle 11i Facts Extract Worklet on page 103
	Universal	Universal Fact Extract Worklet on page 107

Shell Worklet	Source System	Reusable Worklets Details
Extract_Dimensions	SAP	SAP Dimensions Extract Worklet on page 94
	Oracle 11i	Oracle 11i Dimensions Extract Worklet on page 101
	Universal	Universal Dimensions Extract Worklet on page 105
Load_Dimensions	SAP	SAP Dimensions Load Worklet on page 95
	Oracle 11i	Oracle 11i Dimensions Load Worklet on page 102
	Universal	Universal Dimensions Load Worklet on page 106
Load_Facts	SAP	SAP Facts Load Worklet on page 98
	Oracle 11i	Oracle 11i Facts Load Worklet on page 104
	Universal	Universal Fact Load Worklet on page 108

3 Edit each of the reusable worklets, and select the Fail parent if this task fails check box.

NOTE: All the extract worklets run in parallel. The dependent worklets run after the worklets that they depend on. The group worklets that are in the same columns in Siebel Business Analytics Workflows and Dependent Worklets on page 93 can run in parallel.

Siebel Business Analytics Workflows and Dependent Worklets

The following sections list the Siebel Business Analytics workflows, their worklets, and their dependent worklets:

- SAP Dimensions Extract Worklet on page 94
- SAP Dimensions Load Worklet on page 95
- SAP Facts Extract Worklet on page 96
- SAP Facts Load Worklet on page 98
- Oracle 11i Dimensions Extract Worklet on page 101
- Oracle 11i Dimensions Load Worklet on page 102
- Oracle 11i Facts Extract Worklet on page 103
- Oracle 11i Facts Load Worklet on page 104
- Universal Dimensions Extract Worklet on page 105
- Universal Dimensions Load Worklet on page 106
- Universal Fact Extract Worklet on page 107
- Universal Fact Load Worklet on page 108
- Post-Load Processing Initial Worklet on page 109

- Post-Load Processing Incremental Worklet on page 110

SAP Dimensions Extract Worklet

Figure 6 illustrates the worklets and their dependant worklets for the SAP Dimensions Extract worklet.

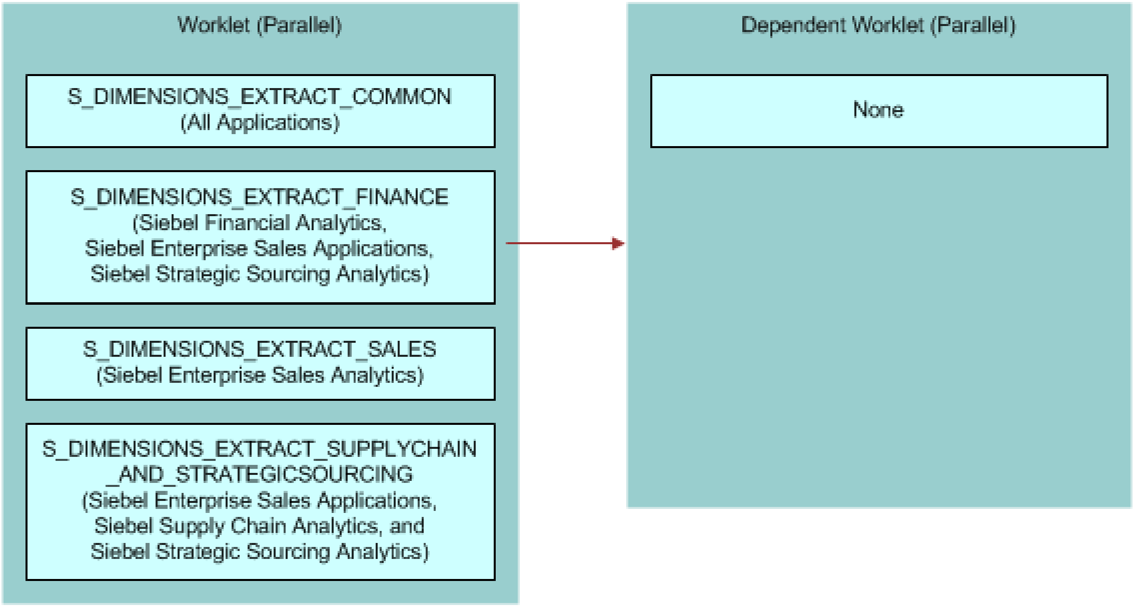

Worklet (Parallel)
S_DIMENSIONS_EXTRACT_COMMON (All Applications)
S_DIMENSIONS_EXTRACT_FINANCE (Siebel Financial Analytics, Siebel Enterprise Sales Applications, Siebel Strategic Sourcing Analytics)
S_DIMENSIONS_EXTRACT_SALES (Siebel Enterprise Sales Analytics)
S_DIMENSIONS_EXTRACT_SUPPLYCHAIN _AND_STRATEGICSOURCING (Siebel Enterprise Sales Applications, Siebel Supply Chain Analytics, and Siebel Strategic Sourcing Analytics)
Dependent Worklet (Parallel)
None

Figure 6. SAP Dimensions Extract Worklet

SAP Dimensions Load Worklet

Figure 7 illustrates the worklets and their dependant worklets for the SAP Dimensions Load worklet.

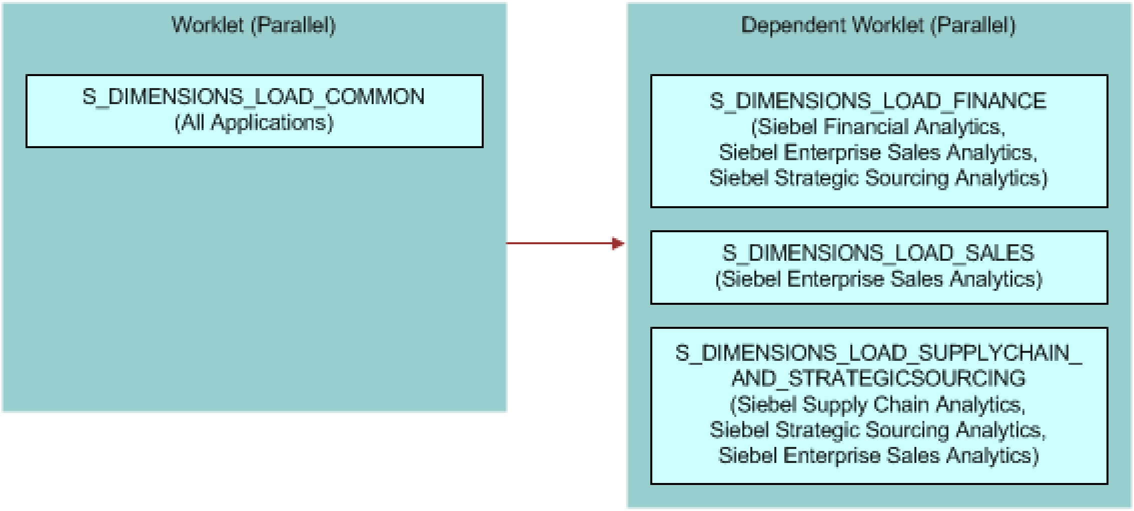

Worklet (Parallel)
S_DIMENSIONS_LOAD_COMMON (All Applications)
Dependent Worklet (Parallel)
S_DIMENSIONS_LOAD_FINANCE (Siebel Financial Analytics, Siebel Enterprise Sales Analytics, Siebel Strategic Sourcing Analytics)
S_DIMENSIONS_LOAD_SALES (Siebel Enterprise Sales Analytics)
S_DIMENSIONS_LOAD_SUPPLYCHAIN_AND_STRATEGICSOURCING (Siebel Supply Chain Analytics, Siebel Strategic Sourcing Analytics, Siebel Enterprise Sales Analytics)

Figure 7. SAP Dimensions Load Worklet

SAP Facts Extract Worklet

Figure 8 illustrates the design you could use if you are combining Siebel Enterprise Sales Analytics and Siebel Financial Analytics for the SAP Facts Extract worklet. You can also add other applications, for example, Siebel Supply Chain Analytics, Siebel Strategic Sourcing Analytics, and so on.

NOTE: You need to choose between the `S_FACTS_EXTRACT_FINANCE_DETAIL_SALES_COMBINATION_ONLY` worklet and the `S_FACTS_EXTRACT_FINANCE_HEADER_SALES_COMBINATION_ONLY` worklet. Your decision depends on how your application extracts sales information. For more information on extracting data posted at the header or detail level for SAP R/3, see Process of Configuring Siebel Financial Analytics for SAP R/3 on page 329.

Figure 9 illustrates the design you could use if you are not combining Siebel Enterprise Sales Analytics with Siebel Financial Analytics for the SAP Facts Extract worklet. For example, use this design if you are combining Siebel Enterprise Sales Analytics, Siebel Supply Chain Analytics, and Siebel Strategic Sourcing Analytics.

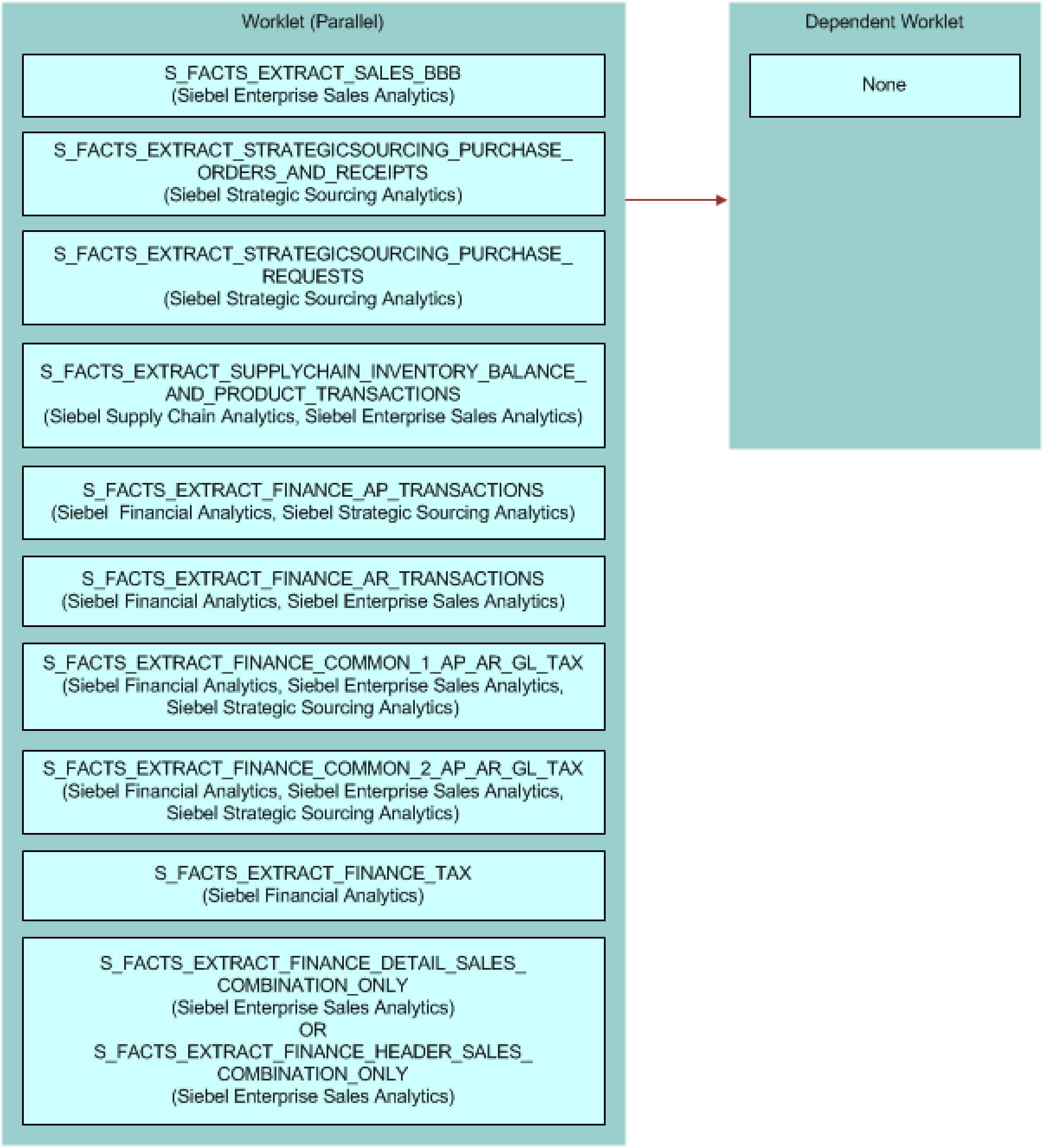

Worklet (Parallel)
S_FACTS_EXTRACT_SALES_BBB
(Siebel Enterprise Sales Analytics)
S_FACTS_EXTRACT_STRATEGICSOURCING_PURCHASE_ORDERS_AND_RECEIPTS
(Siebel Strategic Sourcing Analytics)
S_FACTS_EXTRACT_STRATEGICSOURCING_PURCHASE_REQUESTS
(Siebel Strategic Sourcing Analytics)
S_FACTS_EXTRACT_SUPPLYCHAIN_INVENTORY_BALANCE_AND_PRODUCT_TRANSACTIONS
(Siebel Supply Chain Analytics, Siebel Enterprise Sales Analytics)
S_FACTS_EXTRACT_FINANCE_AP_TRANSACTIONS
(Siebel Financial Analytics, Siebel Strategic Sourcing Analytics)
S_FACTS_EXTRACT_FINANCE_AR_TRANSACTIONS
(Siebel Financial Analytics, Siebel Enterprise Sales Analytics)
S_FACTS_EXTRACT_FINANCE_COMMON_1_AP_AR_GL_TAX
(Siebel Financial Analytics, Siebel Enterprise Sales Analytics, Siebel Strategic Sourcing Analytics)
S_FACTS_EXTRACT_FINANCE_COMMON_2_AP_AR_GL_TAX
(Siebel Financial Analytics, Siebel Enterprise Sales Analytics, Siebel Strategic Sourcing Analytics)
S_FACTS_EXTRACT_FINANCE_TAX
(Siebel Financial Analytics)
S_FACTS_EXTRACT_FINANCE_DETAIL_SALES_COMBINATION_ONLY
(Siebel Enterprise Sales Analytics)
OR
S_FACTS_EXTRACT_FINANCE_HEADER_SALES_COMBINATION_ONLY
(Siebel Enterprise Sales Analytics)
Dependent Worklet
None

Figure 8. SAP Facts Extract Worklet

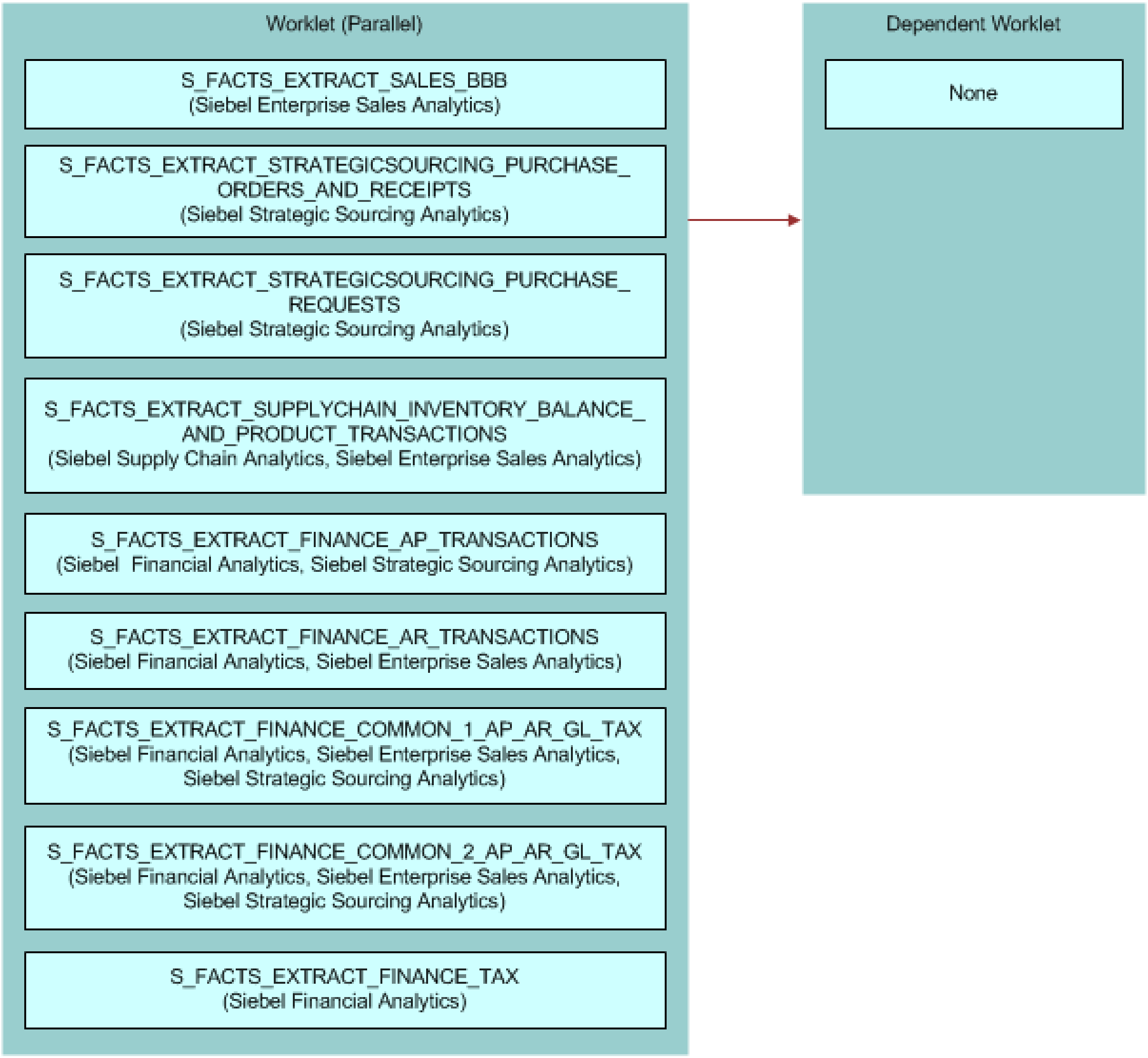

Worklet (Parallel)
S_FACTS_EXTRACT_SALES_BBB
(Siebel Enterprise Sales Analytics)
S_FACTS_EXTRACT_STRATEGICSOURCING_PURCHASE_ORDERS_AND_RECEIPTS
(Siebel Strategic Sourcing Analytics)
S_FACTS_EXTRACT_STRATEGICSOURCING_PURCHASE_REQUESTS
(Siebel Strategic Sourcing Analytics)
S_FACTS_EXTRACT_SUPPLYCHAIN_INVENTORY_BALANCE_AND_PRODUCT_TRANSACTIONS
(Siebel Supply Chain Analytics, Siebel Enterprise Sales Analytics)
S_FACTS_EXTRACT_FINANCE_AP_TRANSACTIONS
(Siebel Financial Analytics, Siebel Strategic Sourcing Analytics)
S_FACTS_EXTRACT_FINANCE_AR_TRANSACTIONS
(Siebel Financial Analytics, Siebel Enterprise Sales Analytics)
S_FACTS_EXTRACT_FINANCE_COMMON_1_AP_AR_GL_TAX
(Siebel Financial Analytics, Siebel Enterprise Sales Analytics, Siebel Strategic Sourcing Analytics)
S_FACTS_EXTRACT_FINANCE_COMMON_2_AP_AR_GL_TAX
(Siebel Financial Analytics, Siebel Enterprise Sales Analytics, Siebel Strategic Sourcing Analytics)
S_FACTS_EXTRACT_FINANCE_TAX
(Siebel Financial Analytics)
Dependent Worklet
None

Figure 9. SAP Facts Extract Worklet

SAP Facts Load Worklet

Figure 10 illustrates the design you could use if you are combining Siebel Enterprise Sales Analytics and Siebel Financial Analytics for the SAP Facts Load worklet. You can also add other applications, for example, Siebel Supply Chain Analytics, Siebel Strategic Sourcing Analytics, and so on.

NOTE: You need to choose between the S_FACTS_LOAD_FINANCE_DETAIL_SALES_COMBINATION_ONLY worklet and the S_FACTS_LOAD_FINANCE_HEADER_SALES_COMBINATION_ONLY worklet. Your decision depends on how your application extracts sales information. For more information on extracting data posted at the detail or header level for SAP R/3, see Process of Configuring Siebel Financial Analytics for SAP R/3 on page 329.

Figure 11 illustrates the design you could use if you are not combining Siebel Enterprise Sales Analytics with Siebel Financial Analytics for the SAP Facts Load worklet. For example, use this design if you are combining Siebel Enterprise Sales Analytics, Siebel Supply Chain Analytics, and Siebel Strategic Sourcing Analytics.

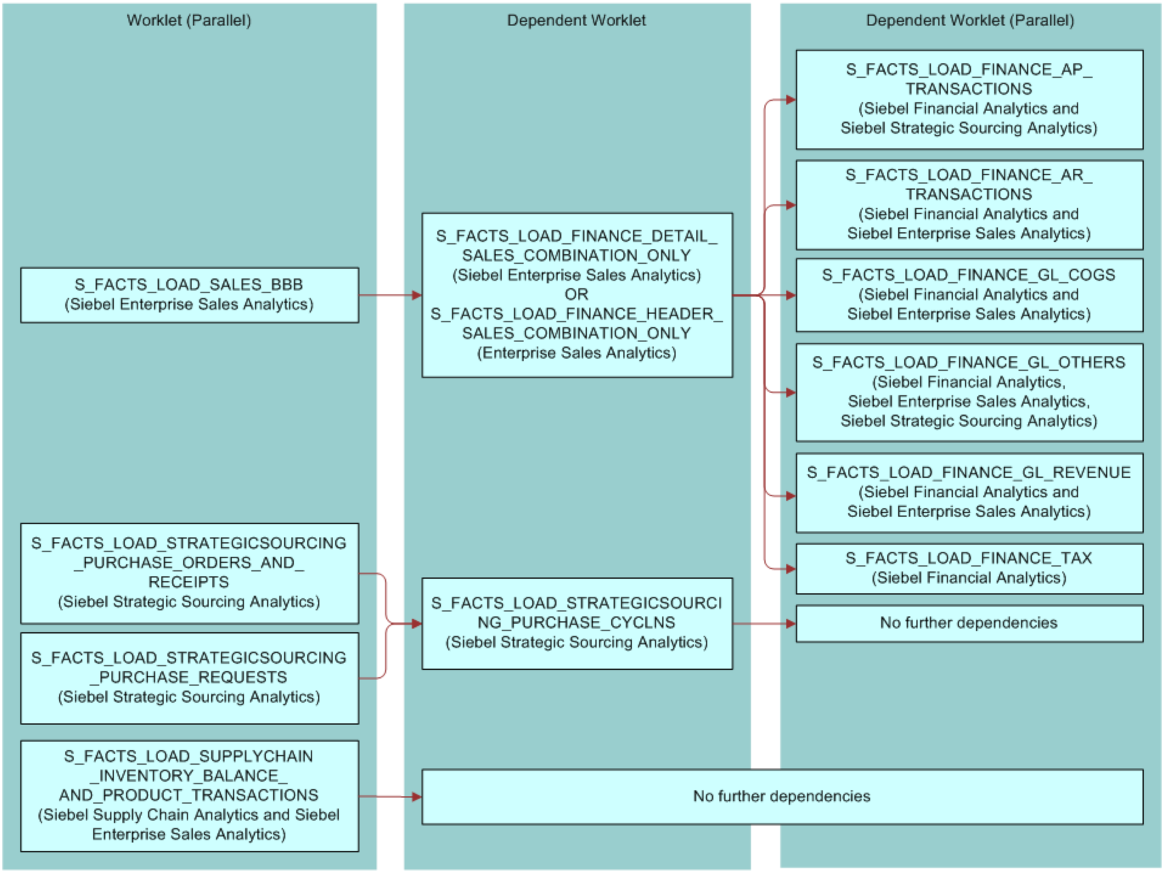

Worklet (Parallel)
Dependent Worklet
Dependent Worklet (Parallel)
S_FACTS_LOAD_SALES_BBB (Siebel Enterprise Sales Analytics)
S_FACTS_LOAD_FINANCE_DETAIL_ SALES_COMBINATION_ONLY (Siebel Enterprise Sales Analytics) OR S_FACTS_LOAD_FINANCE_HEADER_ SALES_COMBINATION_ONLY (Enterprise Sales Analytics)
S_FACTS_LOAD_FINANCE_AP_ TRANSACTIONS (Siebel Financial Analytics and Siebel Strategic Sourcing Analytics)
S_FACTS_LOAD_FINANCE_AR_ TRANSACTIONS (Siebel Financial Analytics and Siebel Enterprise Sales Analytics)
S_FACTS_LOAD_FINANCE_GL_COGS (Siebel Financial Analytics and Siebel Enterprise Sales Analytics)
S_FACTS_LOAD_FINANCE_GL_OTHERS (Siebel Financial Analytics, Siebel Enterprise Sales Analytics, Siebel Strategic Sourcing Analytics)
S_FACTS_LOAD_FINANCE_GL_REVENUE (Siebel Financial Analytics and Siebel Enterprise Sales Analytics)
S_FACTS_LOAD_FINANCE_TAX (Siebel Financial Analytics)
S_FACTS_LOAD_STRATEGICSOURCING _PURCHASE_ORDERS_AND_ RECEIPTS (Siebel Strategic Sourcing Analytics)
S_FACTS_LOAD_STRATEGICSOURCING _PURCHASE_REQUESTS (Siebel Strategic Sourcing Analytics)
S_FACTS_LOAD_STRATEGICSOURCI NG_PURCHASE_CYCLNS (Siebel Strategic Sourcing Analytics)
No further dependencies
S_FACTS_LOAD_SUPPLYCHAIN _INVENTORY_BALANCE_ AND_PRODUCT_TRANSACTIONS (Siebel Supply Chain Analytics and Siebel Enterprise Sales Analytics)
No further dependencies

Figure 10. SAP Facts Load Worklet

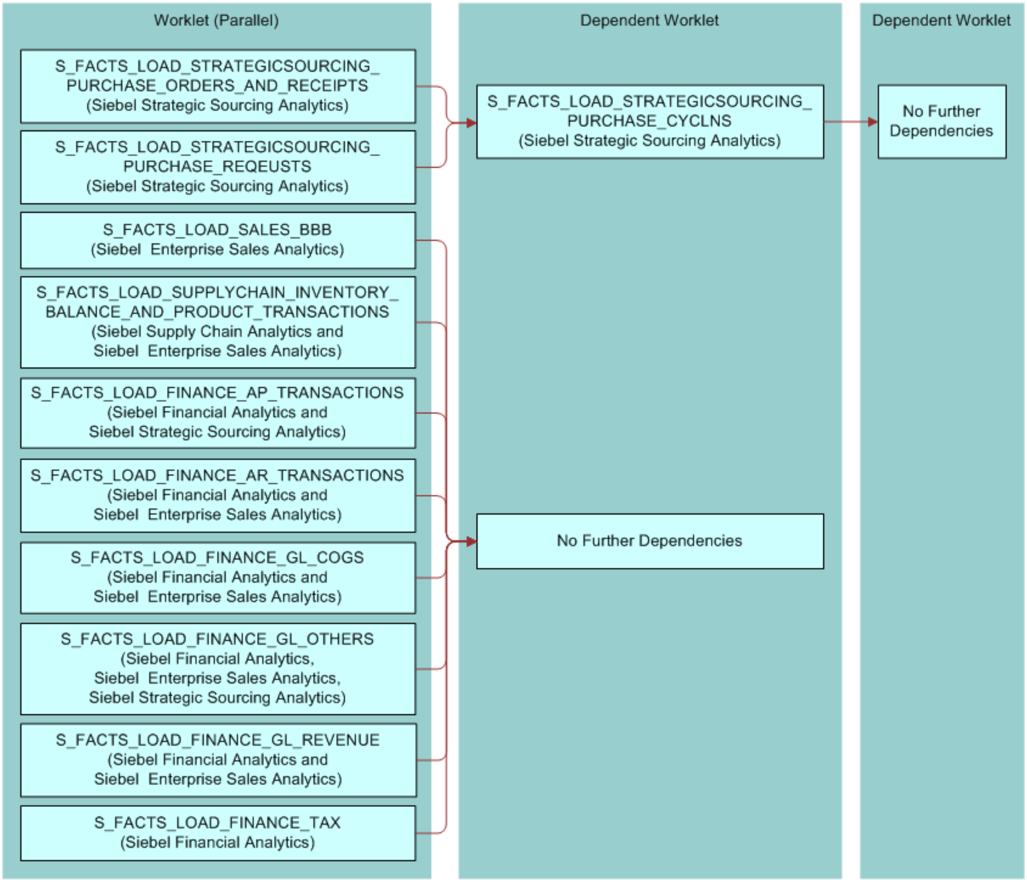

Worklet (Parallel)
S_FACTS_LOAD_STRATEGICSOURCING_ PURCHASE_ORDERS_AND_RECEIPTS (Siebel Strategic Sourcing Analytics)
S_FACTS_LOAD_STRATEGICSOURCING_ PURCHASE_REQEUSTS (Siebel Strategic Sourcing Analytics)
S_FACTS_LOAD_SALES_BBB (Siebel Enterprise Sales Analytics)
S_FACTS_LOAD_SUPPLYCHAIN_INVENTORY_ BALANCE_AND_PRODUCT_TRANSACTIONS (Siebel Supply Chain Analytics and Siebel Enterprise Sales Analytics)
S_FACTS_LOAD_FINANCE_AP_TRANSACTIONS (Siebel Financial Analytics and Siebel Strategic Sourcing Analytics)
S_FACTS_LOAD_FINANCE_AR_TRANSACTIONS (Siebel Financial Analytics and Siebel Enterprise Sales Analytics)
S_FACTS_LOAD_FINANCE_GL_COGS (Siebel Financial Analytics and Siebel Enterprise Sales Analytics)
S_FACTS_LOAD_FINANCE_GL_OTHERS (Siebel Financial Analytics, Siebel Enterprise Sales Analytics, Siebel Strategic Sourcing Analytics)
S_FACTS_LOAD_FINANCE_GL_REVENUE (Siebel Financial Analytics and Siebel Enterprise Sales Analytics)
S_FACTS_LOAD_FINANCE_TAX (Siebel Financial Analytics)
Dependent Worklet
S_FACTS_LOAD_STRATEGICSOURCING_ PURCHASE_CYCLNS (Siebel Strategic Sourcing Analytics)
No Further Dependencies
Dependent Worklet
No Further Dependencies

Figure 11. SAP Facts Load Worklet

Oracle 11i Dimensions Extract Worklet

Figure 12 illustrates the worklets and their dependant worklets for the Oracle 11i Dimensions Extract worklet.

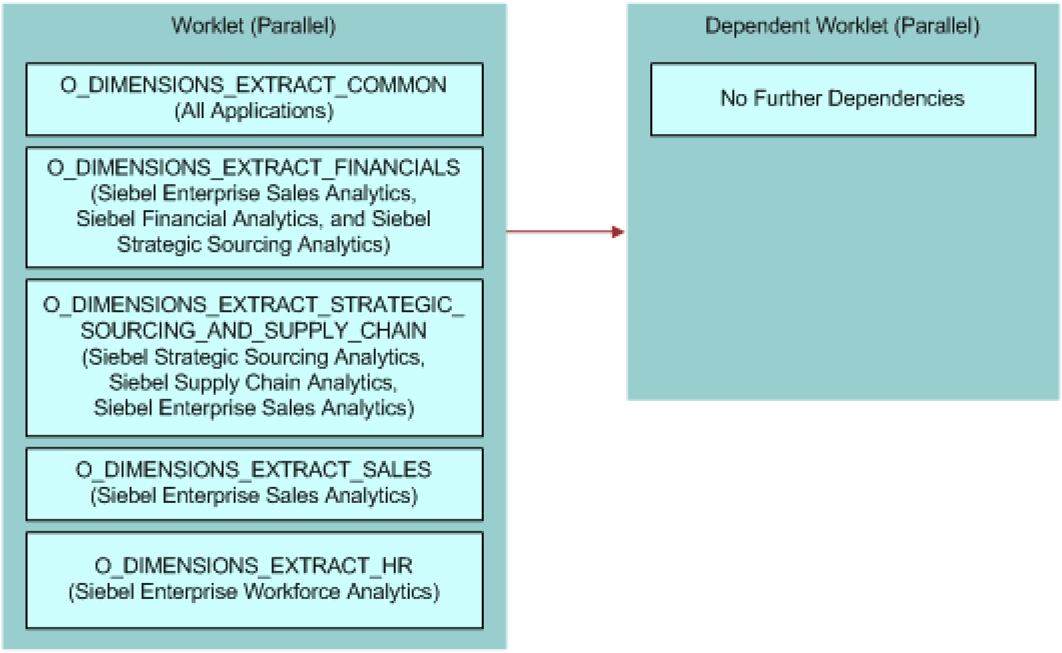

Worklet (Parallel)
O_DIMENSIONS_EXTRACT_COMMON
(All Applications)
O_DIMENSIONS_EXTRACT_FINANCIALS
(Siebel Enterprise Sales Analytics, Siebel Financial Analytics, and Siebel Strategic Sourcing Analytics)
O_DIMENSIONS_EXTRACT_STRATEGIC_SOURCING_AND_SUPPLY_CHAIN
(Siebel Strategic Sourcing Analytics, Siebel Supply Chain Analytics, Siebel Enterprise Sales Analytics)
O_DIMENSIONS_EXTRACT_SALES
(Siebel Enterprise Sales Analytics)
O_DIMENSIONS_EXTRACT_HR
(Siebel Enterprise Workforce Analytics)
Dependent Worklet (Parallel)
No Further Dependencies

Figure 12. Oracle 11i Dimensions Extract Worklet

Oracle 11i Dimensions Load Worklet

Figure 13 illustrates the worklets and their dependant worklets for the Oracle 11i Dimensions Load worklet.

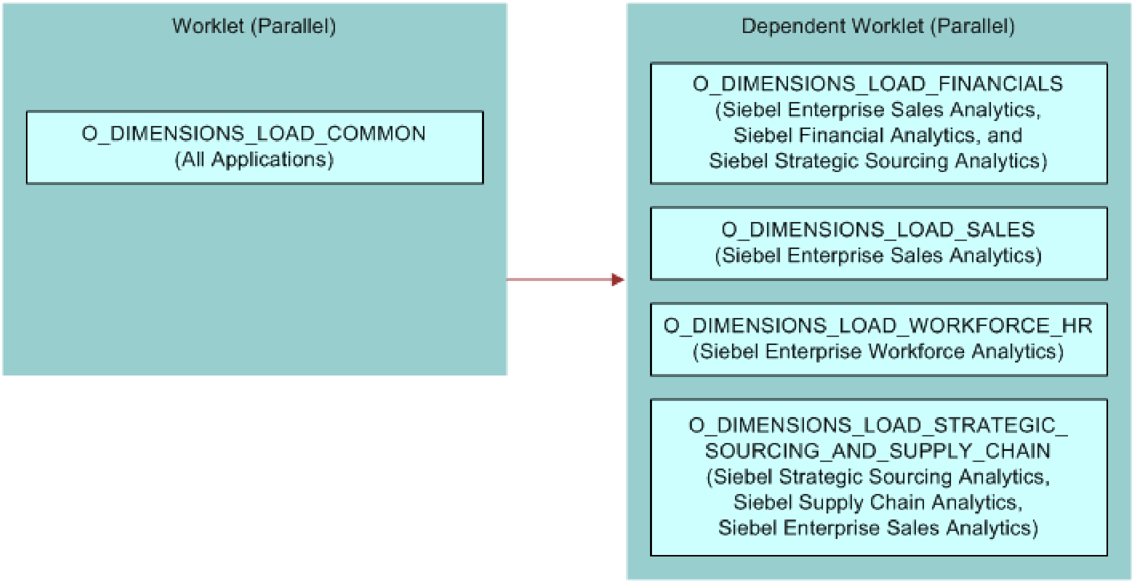

Worklet (Parallel)
O_DIMENSIONS_LOAD_COMMON (All Applications)
Dependent Worklet (Parallel)
O_DIMENSIONS_LOAD_FINANCIALS (Siebel Enterprise Sales Analytics, Siebel Financial Analytics, and Siebel Strategic Sourcing Analytics)
O_DIMENSIONS_LOAD_SALES (Siebel Enterprise Sales Analytics)
O_DIMENSIONS_LOAD_WORKFORCE_HR (Siebel Enterprise Workforce Analytics)
O_DIMENSIONS_LOAD_STRATEGIC_ SOURCING_AND_SUPPLY_CHAIN (Siebel Strategic Sourcing Analytics, Siebel Supply Chain Analytics, Siebel Enterprise Sales Analytics)

Figure 13. Oracle 11i Dimensions Load Worklet

Oracle 11i Facts Extract Worklet

Figure 14 illustrates the worklets and their dependant worklets for the Oracle 11i Facts Extract worklet.

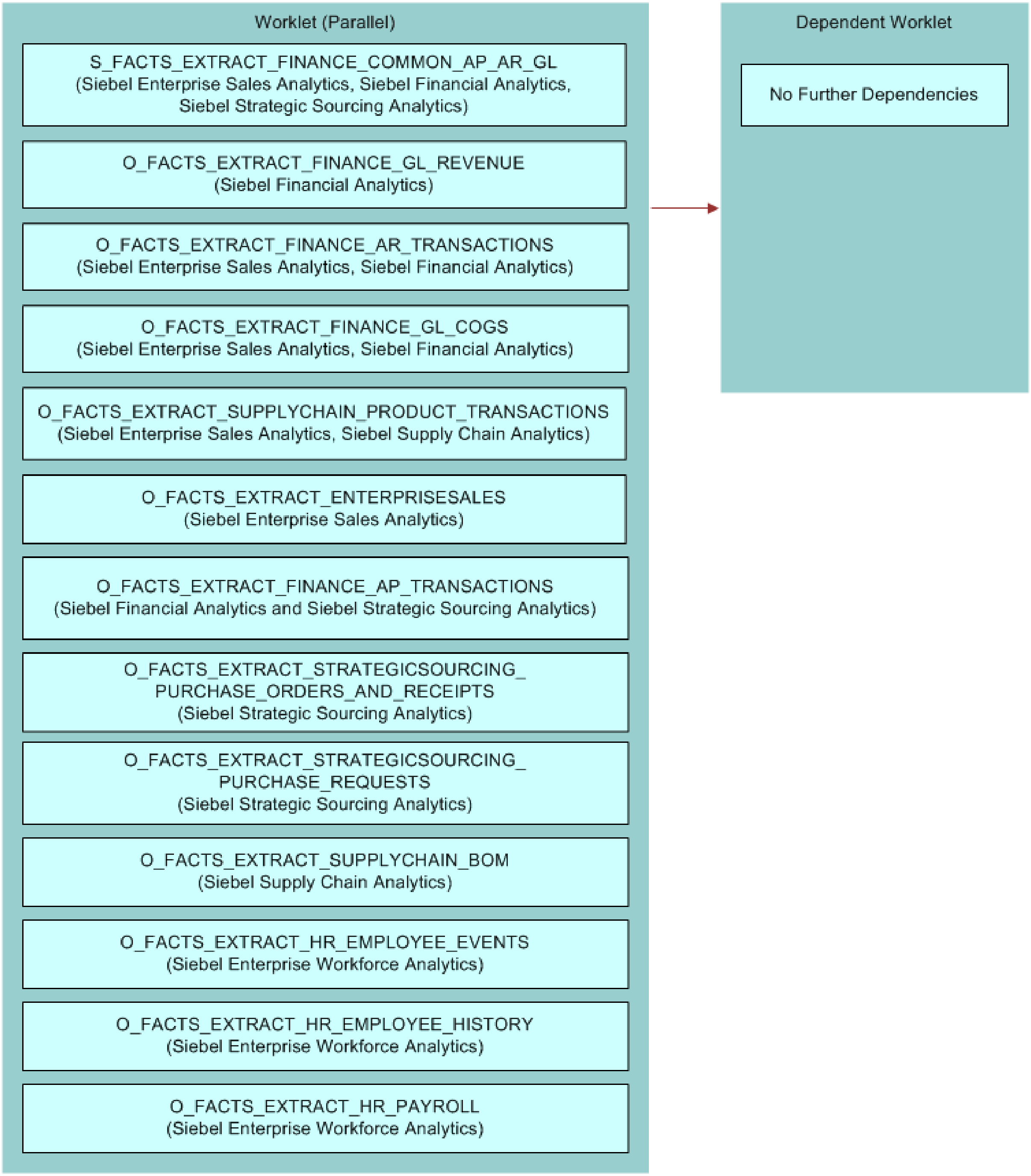

Worklet (Parallel)
S_FACTS_EXTRACT_FINANCE_COMMON_AP_AR_GL (Siebel Enterprise Sales Analytics, Siebel Financial Analytics, Siebel Strategic Sourcing Analytics)
O_FACTS_EXTRACT_FINANCE_GL_REVENUE (Siebel Financial Analytics)
O_FACTS_EXTRACT_FINANCE_AR_TRANSACTIONS (Siebel Enterprise Sales Analytics, Siebel Financial Analytics)
O_FACTS_EXTRACT_FINANCE_GL_COGS (Siebel Enterprise Sales Analytics, Siebel Financial Analytics)
O_FACTS_EXTRACT_SUPPLYCHAIN_PRODUCT_TRANSACTIONS (Siebel Enterprise Sales Analytics, Siebel Supply Chain Analytics)
O_FACTS_EXTRACT_ENTERPRISESALES (Siebel Enterprise Sales Analytics)
O_FACTS_EXTRACT_FINANCE_AP_TRANSACTIONS (Siebel Financial Analytics and Siebel Strategic Sourcing Analytics)
O_FACTS_EXTRACT_STRATEGICSOURCING_ PURCHASE_ORDERS_AND_RECEIPTS (Siebel Strategic Sourcing Analytics)
O_FACTS_EXTRACT_STRATEGICSOURCING_ PURCHASE_REQUESTS (Siebel Strategic Sourcing Analytics)
O_FACTS_EXTRACT_SUPPLYCHAIN_BOM (Siebel Supply Chain Analytics)
O_FACTS_EXTRACT_HR_EMPLOYEE_EVENTS (Siebel Enterprise Workforce Analytics)
O_FACTS_EXTRACT_HR_EMPLOYEE_HISTORY (Siebel Enterprise Workforce Analytics)
O_FACTS_EXTRACT_HR_PAYROLL (Siebel Enterprise Workforce Analytics)
Dependent Worklet
No Further Dependencies

Figure 14. Oracle 11i Facts Extract Worklet

Oracle 11i Facts Load Worklet

Figure 15 illustrates the worklets and their dependant worklets for the Oracle 11i Facts Load worklet.

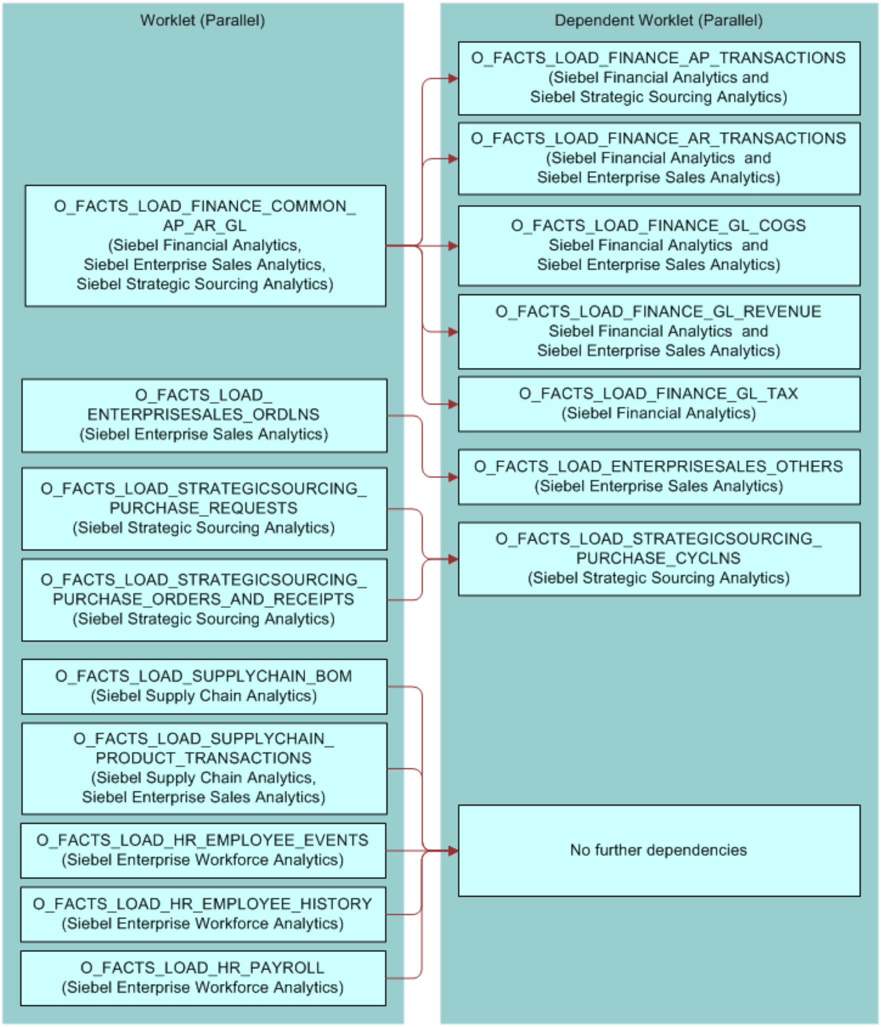

Worklet (Parallel)
Dependent Worklet (Parallel)
O_FACTS_LOAD_FINANCE_COMMON_ AP_AR_GL (Siebel Financial Analytics, Siebel Enterprise Sales Analytics, Siebel Strategic Sourcing Analytics)
O_FACTS_LOAD_ ENTERPRISESALES_ORDLNS (Siebel Enterprise Sales Analytics)
O_FACTS_LOAD_STRATEGICSOURCING_ PURCHASE_REQUESTS (Siebel Strategic Sourcing Analytics)
O_FACTS_LOAD_STRATEGICSOURCING_ PURCHASE_ORDERS_AND_RECEIPTS (Siebel Strategic Sourcing Analytics)
O_FACTS_LOAD_SUPPLYCHAIN_BOM (Siebel Supply Chain Analytics)
O_FACTS_LOAD_SUPPLYCHAIN_ PRODUCT_TRANSACTIONS (Siebel Supply Chain Analytics, Siebel Enterprise Sales Analytics)
O_FACTS_LOAD_HR_EMPLOYEE_EVENTS (Siebel Enterprise Workforce Analytics)
O_FACTS_LOAD_HR_EMPLOYEE_HISTORY (Siebel Enterprise Workforce Analytics)
O_FACTS_LOAD_HR_PAYROLL (Siebel Enterprise Workforce Analytics)
O_FACTS_LOAD_FINANCE_AP_TRANSACTIONS (Siebel Financial Analytics and Siebel Strategic Sourcing Analytics)
O_FACTS_LOAD_FINANCE_AR_TRANSACTIONS (Siebel Financial Analytics and Siebel Enterprise Sales Analytics)
O_FACTS_LOAD_FINANCE_GL_COGS Siebel Financial Analytics and Siebel Enterprise Sales Analytics)
O_FACTS_LOAD_FINANCE_GL_REVENUE Siebel Financial Analytics and Siebel Enterprise Sales Analytics)
O_FACTS_LOAD_FINANCE_GL_TAX (Siebel Financial Analytics)
O_FACTS_LOAD_ENTERPRISESALES_OTHERS (Siebel Enterprise Sales Analytics)
O_FACTS_LOAD_STRATEGICSOURCING_ PURCHASE_CYCLNS (Siebel Strategic Sourcing Analytics)
No further dependencies

Figure 15. Oracle 11i Facts Load Worklet

Universal Dimensions Extract Worklet

Figure 16 illustrates the worklets and their dependant worklets for the Universal Dimensions Load worklet.

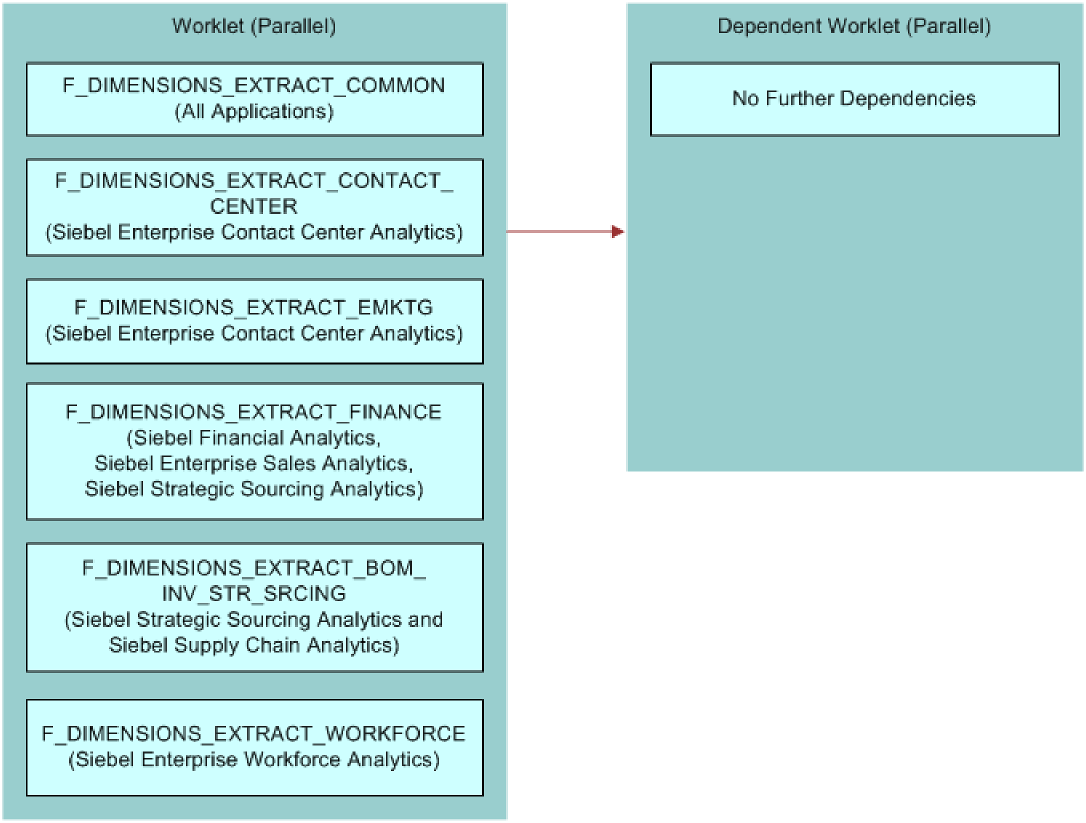

Worklet (Parallel)
F_DIMENSIONS_EXTRACT_COMMON (All Applications)
F_DIMENSIONS_EXTRACT_CONTACT_CENTER (Siebel Enterprise Contact Center Analytics)
F_DIMENSIONS_EXTRACT_EMKTG (Siebel Enterprise Contact Center Analytics)
F_DIMENSIONS_EXTRACT_FINANCE (Siebel Financial Analytics, Siebel Enterprise Sales Analytics, Siebel Strategic Sourcing Analytics)
F_DIMENSIONS_EXTRACT_BOM_INV_STR_SRCING (Siebel Strategic Sourcing Analytics and Siebel Supply Chain Analytics)
F_DIMENSIONS_EXTRACT_WORKFORCE (Siebel Enterprise Workforce Analytics)
Dependent Worklet (Parallel)
No Further Dependencies

Figure 16. Universal Dimensions Extract Worklet

Universal Dimensions Load Worklet

Figure 17 illustrates the worklets and their dependant worklets for the Universal Dimensions Load worklet.

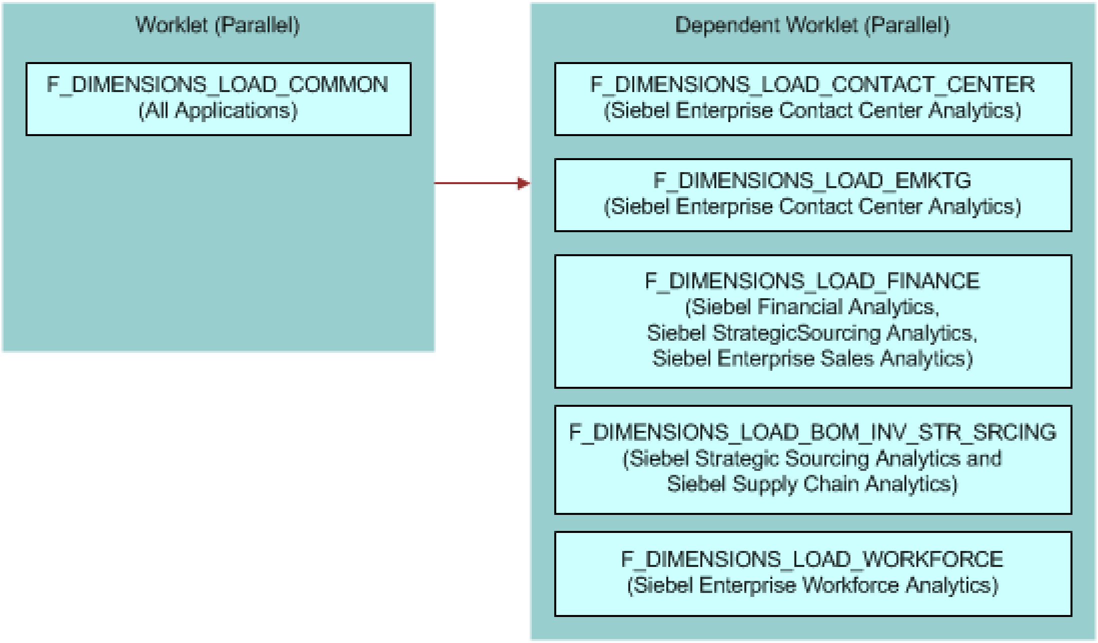

Worklet (Parallel)
F_DIMENSIONS_LOAD_COMMON
(All Applications)
Dependent Worklet (Parallel)
F_DIMENSIONS_LOAD_CONTACT_CENTER
(Siebel Enterprise Contact Center Analytics)
F_DIMENSIONS_LOAD_EMKTG
(Siebel Enterprise Contact Center Analytics)
F_DIMENSIONS_LOAD_FINANCE
(Siebel Financial Analytics,
Siebel StrategicSourcing Analytics,
Siebel Enterprise Sales Analytics)
F_DIMENSIONS_LOAD_BOM_INV_STR_SRCING
(Siebel Strategic Sourcing Analytics and
Siebel Supply Chain Analytics)
F_DIMENSIONS_LOAD_WORKFORCE
(Siebel Enterprise Workforce Analytics)

Figure 17. Universal Dimensions Load Worklet

Universal Fact Extract Worklet

Figure 18 illustrates the worklets and their dependant worklets for the Universal Fact Extract worklet.

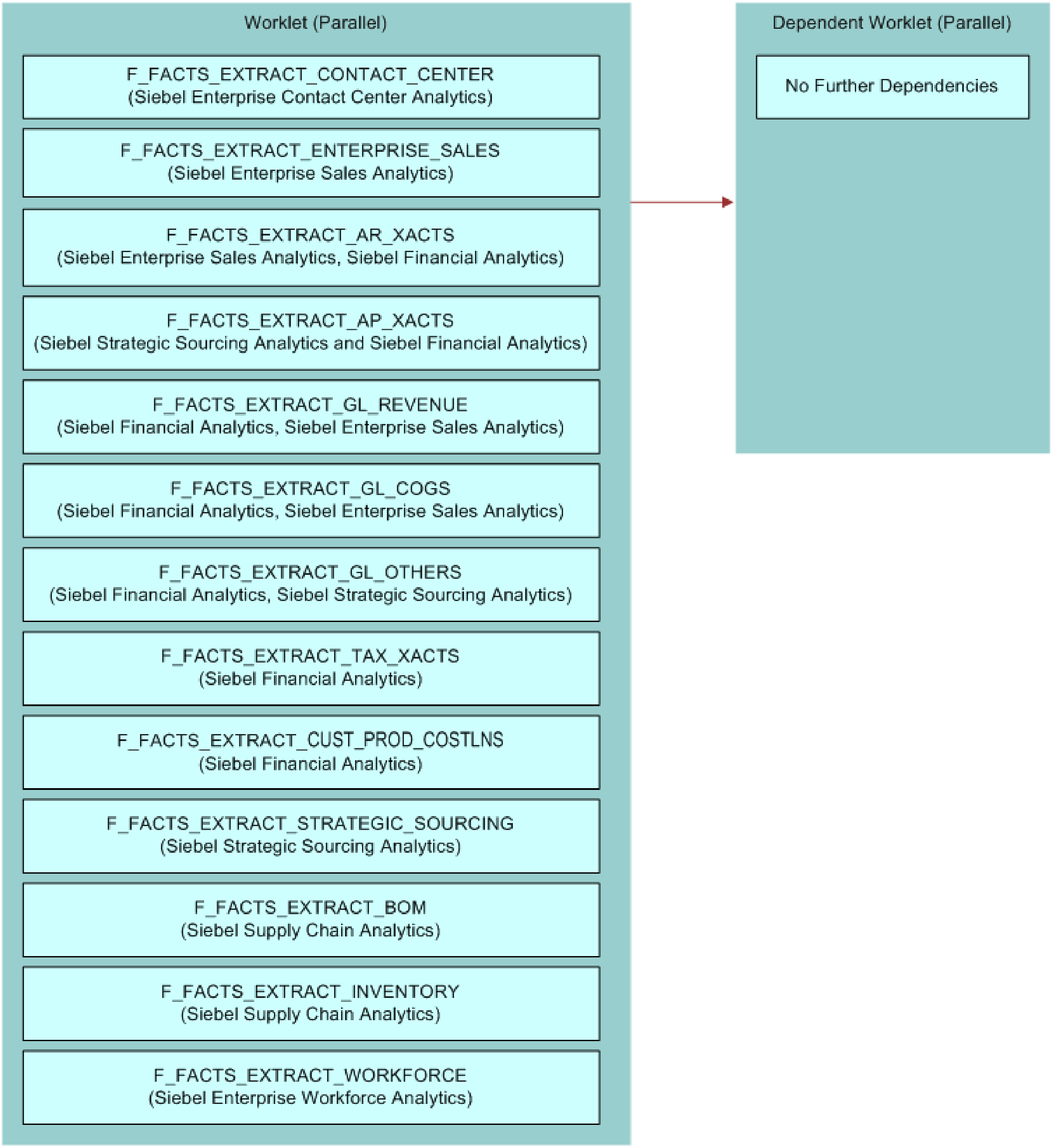

Worklet (Parallel)
F_FACTS_EXTRACT_CONTACT_CENTER
(Siebel Enterprise Contact Center Analytics)
F_FACTS_EXTRACT_ENTERPRISE_SALES
(Siebel Enterprise Sales Analytics)
F_FACTS_EXTRACT_AR_XACTS
(Siebel Enterprise Sales Analytics, Siebel Financial Analytics)
F_FACTS_EXTRACT_AP_XACTS
(Siebel Strategic Sourcing Analytics and Siebel Financial Analytics)
F_FACTS_EXTRACT_GL_REVENUE
(Siebel Financial Analytics, Siebel Enterprise Sales Analytics)
F_FACTS_EXTRACT_GL_COGS
(Siebel Financial Analytics, Siebel Enterprise Sales Analytics)
F_FACTS_EXTRACT_GL_OTHERS
(Siebel Financial Analytics, Siebel Strategic Sourcing Analytics)
F_FACTS_EXTRACT_TAX_XACTS
(Siebel Financial Analytics)
F_FACTS_EXTRACT_CUST_PROD_COSTLNS
(Siebel Financial Analytics)
F_FACTS_EXTRACT_STRATEGIC_SOURCING
(Siebel Strategic Sourcing Analytics)
F_FACTS_EXTRACT_BOM
(Siebel Supply Chain Analytics)
F_FACTS_EXTRACT_INVENTORY
(Siebel Supply Chain Analytics)
F_FACTS_EXTRACT_WORKFORCE
(Siebel Enterprise Workforce Analytics)
Dependent Worklet (Parallel)
No Further Dependencies

Figure 18. Universal Fact Extract Worklet

Universal Fact Load Worklet

Figure 19 illustrates the worklets and their dependant worklets for the Universal Fact Load worklet.

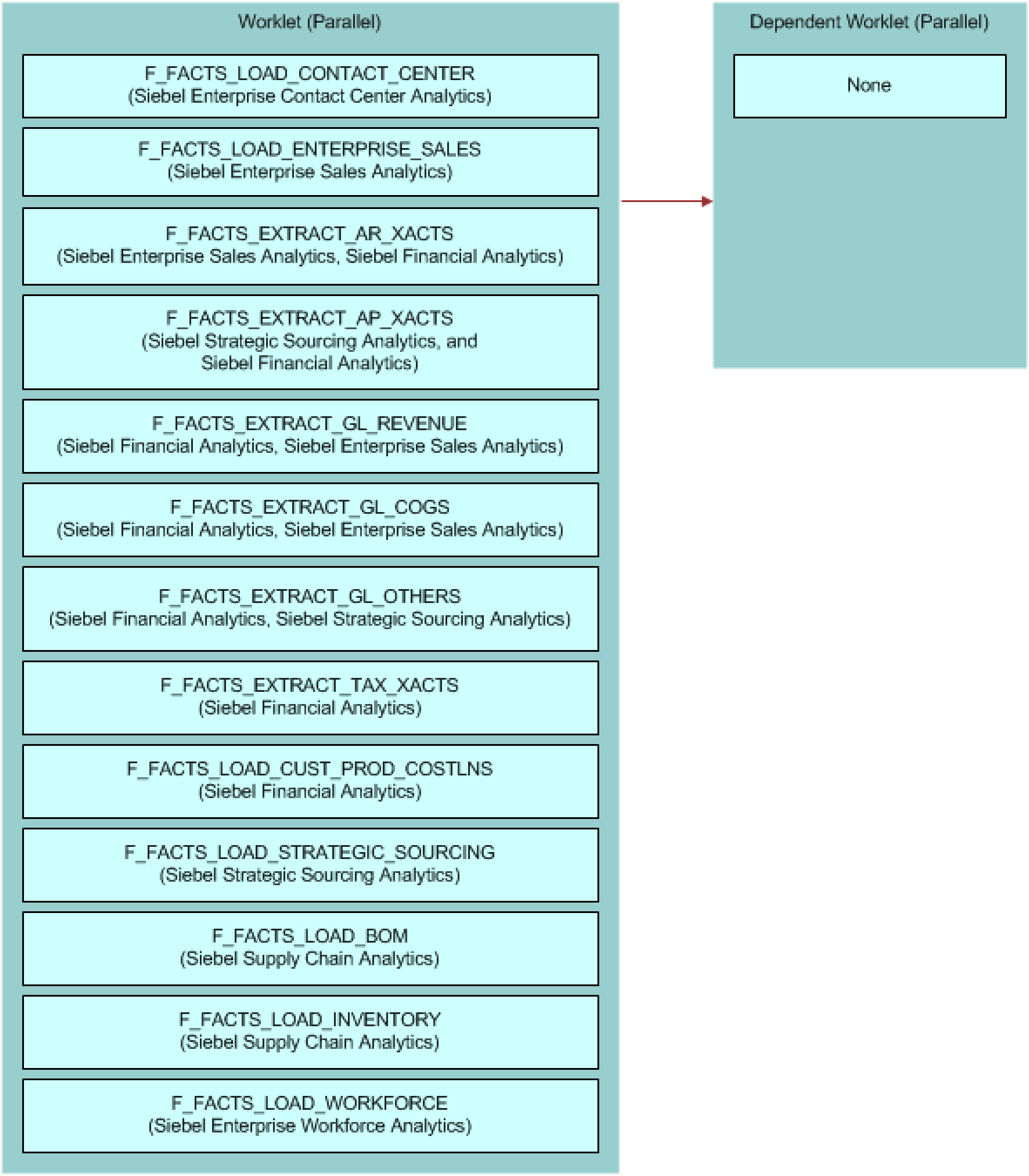

Worklet (Parallel)
F_FACTS_LOAD_CONTACT_CENTER
(Siebel Enterprise Contact Center Analytics)
F_FACTS_LOAD_ENTERPRISE_SALES
(Siebel Enterprise Sales Analytics)
F_FACTS_EXTRACT_AR_XACTS
(Siebel Enterprise Sales Analytics, Siebel Financial Analytics)
F_FACTS_EXTRACT_AP_XACTS
(Siebel Strategic Sourcing Analytics, and Siebel Financial Analytics)
F_FACTS_EXTRACT_GL_REVENUE
(Siebel Financial Analytics, Siebel Enterprise Sales Analytics)
F_FACTS_EXTRACT_GL_COGS
(Siebel Financial Analytics, Siebel Enterprise Sales Analytics)
F_FACTS_EXTRACT_GL_OTHERS
(Siebel Financial Analytics, Siebel Strategic Sourcing Analytics)
F_FACTS_EXTRACT_TAX_XACTS
(Siebel Financial Analytics)
F_FACTS_LOAD_CUST_PROD_COSTLNS
(Siebel Financial Analytics)
F_FACTS_LOAD_STRATEGIC_SOURCING
(Siebel Strategic Sourcing Analytics)
F_FACTS_LOAD_BOM
(Siebel Supply Chain Analytics)
F_FACTS_LOAD_INVENTORY
(Siebel Supply Chain Analytics)
F_FACTS_LOAD_WORKFORCE
(Siebel Enterprise Workforce Analytics)
Dependent Worklet (Parallel)
None

Figure 19. Universal Fact Load Worklet

Post-Load Processing Initial Worklet

Figure 20 illustrates the worklets and their dependant worklets for the Post-Load Processing Initial worklet.

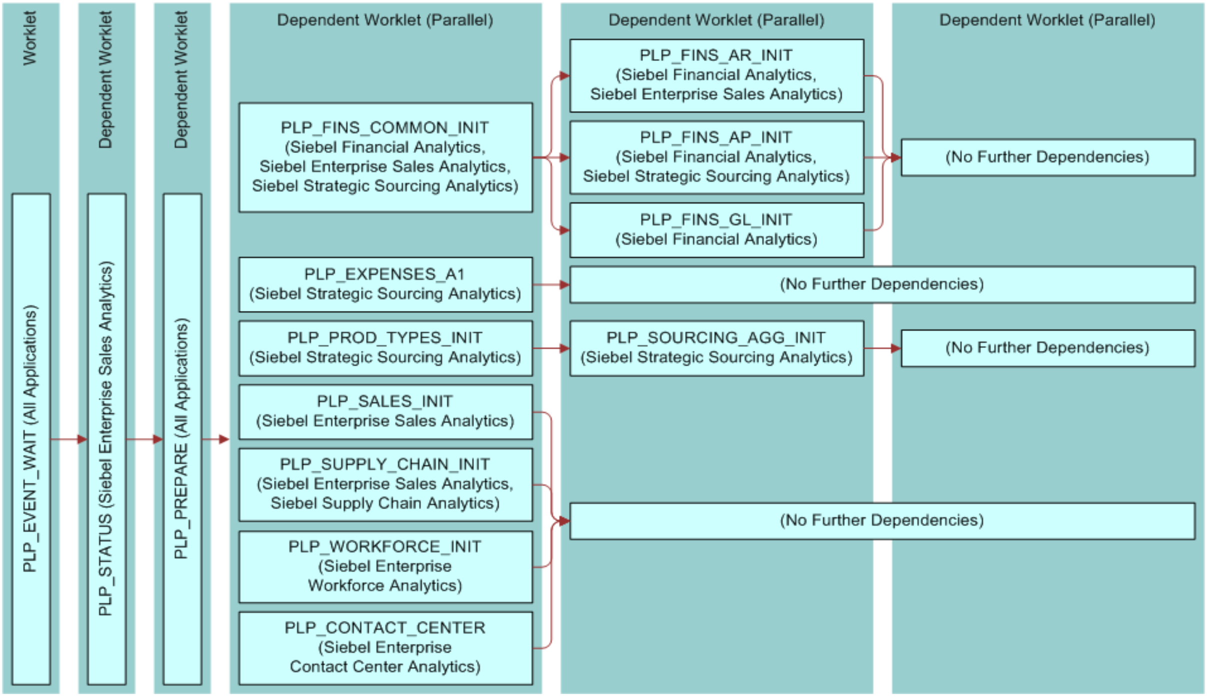

Worklet
Dependent Worklet
Dependent Worklet
Dependent Worklet (Parallel)
Dependent Worklet (Parallel)
Dependent Worklet (Parallel)
PLP_EVENT_WAIT (All Applications)
PLP_STATUS (Siebel Enterprise Sales Analytics)
PLP_PREPARE (All Applications)
PLP_FINS_COMMON_INIT (Siebel Financial Analytics, Siebel Enterprise Sales Analytics, Siebel Strategic Sourcing Analytics)
PLP_FINS_AR_INIT (Siebel Financial Analytics, Siebel Enterprise Sales Analytics)
PLP_FINS_AP_INIT (Siebel Financial Analytics, Siebel Strategic Sourcing Analytics)
PLP_FINS_GL_INIT (Siebel Financial Analytics)
(No Further Dependencies)
PLP_EXPENSES_A1 (Siebel Strategic Sourcing Analytics)
(No Further Dependencies)
PLP_PROD_TYPES_INIT (Siebel Strategic Sourcing Analytics)
PLP_SOURCING_AGG_INIT (Siebel Strategic Sourcing Analytics)
(No Further Dependencies)
PLP_SALES_INIT (Siebel Enterprise Sales Analytics)
PLP_SUPPLY_CHAIN_INIT (Siebel Enterprise Sales Analytics, Siebel Supply Chain Analytics)
PLP_WORKFORCE_INIT (Siebel Enterprise Workforce Analytics)
PLP_CONTACT_CENTER (Siebel Enterprise Contact Center Analytics)
(No Further Dependencies)

Figure 20. Post-Load Processing Initial Worklet

Post-Load Processing Incremental Worklet

Figure 21 illustrates the worklets and their dependant worklets for the Post-Load Processing Incremental worklet.

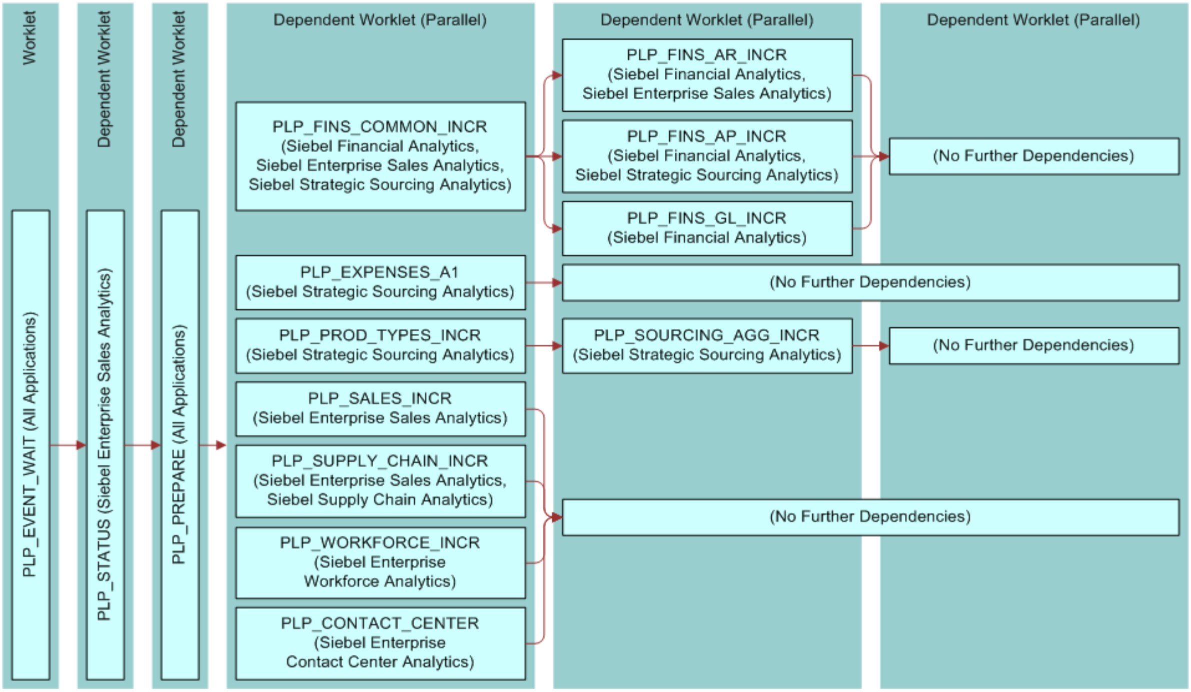

Worklet
Dependent Worklet
Dependent Worklet
Dependent Worklet (Parallel)
Dependent Worklet (Parallel)
Dependent Worklet (Parallel)
PLP_EVENT_WAIT (All Applications)
PLP_STATUS (Siebel Enterprise Sales Analytics)
PLP_PREPARE (All Applications)
PLP_FINS_COMMON_INCR (Siebel Financial Analytics, Siebel Enterprise Sales Analytics, Siebel Strategic Sourcing Analytics)
PLP_FINS_AR_INCR (Siebel Financial Analytics, Siebel Enterprise Sales Analytics)
PLP_FINS_AP_INCR (Siebel Financial Analytics, Siebel Strategic Sourcing Analytics)
PLP_FINS_GL_INCR (Siebel Financial Analytics)
(No Further Dependencies)
PLP_EXPENSES_A1 (Siebel Strategic Sourcing Analytics)
(No Further Dependencies)
PLP_PROD_TYPES_INCR (Siebel Strategic Sourcing Analytics)
PLP_SOURCING_AGG_INCR (Siebel Strategic Sourcing Analytics)
(No Further Dependencies)
PLP_SALES_INCR (Siebel Enterprise Sales Analytics)
PLP_SUPPLY_CHAIN_INCR (Siebel Enterprise Sales Analytics, Siebel Supply Chain Analytics)
PLP_WORKFORCE_INCR (Siebel Enterprise Workforce Analytics)
PLP_CONTACT_CENTER (Siebel Enterprise Contact Center Analytics)
(No Further Dependencies)

Figure 21. Post-Load Processing Incremental Worklet

About Incremental Deployment

Incremental deployment allows you modify an existing Siebel Business Analytics configuration by adding new applications.

Incremental deployment occurs when:

- You purchase one or more Siebel Business Analytics applications
- Your Siebel Business Analytics environment is configured and running these applications
- You need to add one or more new applications with the same source system

For example, you have an environment with Siebel Enterprise Sales Analytics for Oracle 11i and Siebel Financial Analytics for Oracle 11i. You then purchase two new applications, for example, Siebel Strategic Sourcing Analytics for Oracle 11i and Supply Chain Analytics for Oracle 11i. You need to merge all four of your applications and make sure that your ETL processes are successfully integrated.

NOTE: If you have deployed an application for one source system (for example, Oracle 11i) and you plan to deploy another application for a universal source system, see About Deploying Source Systems with Universal Source Systems for Incremental Deployment on page 112.

Setting Up Incremental Deployment

Incremental deployment allows you modify an existing Siebel Business Analytics configuration by adding new applications.

The following procedure incrementally deploys Siebel Enterprise Sales Analytics for Oracle 11i to an running Siebel Financial Analytics for Oracle 11i production environment. The full ETL run is completed by Siebel Financial Analytics and it is successfully running daily incremental ETL runs. The Siebel Enterprise Sales Analytics needs to complete a full ETL run and join Siebel Financial Analytics in future incremental ETL runs.

NOTE: It is recommended that you backup your production repository and make a snapshot of your data warehouse before using the following procedure.

To set up incremental deployment

1. Import the Siebel Enterprise Sales Analytics workflows into the Siebel Business Analytics repository.

   For more information on importing application workflows into the repository, see Importing Application Workflows into the Repository on page 45.

2. Start the Workflow Manager, connect to your development repository, and note the new reusable worklets, under the non-reusable shell worklets, that are specific to the ORACLE11i_EnterpriseSales_Application workflow.

3. Create the combined Siebel Enterprise Sales Analytics and Siebel Financial Analytics workflow:

   a. Rename the ORACLE11i_Finance_Application workflow to ORACLE11i_Finance_And_Sales_Application.

   b. Add the reusable worklets from Step 2 to the non-reusable shell worklets in the ORACLE11i_Finance_And_Sales_Application workflow.

      For information on Siebel Business Analytics workflows and their dependent worklets, see Process of Building Multi-Application Workflows for the Siebel Business Analytics on page 89.

   c. For all the new reusable worklets, check the dependencies, and note all the unique sessions that they use.

   d. On the repository navigator, right-click the worklets in these sessions, and click Dependencies.

   e. In the Dependencies dialog box, select only Sessions as object types, and deselect all others.

4. Open the `file_parameters_ora11i.csv` parameter file using Notepad, and edit this file to perform a full ETL run for all the new sessions:

   a. Change the PARM_DVALUE_1 column value to 19700101000000,

   b. Change the PARM_NVALUE_1 column value to 0.

5. Run the ORACLE11i_Finance_And_Sales_Application workflow.

6. Open the file_parameters_ora11i.csv parameter file using Notepad, and change the PARM_NVALUE_1 column value to 1, 2, or 3, depending on your incremental window choice.

7. Run the Siebel Enterprise Sales Analytics PLP initial workflow:

   a. Open the Siebel Enterprise Sales Analytics initial workflow, PLP_EnterpriseSales_Application_INIT.

   b. Disable all the reusable worklets that are present in the Finance Initial Workflow, PLP_Finance_Application_INIT.

   c. Run the PLP_EnterpriseSales_Application_INIT initial workflow.

8. Create the PLP incremental workflow for the two applications:

   a. Rename the PLP_Finance_Application_INCR workflow to PLP_Finance_And_EnterpriseSales_Application_INCR.

   b. Add the Siebel Enterprise Sales Analytics reusable worklets to the PLP_Finance_And_EnterpriseSales_Application_INCR.

      For more information on building multi-application workflows, see Process of Building Multi-Application Workflows for the Siebel Business Analytics on page 89.

9. Run the merged workflows:

   a. ORACLE11i_Finance_And_Sales_Application from the base folder.

   b. PLP_Finance_And_EnterpriseSales_Application_INCR from the Post Load Processing folder.

   NOTE: You can run these workflow frequently to verify the data before you move the repository and the data warehouse to production.

About Deploying Source Systems with Universal Source Systems for Incremental Deployment

The following section discusses the implementation of one or more applications from a specific source system (for example, SAP R/3, Oracle 11i, and PeopleSoft 8.4) with one or more applications using universal source. For example, if you implement Siebel Financial Analytics for Oracle 11i with Siebel Strategic Sourcing Analytics using universal source.

In a cross folder deployment environment, there are two possible scenarios:

- The data sources are mutually exclusive:
  - The dimensional data sources are exclusive. For example, if you have a data source for a Customers dimension in the Oracle 11i source system, you cannot have a similar data source in your universal source.
  - The dimensional data sources are partially exclusive. For example, you may have a data source for the Customers dimension in the Oracle 11i source system and in the universal source, but the list of customers in both of these sources are exclusive.

  - The dimensional data sources are partially exclusive and they can have data similarities. For example, you might have a data source for the Customers dimension in the Oracle source system and in the universal source, and some customers appear in both sources. You can decide to use these customers as different sources and load them separately into the common data warehouse.

  For more information on configuring mutually exclusive data sources, see Configuring Mutually Exclusive Source Systems for Incremental Deployment on page 113

- The data sources are not mutually exclusive.

  The dimensional data sources are partially exclusive and they can have data similarities. For example, you might have a data source for the Customers dimension in the Oracle source system and in the universal source, and some customers appear in both sources. You decide to join these customers to get a uniform view from your data warehouse.

  The assumption here is that one data source is a subset of the other data source. For example, the master list of customers are stored in Oracle 11i source system, whereas a subset of customers (for example, web customers) are stored in the universal source. You need to populate the data warehouse and analyze the data with the Oracle 11i source system.

  For more information on configuring nonmutually exclusive data sources, see Configuring Nonmutually Exclusive Source Systems for Incremental Deployment on page 115

For mutually exclusive data sources, you have a disjointed set of data loaded in the Siebel Customer-Centric Enterprise Warehouse. The entities in the Siebel Customer-Centric Enterprise Warehouse are separated by the value of Source ID (SOURCE_ID) column. The advantages of this separation are:

- You can query against a particular source system by including the SOURCE_ID column into your query filter.
- If you want to see the number of orders placed by a customer, you can add up the distinct orders grouping by the customer.
- There are no large changes in the Siebel Business Analytics metadata repository. The fact tables are connected to the dimension tables which are populated by the two distinct sources, and there are no conflicts to resolve the surrogate keys.

  For more information on modifying session parameters for source system parameter files, see About Modifying Session Parameters for Initial and Incremental Loads on page 61.

Configuring Mutually Exclusive Source Systems for Incremental Deployment

Every Siebel Customer-Centric Enterprise Warehouse table has a Source ID (SOURCE_ID) column. This column identifies what source system a record originates from. For example OAP11I is the value for the Oracle 11i source system. The data sources in the Siebel Customer-Centric Enterprise Warehouse are separated by the value of the Source ID column.

The following procedure provides information on loading the Siebel Customer-Centric Enterprise Warehouse for mutually exclusive data sources.

To configure mutually exclusive source systems

1 Import both the applications into a new repository.

For more information on importing applications, see About the Incremental Deployment of the Siebel Business Analytics Repository on page 87.

2 Create multi-application Post Load Initial and Post Load Incremental workflows.

For more information on building multi-application workflows, see Process of Building Multi-Application Workflows for the Siebel Business Analytics on page 89.

For example, for Siebel Enterprise Sales Analytics (Oracle 11i) and Siebel Enterprise Contact Center Analytics (universal source), create workflows called PLP_EnterpriseSales_ContactCenter_Application_INIT and PLP_EnterpriseSales_ContactCenter_Application_INCR.

3 Start the application workflows.

For example, for a full ETL load for Siebel Enterprise Sales Analytics (Oracle 11i) and Siebel Enterprise Contact Center Analytics (universal source), run the following workflows:

a The INITIALIZE workflow in the Siebel Applications folder.

b ORACLE11i_EnterpriseSales_Application workflow in the Configuration for Oracle Applications v11i folder.

c Universal_All_Applications_Common_Initialize workflow in the Configuration for Universal Source folder.

d Universal_EnterpriseContactCenter_Application workflow in the Configuration for Universal Source folder.

e PLP_EnterpriseSales_ContactCenter_Application_INIT in the Configuration for Post Load Processing folder.

This workflow is created in Step 2 on page 114.

4 Change the source system parameter file (for example, `file_parameters_ora11i.csv`) and provide an incremental value in the PARM_NVALUE_1 column.

For more information on modifying session parameters for parameter files, see About Modifying Session Parameters for Initial and Incremental Loads on page 61.

5 Start the incremental ETL runs.

For example, for an incremental ETL load for Siebel Enterprise Sales Analytics (Oracle 11i) and Siebel Enterprise Contact Center Analytics (universal source), run the following workflows:

a ORACLE11i_EnterpriseSales_Application workflow in the Configuration for Oracle Applications v11i folder.

b Universal_EnterpriseContactCenter_Application workflow in the Configuration for Universal Source folder.

c PLP_EnterpriseSales_ContactCenter_Application_INCR in the Configuration for Post Load Processing folder.

This workflow is created in Step 2 on page 114.

Configuring Nonmutually Exclusive Source Systems for Incremental Deployment

For applications such as Siebel Enterprise Sales Analytics (Oracle 11i) and Siebel Enterprise Contact Center Analytics (universal source), there are common dimensions, for example, Customers, Products and so on. You need to choose the master source for each common dimension. For application specific dimensions, for example, Sales Status for Sales, it is not necessary to choose a master source.

It is not necessary to select a master source for fact tables. However, for fact tables that refer to common dimensions (for example, Customers or Products), you need to take necessary care to so that they resolve correctly against the correct dimension. The `<DIMENSION>_ID` column in the fact table must be formed correctly so that it matches with the format of the `KEY_ID` column for the corresponding dimension. These changes are normally carried out in the Source Adapter mapplets of the fact table being loaded.

To configure nonmutually exclusive source systems

1. Import both the applications into a new repository.

   For more information on importing applications into a repository, see About the Incremental Deployment of the Siebel Business Analytics Repository on page 87.

2. Create multi-application Post Load Initial and Post Load Incremental workflows.

   For more information on building multi-application workflows, see Process of Building Multi-Application Workflows for the Siebel Business Analytics on page 89.

   For example, for Siebel Enterprise Sales Analytics (Oracle 11i) and Siebel Enterprise Contact Center Analytics (universal source), create workflows called PLP_EnterpriseSales_ContactCenter_Application_INIT and PLP_EnterpriseSales_ContactCenter_Application_INCR.

3. Identify the tables used in both applications, and for each table, identify the main source system, for example, Oracle 11i.

4. Disable the dimension extract and load sessions in the application workflow.

   For example, for Siebel Enterprise Sales Analytics (Oracle 11i), disable the dimension extract and load sessions in the ORACLE11i_EnterpriseSales_Application workflow, for the tables you are loading from the Oracle 11i source system.

   For Siebel Enterprise Contact Center Analytics (universal source), disable the dimension extract and load sessions in the Universal_EnterpriseContactCenter_Application workflow for the tables you are loading from the Oracle 11i source system.

5. Change the PARM_SVALUE_2 column value in the source system parameter file.

   For example, for the `file_parameters_ora11i.csv` parameters file, change the PARM_SVALUE_2 column value from OAP11I to SLS_ECC. For the `file_parameters_univ.csv` parameters file, change the PARM_SVALUE_2 column value from GENERIC to SLS_ECC.

   NOTE: You must not change the preconfigured values of ORA11I and UNIV for the PARM_TYPE and SOURCE_ID columns.

   For Siebel Contact Center Analytics you need to also run the Universal_All_Applications_Common_Initialize workflow once. This workflow loads Siebel Customer-Centric Enterprise Warehouse domain values—IA_STATUS, IA_EVENT_TYPES and IA_CHNL_TYPES. You can load these domain values through your universal source system.

6. Start the application workflows.

   For example, for a full ETL load for Siebel Enterprise Sales Analytics (Oracle 11i) and Siebel Enterprise Contact Center Analytics (universal source), run the following workflows:

   a. INITIALIZE workflow in the Siebel Applications folder.

   b. ORACLE11i_EnterpriseSales_Application workflow in the Configuration for Oracle Applications v11i folder.

   c. Universal_All_Applications_Common_Initialize workflow in the Configuration for Universal Source folder.

   d. Universal_EnterpriseContactCenter_Application workflow in the Configuration for Universal Source folder

   e. PLP_EnterpriseSales_ContactCenter_Application_INIT in the Configuration for Post Load Processing folder.

      This workflow is created in Step 2 on page 115.

7. Change the source system parameter files, `file_parameters_ora11i.csv` and `file_parameters_univ.csv`, and provide an incremental value in the PARM_NVALUE_1 column.

   For more information on modifying session parameters for parameter files, see About Modifying Session Parameters for Initial and Incremental Loads on page 61.

8. Start the incremental ETL runs.

   For example, for an incremental ETL load for Siebel Enterprise Sales Analytics (Oracle 11i) and Siebel Enterprise Contact Center Analytics (universal source), run the following workflows:

   a. ORACLE11i_EnterpriseSales_Application workflow in the Configuration for Oracle Applications v11i folder.

   b. Universal_EnterpriseContactCenter_Application workflow in the Configuration for Universal Source folder.

   c. PLP_EnterpriseSales_ContactCenter_Application_INCR in the Configuration for Post Load Processing folder.

      This workflow is created in Step 2 on page 115.

8 Configuring Common Components of the Siebel Customer-Centric Enterprise Warehouse

This chapter provides procedural information on how to configure components that are common, regardless of which application you purchased.

It contains the following topics:

Configuring Extracts

Each application has prepackaged logic to extract particular data from a particular source. This section discusses how to capture all data relevant to your reports and ad hoc queries by addressing what type of records you want and do not want to load into the data warehouse, and includes the following topics:

Disabling Workflow Sessions

After you determine that the information you want in your data warehouse, you may choose to enable or disable particular workflow sessions. If you do so, remember that most sessions must be disabled in pairs. For example, if you decide you do not want to build a particular table, you must disable both the extract and load sessions that support that object.

To disable a workflow

1. In PowerCenter Workflow Manager, open the applicable source system configuration folder.
2. On the Workflow menu, click Edit to open the Edit Workflow window.
3. Select the Disabled check box to disable the workflow, and click OK.

To disable a workflow session

1. In PowerCenter Workflow Manager, open the applicable source system configuration folder.
2. On the Workflow menu, click Edit to open the Edit Workflow window.
3. Select the Disable this task check box to disable the session, and click OK.

Extracting Additional Data

You can configure extract mappings and mapplets in the Siebel Customer-Centric Enterprise Warehouse to accommodate additional source data. For example, if your business divides customer information into separate tables based on region, then you would have to set up the extract mapping to include data from both tables.

You can modify extract mappings so that new data is loaded into extension columns that act as placeholders for additional data. Extension columns make it possible to extend any fact or dimension table without changing the schematic structure of the Siebel Customer-Centric Enterprise Warehouse or making modifications to the load mapping, as the load mappings already include the extension columns. Keeping the data model intact allows you to implement upgrades without losing any customization.

Extracting New Data Using an Existing Source Table

Extract mappings generally consist of a source table or Business Component, an Expression transformation, and a staging table. If you want to extract new data using the existing mapping, you have to modify the extract mapping to include the new data by performing the following tasks:

To modify an existing mapping to include new data

1. Modify the existing Business Component to extract information from the source, and add it to an appropriate extension column.

   TIP: You can perform calculation transformations in the Business Component mapplet of the extract mapping or in the Source Adapter mapplet of the load mapping. However, do not use performance-expensive calculations in the extract that could tie up your source transaction system. For these types of calculations, it is recommended that you perform them in the Source Adapter mapplet in the load mapping.

2. Modify the Expression transformation to perform any necessary transformations.
3. Connect all input and output ports within the extract mapping so that the data moves from the source or Business Component to the Expression transformation, and finally to the staging table's appropriate extension column.

You have to determine which type of extension column to map the data to in the staging table.

Extracting Data from a New Source Table

Business Components are packaged as mapplets, which reside in source-specific folders within the repository. Business Components are used to extract data from the source system. You can configure these mapplets to perform the following:

- Extract data from a new source table
- Set incremental extraction logic

The following procedure contains instructions for adding a new table to the Business Component. The procedure includes adding a new source definition, connecting the ports to the Source Qualifier, editing the Source Qualifier, connecting the ports to the Output transformation, and editing the Output transformation.

To add a new source table to an existing Business Component mapplet

1. In PowerCenter Designer, open the applicable source system configuration folder.
2. Open Mapplet Designer tool.
3. Drag the Business Component mapplet into Mapplet Designer to view the transformations that comprise the Business Component.
4. Expand the Sources folder, and copy a source table into the mapplet by dragging and dropping the table into Mapplet Designer.
5. Connect the applicable ports from the new source definition to the Source Qualifier by clicking on the port in the new source table and dragging it to the connecting port in the Source Qualifier.
6. Double-click the Source Qualifier to open the Edit Transformations box.

   In the Ports tab, make any changes to the new ports for data type, precision, scale, or all these values, as necessary.

7 Connect the applicable ports from the Source Qualifier to the Mapplet Output transformation (MAPO).

NOTE: In some cases, the Business Component contains an Expression transformation between the Source Qualifier and the MAPO.

8 In the Properties tab, make changes to the SQL statement as necessary.

9 Validate and save your changes to the repository.

Filtering the Data Extract

The Siebel Customer-Centric Enterprise Warehouse prepackages some filters to extract only particular types of records. For example, the Siebel Enterprise Sales Analytics application, by default, extracts only booked orders. Given your unique business needs, you may need to extract nonbooked orders as well. In general, you can manage all extract filters in the same way, and the following procedure describes how to modify an extract. However, for information about application-specific extract filters, see the application-specific configuration chapters.

To modify extract filters

1 In PowerCenter Designer, open the applicable source system configuration folder.

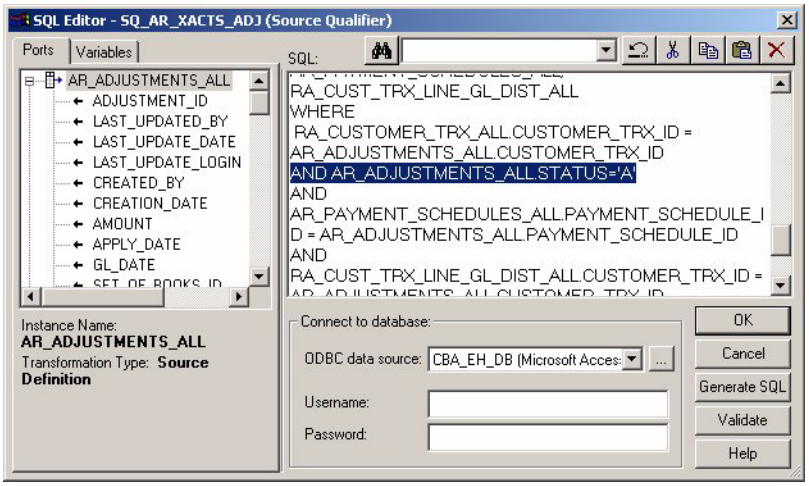
2 Open Mapplet Designer.

3 Drag the Business Component mapplet into Mapplet Designer.

4 Double-click the Source Qualifier to open the Edit Transformations window, and select the Properties tab.

5 In both the User Defined Join field and in the SQL Query field, remove or add a filter statement.

For example, if you want to change the Accounts Receivable Schedules filter so that it is not restricted to completed schedules only, you would remove the statement, as shown in the following figure.



```
RA_CUST_TRX_LINE_GL_DIST_ALL
WHERE
 RA_CUSTOMER_TRX_ALL.CUSTOMER_TRX_ID =
AR_ADJUSTMENTS_ALL.CUSTOMER_TRX_ID
AND AR_ADJUSTMENTS_ALL.STATUS='A'
AND
AR_PAYMENT_SCHEDULES_ALL.PAYMENT_SCHEDULE_I
D = AR_ADJUSTMENTS_ALL.PAYMENT_SCHEDULE_ID
AND
RA_CUST_TRX_LINE_GL_DIST_ALL.CUSTOMER_TRX_ID =
AR_ADJUSTMENTS_ALL.CUSTOMER_TRX_ID
```

Instance Name: **AR_ADJUSTMENTS_ALL**

Transformation Type: **Source Definition**

ODBC data source: CBA_EH_DB (Microsoft Acces:

Username:

Password:

6 Validate and save your changes to the repository.

NOTE: If a primary extract exists, you must modify both the regular mapplet and the primary extract mapplet. For information on primary extract mappings, see About Primary Extract and Delete Mappings Process on page 124.

Setting Up the Delimiter for Source Files

You must make sure that the delimiter used in the CSV files is not present in the source data. Siebel Customer-Centric Enterprise Warehouse is preconfigured with a comma as the source file delimiter. If your data contains commas, than you need to enclose the data fields using a character that is not present in the data set. For example, common enclosing characters include single quotes and double quotes.

To set up the delimiter for source files

1 Open the CSV file.

2 Enclose the data fields with the enclosing character that you have identified.

   You can identify an enclosing character by identifying a character that is not present in the source data. Common enclosing characters include single quotes and double quotes.

3 Save and close the CSV file.

4 Identify all the source definitions associated with the modified files.

5 Change the properties for each of these source definitions to use the enclosing character.

6 Validate and save your changes to the repository.

Configuring Loads

The Siebel Customer-Centric Enterprise Warehouse prepackages load mappings for every data warehouse table. Within each load mapping, every input and output port for each extension column is already connected. Thus, if you connect new data to a staging table's extension column using the extract mapping, the default configuration pulls that data through all the load mapping's components and inserts it into the corresponding data warehouse table. However, because the load mapping does not transform data in extension columns, you may need to reconfigure the load mapping to do so. Each of the following sections describes potential configuration approaches:

- Deriving a metric or attribute from other attributes in other staging tables

  If you are going to derive a new metric or attribute from other metrics, attributes, or both in other staging tables, then you need to join all staging tables that contain the data you need. You can join these tables in the load mapping, using the Source Qualifier. For information on how to join tables in the Staging Area, see Joining Objects in the Staging Area on page 122.

- Performing calculation transformations

  It is recommended that you modify the Expression transformation to perform calculation transformations in the Source Adapter mapplet.

- Performing a lookup override at the session-level

  If you want to store an additional domained attribute, and you stored the code and code name in the `IA_CODES` table, then you need to modify the load session's lookup. You have to modify the SQL statement to look up the code using the correct category. This lookup returns the appropriate code name from the `IA_CODES` table, and the ADI loads both the supplied code and the lookup code name into the target table.

- Resolving currency conversion issues

  If you are storing amount metrics, you only need to supply the amounts in one of the three currencies—document, local, or group currency. By loading one of these three values, the ADI contains logic for deriving the other two. For more information on potential configuration points with currency, see Process for Configuring Currencies on page 138. For more information on how each of the three types of currency is handled by the ADI, see About Document, Local, and Group Currencies on page 136.

Joining Objects in the Staging Area

If you want to join objects from multiple tables, you can do so in the Staging Area. This section describes how to join objects in the Staging Area by configuring the load mapping.

To join staging tables in the Staging Area

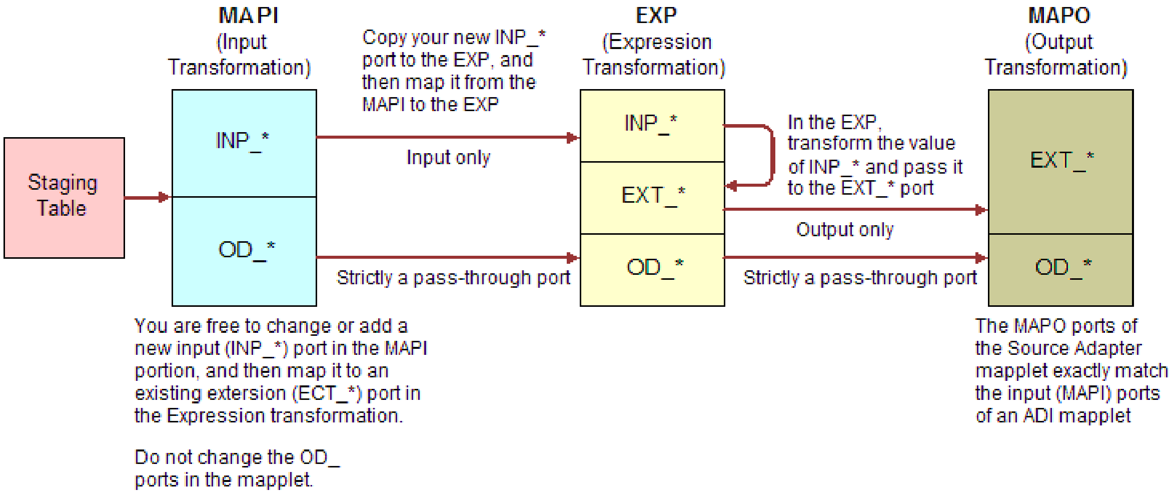
1. In PowerCenter Designer, open the applicable source system configuration folder.
2. Open the applicable load mapping.
3. Copy the table you want to add to the load mapping by dragging and dropping it into the mapping definition in Mapping Designer.

   Perform this step for all tables you want to join in the Staging Area.
4. Drag and drop the columns you wish to join from the source definition to the Source Qualifier.
5. Double-click the Source Qualifier transformation to open the Edit Transformations box, and select the Properties tab.
6. Select the small arrow in the Value column to open the SQL Editor.
7. Edit the SQL statement to add the join conditions between the new table and the existing table in the mapping.
8. Drag and drop those columns from the Source Qualifier to the respective ports in the Source Adapter mapplet.
9. Validate and save your changes to the repository.

Creating a Source Adapter Mapplet

The majority of all source-specific transformations occur in the Source Adapter mapplet; source-independent transformations generally take place in the Analytic Data Interface (ADI). The Source Adapter mapplet converts source-specific data elements that come from a staging table into those data elements required by the source-independent ADI.

Figure 22 illustrates the three components of the Source Adapter mapplet that allow transformations of data to occur. The three components are Mapplet Input (MAPI), Expression transformation (EXP), and Mapplet Output (MAPO).



Copy your new INP_* port to the EXP, and then map it from the MAPI to the EXP

In the EXP, transform the value of INP_* and pass it to the EXT_* port

You are free to change or add a new input (INP_*) port in the MAPI portion, and then map it to an existing extersion (ECT_*) port in the Expression transformation.

Do not change the OD_ ports in the mapplet.

The MAPO ports of the Source Adapter mapplet exactly match the input (MAPI) ports of an ADI mapplet

Figure 22. Input, Expression, and Output Ports of a Source Adapter Mapplet

In Figure 22, notice that the MAPI receives data from the Staging table. This data is passed through ports prefixed with INP_. If the input data is transformed, the data is passed to the Expression transformation (EXP) as input only. After the data is transformed, it is output through a new port, which is prefixed with EXT_. If the data is not transformed, it comes in as input-only and leaves through an output-only port.

If you want to add a new transformation, you must add a new port to contain the expression that is used to transform the data.

To add a new port to the Source Adapter mapplet

1 In PowerCenter Designer, open the applicable source system configuration folder.

2 Open the applicable Source Adapter mapplet.

3 Double-click the MAPI component of the mapplet, and add a new input port following the `INP_*` naming convention.

4 Copy the new input port from the MAPI to the Expression transformation.

5 Connect the new port from the MAPI to the Expression transformation.

6 In the Expression transformation, uncheck the Output indicator for the new input port; you use the value from this port in an transformation expression.

7 Perform any necessary transformations within the Expression transformation.

The transformed data is passed out of an EXT_* output-only port.

8 Connect the port from the Expression transformation to the MAPO.

Figure 22 shows that the ports in the Mapplet Output match the input ports of the ADI mapplet exactly, with the exception of new ports. If you are adding a new port, you must connect it to an extension column, because you cannot add the new port to the ADI.

9 Validate and save your repository.

Filtering and Deleting Records

In a typical implementation, records that are deleted from your source system are not removed from the Siebel Customer-Centric Enterprise Warehouse. If you want to delete records in the data warehouse that were removed from the source system's database and archived in a separate database, you must enable the primary extract and delete mappings.

Primary extract mappings flag records that are deleted from the data warehouse. Delete mappings perform the deletion action. 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 remove them from the data warehouse as well.

CAUTION: It is important to note that delete and primary extract mappings must always be disabled together; you may not disable only one type.

The delete and primary extract sessions are found in each application's fact-extract and fact-load subbatches, and are stored in the `[SOURCE ABBREVIATION]_[SUBJECT]_MAIN` workflow.

About Primary Extract and Delete Mappings Process

Before you decide to enable primary extract and delete sessions, it is important to understand their function within the Siebel Customer-Centric Enterprise 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 Siebel Customer-Centric Enterprise Warehouse table.

The primary extract mappings perform a full extract of the primary keys from the source system. Although many rows are generated from this extract, the data only extracts the Key ID and Source ID information from the source table. The primary extract mappings load these two columns into staging tables that are marked with a *_PE suffix.

Figure 23 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 Siebel Customer-Centric Enterprise 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.

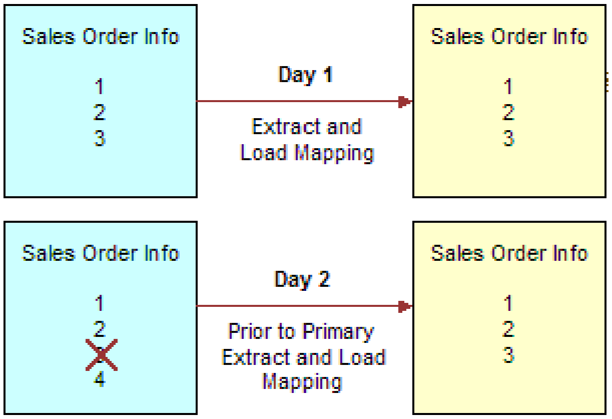


Figure 23. Changing Sales Order Information

Figure 24 shows the primary extract and delete process that occurs when day two's information is extracted and loaded into the Siebel Customer-Centric Enterprise Warehouse from the source. The initial extract brings record four into the Siebel Customer-Centric Enterprise 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 Siebel Customer-Centric Enterprise Warehouse table. It looks for records that exist in the Siebel Customer-Centric Enterprise 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 an eventual deletion in the Siebel Customer-Centric Enterprise Warehouse.

The extract mapping also looks for any new records that have been added to the source, and which do not already exist in the Siebel Customer-Centric Enterprise Warehouse; in this case, record four. Based on the information in the staging table, Sales Order number three is physically deleted from Siebel Customer-Centric Enterprise Warehouse, as shown in Figure 24. When the extract and load mappings run, the new sales order is added to the warehouse.

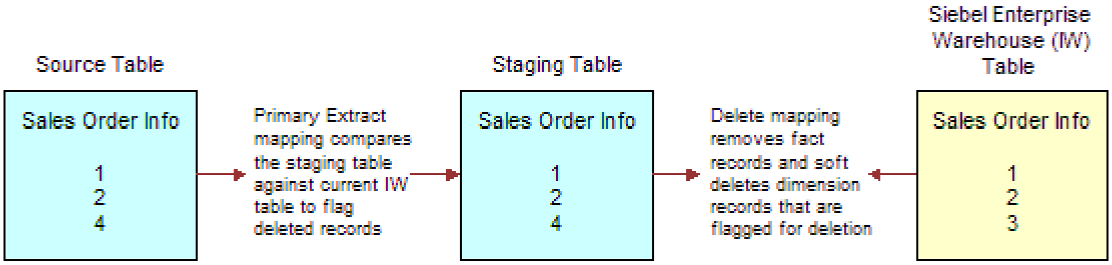

Source Table
Sales Order Info
1
2
4
Primary Extract mapping compares the staging table against current IW table to flag deleted records
Staging Table
Sales Order Info
1
2
4
Delete mapping removes fact records and soft deletes dimension records that are flagged for deletion
Siebel Enterprise Warehouse (IW) Table
Sales Order Info
1
2
3

Figure 24. Primary Extract and Delete Mappings

About Working with Primary Extract and Delete Mappings

The primary extract and delete mappings serve a critical role in identifying which records have been physically deleted from the source system. However, there are some instances when you can disable or remove the primary extract and delete mappings, such as when you want to retain records in the data warehouse that were removed from the source systems' database and archived in a separate database.

Because delete mappings use Source IDs and Key IDs to identify purged data, if you are using multiple source systems, you must modify the SQL Query statement to verify that the proper Source ID is used in the delete mapping. In addition to the primary extract and delete mappings, the configuration of the delete flag in the ADI also determines how record deletion is handled.

You can manage the extraction and deletion of data in the following ways:

- Deleting the configuration for source-archived records
- Deleting records from a particular source
- Enabling delete and primary-extract sessions
- Configuring the Record Deletion flag
- Configuring the Record Reject flag

This topic provides procedures for these management tasks.

Deleting the Configuration for Source-Archived Records

Some sources archive records in separate databases and retain only the current information in the main database. If you have enabled the delete mappings, you must reconfigure the delete mappings in the Siebel Customer-Centric Enterprise Warehouse to retain the archived data.

To retain source-archived records in the Siebel Customer-Centric Enterprise Warehouse, perform two tasks on each delete mapping:

1. Create a parameter for the archive date.
2. Edit the SQL Query field in the Source Qualifier.

   For a list of all delete sessions, see the discussion on disabling delete and primary extract sessions in About Working with Primary Extract and Delete Mappings on page 126.

To create a parameter for the archive date

1. In PowerCenter Designer, open the applicable source system configuration folder, and open a delete mapping.
2. Select Mappings > Parameters and Variables and select Add.
3. Enter `$$ARCHIVE_DK` as the name, and select Parameter for the type.
4. Select Date/Time for the data type, using the format that matches your source system.
5. Set the precision or scale required.
6. Double-click the Source Qualifier of your delete mapping.
7. Select the Variables tab, then select the parameter you just created, and click OK.

To add a clause to your Source Qualifier

1. In the delete mapping, open the Source Qualifier to edit the SQL Query field.
2. In the Properties tab, edit SQL Query field.

   Use the Archive Date as a filter in the `WHERE` clause.

   For example, if you were to create this statement for sales order lines, your clause would look like this:

   ```
   SELECT <column(s)> FROM OD_SALES_ORDLNS
   ```

   ```
   WHERE CREATED_ON_DK > $$ARCHIVE_DK AND NOT IN TO_SALES_ORDLNS_PE
   ```

3. Validate the SQL, and save your changes to the repository.

Deleting Records from a Particular Source

Delete mappings use Source IDs and Key IDs to identify the data that has been purged from the source system. Therefore, if you are using multiple source systems (where each type is identified by a Source ID), you must verify that the proper Source ID is used in the delete mapping so that only the desired records are deleted. To specify the correct source, edit the SQL statement in the delete mapping.

To specify the source for the delete mapping

1. In PowerCenter Designer, open the applicable source system configuration folder.

2. Open the delete mapping.
3. Open the Source Qualifier transformation to edit the SQL statement in the SQL Query field.

   Edit the SOURCE_ID expression in the OD table by adding a source abbreviation. A sample source abbreviation is shown in the following table.

Source	Source Abbreviation for SOURCE_ID
Oracle 11i	OAP11i

4. Validate the SQL statement and save your changes to the repository.

Enabling Delete and Primary Extract Sessions

If you want to remove your source-deleted records in the Siebel Customer-Centric Enterprise Warehouse, you need to enable the delete and primary extract sessions for your application.

To enable primary extract and delete sessions

1. In PowerCenter Workflow Manager, open the applicable source system configuration folder.
2. Double-click the session to open the Edit Tasks dialog box.
3. Edit the session, and then clear the Disable the task check box to enable the session.
4. Repeat these steps for each applicable primary extract and delete sessions.

Configuring the Record Deletion Flag

Record deletion is performed in the ADI if there are records you want to delete regardless of whether or not they have been deleted from the source. Record deletion is also performed in the ADI when the primary extract cannot determine if records have been deleted from the source, as in the case of Web logs where there are no primary keys for the extract to recognize. The Delete Flag determines how the record deletion is handled.

You can configure the Delete Flag in the Source Adapter mapplet by modifying the transformation for the EXT_DELETE_FLAG port. To reconfigure the handling of deletions, you can modify the Delete Flag definition. There are different values that you can use when defining your Delete Flag; these values depend on whether you are dealing with a fact table or a dimension table.

When you define the Delete Flag for fact tables, it is recommended you use a conditional statement. For example, you could enter the following statement:

```
IIF(<SOURCE>.<COLUMN_NAME> = ‘Y’, 'Y', 'N')
```

For fact tables, there are two values for which you can set the Delete Flag—Y and N. By setting your Delete Flag to Y, records that already exist in the data warehouse are purged from the fact table by the use of delete mappings. By setting your Delete Flag to N or any other value besides Y, your records are not deleted. If the record is marked as deleted by the source and has not yet been loaded into the data warehouse, then the record is not loaded.

When defining the Delete Flag for dimension tables, it is recommended that you use a conditional statement as well. For example, you could enter the following statement:

```
IIF(<SOURCE>.<SOME_COLUMN_NAME> = 'Y', 'D', 'N')
```

For dimension tables, there are two values for which you can set your Delete Flag—D and N. By setting your Delete Flag to D, your records are marked for deletion, but are not be purged from the dimension table just in case you want to query these values at a later time. If you wish to analyze historical dimension records, you must enable Type II functionality, which updates records by inserting a new record and leaving the old record intact. For more information about Type II dimensions, see Type I and Type II Slowly Changing Dimensions on page 131.

NOTE: If you set the Delete flag as P for a dimension record, the deletion logic behaves as if it was marked as D. No dimensions are ever purged from the data warehouse.

To configure the Delete Flag

1. In PowerCenter Designer, open the applicable source system configuration folder.
2. Open the applicable Source Adapter mapplet.
3. Double-click the Expression transformation to open the Edit Transformations box.
4. In the Ports tab, edit the expression for the EXT_DELETE_FLAG port.

   For example, if your source system sets the document type to DEL when a record is to be deleted, this expression contains a statement similar to the following:

   ```
   IIF (DOCUMENT_TYPE = ‘DEL’, ‘D’, ‘N’)
   ```

5. Validate and save your changes to the repository.

Configuring Record Reject Flag

The purpose of rejecting records is to verify that they are not loaded into the data warehouse. You can set up the rejection logic in one of two places—the Source Qualifier in the Business Component mapplet of an extract mapping or the Source Adapter mapplet of a load mapping.

By default, the Siebel Customer-Centric Enterprise Warehouse provides a reject flag in the Source Adapter mapplet that you can use to set up your record rejection logic. If the Reject Flag is set to Y for any records, the ADI skips those records and does not load it into the data warehouse. However, if the Reject Flag is set to N or any other value, the ADI processes the record. The reject logic must be configured in the Source Adapter mapplets according to your requirements.

You can configure the Reject Flag in the Source Adapter mapplet by modifying the transformation for the port EXT_REJECT_FLAG.

NOTE: If you want to set up the rejection logic in the Source Qualifier in the extract mapping, you can do so. The Siebel Customer-Centric Enterprise Warehouse performs this in the load mapping because some extract mappings load multiple staging tables in the data warehouse.

To configure the Reject flag

1. In PowerCenter Designer, open the applicable source system configuration folder.

2 Open the applicable Source Adapter mapplet.

3 Double-click the Expression transformation to open the Edit Transformations box.

4 In the Ports tab, edit the expression for the EXT_REJECT_FLAG port.

5 Validate and save your changes to the repository.

Configuring Slowly Changing Dimensions

The Siebel Customer-Centric Enterprise 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 Siebel Customer-Centric Enterprise Warehouse has an update, the updated information is posted into a new row and the old information is kept for historical reporting purposes.

The Siebel Customer-Centric Enterprise Warehouse identifies and applies the slowly changing dimension logic chosen by the user after data has been extracted and transformed to be source-independent, as shown in Figure 25. Users may configure the Source Adapter mapplet 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. Choosing Type I or Type II SCDs depends on identifying your historically significant attributes.

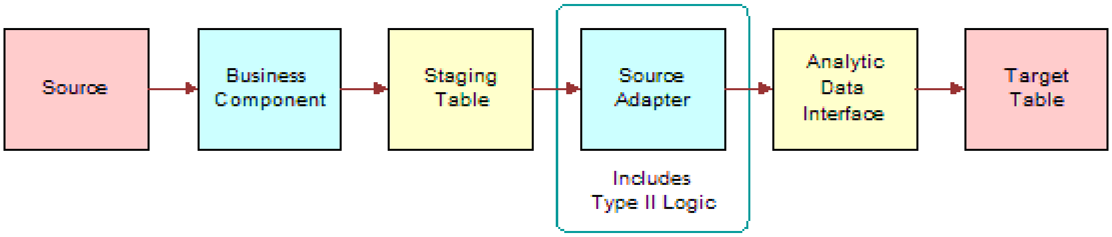


Figure 25. Slowly Changing Dimension Logic Flow

About Identifying Historically Significant Attributes

You may want to retain a history of all the updates to a particular dimension so that you can use them in reports. These dimensions are known as *historically significant* attributes. For example, if a customer moves to a different region and you assign that customer a new regional salesperson and territory ID, you may want to keep records of that customer's account history with the original salesperson and territory ID. In this case, the salesperson and territory IDs are *historically significant* attributes. In contrast, you may have a load that populates the telephone number field. If your business does not perform data analysis on phone number history, then this information may be considered a *historically insignificant* attribute.

Identifying attributes as significant or insignificant allows you to determine the 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 data warehouse. Records with changes that are important but not historically tracked are overwritten in the data warehouse, based on the primary key.

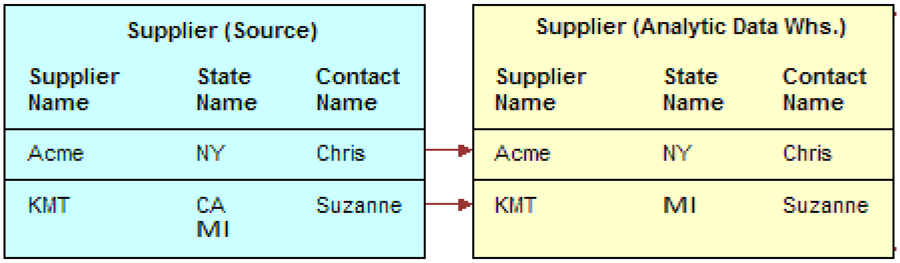
Type I and Type II Slowly Changing Dimensions

After you have correctly identified your significant and insignificant attributes, you can configure the Siebel Customer-Centric Enterprise 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 Siebel Customer-Centric Enterprise 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 may want to use Type I when changing incorrect values in a column.

In Figure 26, the State Name column for the supplier KMT is changed in the source table Suppliers, because it was incorrectly entered as California. When the data is loaded into the data warehouse table, no historical data is retained and the value is overwritten. If you look up supplier values for California, records for KMT do not appear; they only appear for Michigan, as they have from the beginning.



Supplier (Source)

Supplier Name	State Name	Contact Name
Acme	NY	Chris
KMT	CA MI	Suzanne

Supplier (Analytic Data Whs.)

Supplier Name	State Name	Contact Name
Acme	NY	Chris
KMT	MI	Suzanne

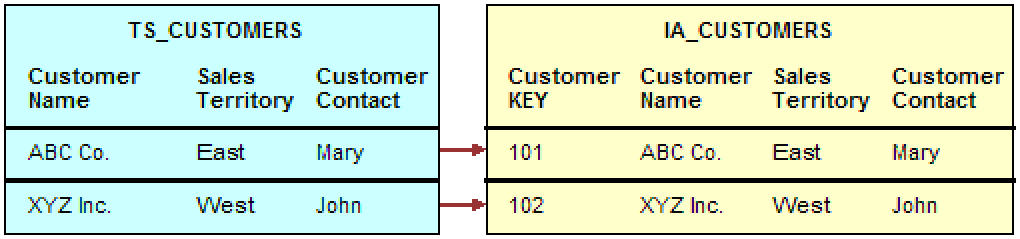
Figure 26. Type I Slowly Changing Dimension Example

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). In Figure 27, shows how an extract table (TS_CUSTOMERS) becomes a data warehouse dimension table (IA_CUSTOMERS). 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 TS_CUSTOMERS is loaded into the target table IA_CUSTOMERS, and each record is assigned a unique primary key (Customer Key).



TS_CUSTOMERS

Customer Name	Sales Territory	Customer Contact
ABC Co.	East	Mary
XYZ Inc.	West	John

IA_CUSTOMERS

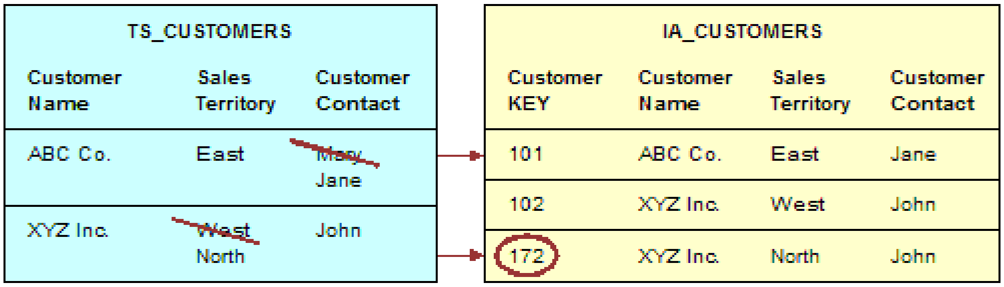
Customer KEY	Customer Name	Sales Territory	Customer Contact
101	ABC Co.	East	Mary
102	XYZ Inc.	West	John

Figure 27. Day One: The CUSTOMERS Extract and Data Warehouse Tables

However, this data is not static; the next time a data extract shows a change for your customers in IA_CUSTOMERS, the records must change. This situation occurs when slowly changing dimensions are invoked. Figure 27 shows that records for the two customers, ABC Co., and XYZ inc. have changed when compared with Figure 26. 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 Figure 28, instead of overwriting the Sales Territory column in the XYZ's record, a new record is added, assigning a new Customer Key, 172, to XYZ in IA_CUSTOMERS. 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 Customer Key 172 in the North sales territory.



TS_CUSTOMERS

Customer Name	Sales Territory	Customer Contact
ABC Co.	East	~~Mary~~ Jane
XYZ Inc.	~~West~~ North	John

IA_CUSTOMERS

Customer KEY	Customer Name	Sales Territory	Customer Contact
101	ABC Co.	East	Jane
102	XYZ Inc.	West	John
172	XYZ Inc.	North	John

Figure 28. Changes in the Customer Extract Table

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 Siebel Customer-Centric Enterprise 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 IA_CUSTOMERS table previously discussed, XYZ moved to a new sales territory.

If your source system supplied historical data on the location changes, your table may contain a record for XYZ in the West sales territory with an effective from date of January 1, 2001 and an effective to date of January 1, 3714. If the next year your source indicates XYZ has moved to the North sales territory, then a second record is inserted with an effective from date of January 1, 2002, and an effective to date of January 1, 3714, as shown in Table 18.

Table 18. Records Before a Wrapper Session

IA_CUSTOMER					
Customer Name	Sales Territory	Customer Contact	Effective From	Effective To	Current
ABC	East	Jane	1/1/2001	1/1/3714	Y

Table 18. Records Before a Wrapper Session

IA_CUSTOMER					
Customer Name	Sales Territory	Customer Contact	Effective From	Effective To	Current
XYZ	West	John	1/1/2001	1/1/3714	Y
XYZ	North	John	1/1/2002	1/1/3714	Y

Note your first record for XYZ still shows as effective from January 1, 2001 to January 1, 3714, while a second record has been added for XYZ in the North territory with the new effective from date of January 1, 2002. In this second record the effective to date remains the same, January 1, 3714.

When you schedule a wrapper session to execute, the effective dates for the first XYZ are corrected (January 1, 2001-January 1, 2002), and the Current Flag is adjusted in the Analytic Data Interface (ADI) so that only the second record (January 1, 2002-January 1, 3714) is set to Y. After the wrapper session completes its work, you have Type II information for XYZ in your data warehouse rather than two disparate records, as shown in Table 19.

Table 19. Records After a Wrapper Session

IA_CUSTOMER					
Customer Name	Sales Territory	Customer Contact	Effective From	Effective To	Current
ABC	East	Jane	1/1/2001	1/1/3714	Y
XYZ	West	John	1/1/2001	1/1/2002	N
XYZ	North	John	1/1/2002	1/1/3714	Y

In the previous paragraph, the wrapper session corrected the effective to dates and current flag. However, if the record's dates had been correct, the wrapper mapping would simply have set the current flag as needed, because its logic is set to check dates and flags and only adjust columns that contain discrepancies. Finally, if your source system does not supply any Type II information, you may disable the wrapper session completely; in this case all Type II work is handled by the Analytics Data Interface mapplet.

Enabling Type II Slowly Changing Dimensions (SCDs)

You can configure the Siebel Customer-Centric Enterprise Warehouse to maintain history down to the column level. For example, if you are loading the IA_ACTIVITY_COSTS table and you only want to maintain history for the AVG_ACTV_DURATION column, you can set the Type II flag to Y for this column and N for all other columns. A brief summary is provided in Table 20.

Table 20. Summary of Types of Slowly Changing Dimensions

Slowly Changing Dimensions Type	Type II Flag	Description
Type I Slowly Changing Dimension	N	Overwrites the data with the latest value.
Type II Slowly Changing Dimension	Y	Creates a new record for the updated records and, if applicable, updates the effective dates and current flag for any existing record.

If you want to use Type II SCDs, you need to set the value of the Type II Flag. By default it is set to N, but you can enter a conditional statement that sets the flag to Y. For example, you may only want to create new records if particular columns change values. In this case, you can set this up in your conditional statement.

You can configure the Type II Flag in the Source Adapter mapplet by modifying the port EXT_TYPE2_FLAG in the Expression transformation. Use the following procedure.

To configure the Type II flag

1. In PowerCenter Designer, open the applicable source system configuration folder.
2. Open the applicable Source Adapter mapplet.
3. Double-click the Expression transformation to open the Edit Transformations box.
4. In the Ports tab, edit the expression for the EXT_TYPE2_FLAG port.

   The default for the Type II Flag is N. Here you can enter Y to enable Type II functionality for all columns in the table. You can also enable only certain columns to maintain history. To enable some columns, but not others, you need to insert a conditional statement. For example, if you want to maintain history only for the Channel Point Name column, then you could write the following conditional statement:

   ```
   IIF(EXT_CHNL_POINT_NAME!= OD_CHNL_POINT_NAME, ‘Y’, ‘N’)
   ```

   In this case, only the Channel Point Name column has historical values—all other columns do not.

5. Validate and save your changes to the repository.

About Document, Local, and Group Currencies

Currency lookups are required because your business may have transactions involving multiple currencies. To create a meaningful report, you have to use a common currency. The Siebel Customer-Centric Enterprise Warehouse provides a means for converting a variety of currencies, as well as prepackaging the following three currency types for each amount stored in the Siebel Customer-Centric Enterprise Warehouse:

- **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.
- **Local currency.** The currency in which the financial books, including the transaction, are closed. For example, if your business organization is located in France and orders a part from a supplier in Britain, it may pay in British pounds, but it closes its books in French francs. In this case the local currency for the transaction is French francs and the document currency for the transaction is British pounds. The local currency is useful when each business unit of the enterprise creates its own internal reports. For example, your Japanese site may produce internal reports using Japanese yen, while your United States site may produce internal reports using United States dollars.
- **Group currency.** The standard currency used by your entire enterprise. For example, if a multinational enterprise has its headquarters in the United States, its group currency is probably U.S. dollars. The group currency is useful when creating enterprise-wide reports.

For every monetary amount extracted from the source, the ADI loads the document, local and group currency amounts into the target table. The method that the ADI uses to load the three different currency values depends on what the source provides.

There are three possible situations that can occur:

- Source System Provides Three Currency Amounts on page 137

  In this situation, the source system provides all three currency amounts. All three amounts are extracted and loaded into the corresponding Siebel Customer-Centric Enterprise Warehouse table; the system does not need to do any currency conversions.

- Source System Provides Document Amount, Codes, and Exchange Rates for Local and Group Currencies on page 137

  In this situation, the source system provides the document currency amount and the exchange rates for finding the local and group currency amounts.

- Source System Provides the Document Currency Amount on page 138

  In this situation, the source system provides the document currency amount, but it does not provide the local and group currency amounts or exchange rates used for currency conversion.

The Siebel Customer-Centric Enterprise Warehouse has a predefined logic for loading the document, local, and group currency values into an IA table. The logic is shown in Figure 29.

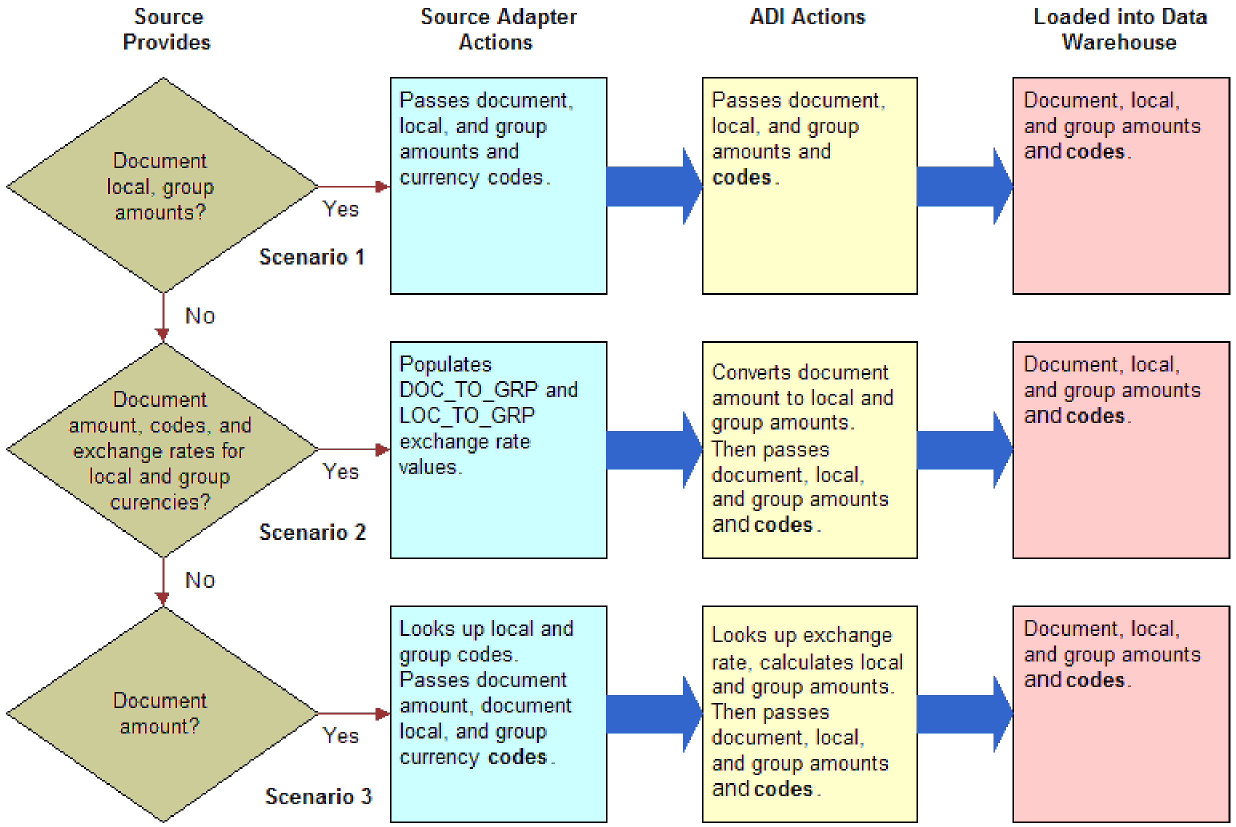

Source Provides
Source Adapter Actions
ADI Actions
Loaded into Data Warehouse
Document local, group amounts?
Yes
Scenario 1
Passes document, local, and group amounts and currency codes.
Passes document, local, and group amounts and codes.
Document, local, and group amounts and codes.
No
Document amount, codes, and exchange rates for local and group curencies?
Yes
Scenario 2
Populates DOC_TO_GRP and LOC_TO_GRP exchange rate values.
Converts document amount to local and group amounts. Then passes document, local, and group amounts and codes.
Document, local, and group amounts and codes.
No
Document amount?
Yes
Scenario 3
Looks up local and group codes. Passes document amount, document local, and group currency codes.
Looks up exchange rate, calculates local and group amounts. Then passes document, local, and group amounts and codes.
Document, local, and group amounts and codes.

Figure 29. Currency Conversion Logic

Source System Provides Three Currency Amounts

If the source system provides the amounts for the three currencies—document, local, and group. Each of these amounts is loaded into the data warehouse as provided.

Source System Provides Document Amount, Codes, and Exchange Rates for Local and Group Currencies

The source system provides the following:

- Amount in document currency
- Local currency and group currency codes
- Exchange rates for local and group currencies (DOCUMENT_TO_GROUP and LOCAL_TO_GROUP exchange rates).

All of this information is fed to the ADI. The ADI uses the DOCUMENT_TO_GROUP rate to convert the document currency amount to the group currency amount and uses the ratio of DOCUMENT_TO_GROUP and LOCAL_TO_GROUP to convert the document currency amount to the local currency amount. The ADI then passes the three currency amounts and three currency codes to the corresponding IA table.

Source System Provides the Document Currency Amount

This is the most common situation, and thus is the Siebel Customer-Centric Enterprise Warehouse's default for handling currency. If the source system provides only the document currency amount, the Source Adapter performs lookups to identify the local and group currency codes based on the source system. Based on the source system the appropriate currencies are assigned. After the lookups occur, the Source Adapter mapplet provides the ADI with the document currency amount and the document, local, and group currency codes.

Because the ADI does not have the local and group amounts, or document-to-group (DOC_TO_GRP) and local-to-group (LOC_TO_GRP) currency conversion rates to derive the amounts, it looks up the exchange rates. By default, the ADI looks up the exchange rates in the IA_XRATES table. However, if you have custom tables that maintain exchange rate values, you can reconfigure the system to do the extraction from that source system.

The ADI then populates the exchange rate values in the DOC_TO_GRP and LOC_TO_GRP variables. After this population is complete, the ADI uses the DOC_TO_GRP rate to convert the document currency amount to the group currency amount. The ADI then uses the ratio of DOC_TO_GRP and LOC_TO_GRP to convert the document currency amount to the local currency amount. Finally, the ADI passes the three currency amounts and currency codes to the corresponding IA table.

NOTE: For currency code information to be resolved, you must verify that the currency codes extracted map to the currency codes in the IA_XRATES table.

Process for Configuring Currencies

Depending on how your organization handles currency, some of the following currency configuration points may apply to you. For example, you may find it necessary to reconfigure the group currency type, or you may need to reconfigure the way exchange rates are handled. The following sections provide procedures for configuration points related to these currency types.

To configure currencies, perform the following tasks:

- Supplying Document-to-Group and Local-to-Group Currency Exchange Rates on page 139
- Extracting Exchange Rates from a Custom Table on page 139
- Configuring the Exchange Rate Type on page 140
- Configuring Currency Code for Oracle 11i on page 141

Supplying Document-to-Group and Local-to-Group Currency Exchange Rates

This task is a step in the Process for Configuring Currencies on page 138.

As mentioned in the previous section, the Siebel Customer-Centric Enterprise Warehouse can present all amount values in three different currencies—document, local, and group. If all three amounts are not supplied by the source system, then the ADI can calculate the amounts using prepackaged exchange rate logic.

Usually the EXT_DOC_TO_GRP and EXT_LOC_TO_GRP values are null at the input of the ADI. Therefore, the ADI has to perform a lookup to retrieve the exchange rates. However, if you want to supply an exchange rate directly, you can do so by using the appropriate column. The EXT_XRATE_DOC_TO_GRP column supplies the exchange rate that converts the document currency to group currency, and the EXT_XRATE_LOC_TO_GRP column supplies the exchange rate that converts the local currency amount to the group currency amount. In this topic, you can find procedures for providing exchange rates.

To supply exchange rate values directly

1. In PowerCenter Designer, open the applicable source system configuration folder.
2. In the MPLT_SA[Source Abbreviation]_XRATES Source Adapter mapplet, edit the expression for the applicable port.

   You can use the EXT_DOC_TO_GRP column to convert the document currency to the group currency. You can use the EXT_LOC_TO_GRP column to convert the local currency to the group currency.
3. Validate and save your changes to the repository.

Extracting Exchange Rates from a Custom Table

This task is a step in the Process for Configuring Currencies on page 138.

If you do not maintain exchange rates in your ERP tables, but instead, you maintain them in your custom tables, then you can load them into the IA_XRATES table of the Siebel Customer-Centric Enterprise Warehouse by creating a new mapping. The easiest way to create the mapping is to copy an existing exchange rate extraction and load mapping. After you copy it, you can modify the mapping to work with your custom tables.

To create new exchange rate extraction and load mappings

1. In PowerCenter Designer, open the applicable source system configuration folder.
2. Locate an existing exchange rate extract mapping.

   Copy the entire exchange rate mapping. Be sure to copy the entire set of objects—Business Components, and staging tables.

3 Reconfigure the Business Component to select the columns from your custom tables, instead of selecting them from the prepackaged tables.

  You can name and save this modified version of the Business Component as MPLT_BC1_XRATES.

4 Reconfigure the staging table to hold all the source system related columns.

  You can name and save this staging table as T1_XRATES.

5 Reconfigure the extract mapping that uses the MPLT_BC1_XRATES as the source system, and T1_XRATES as the target, and form the KEY_ID in the extract mapping.

  This mapping is the unique record identifier from the source system. However, it may be a combination of multiple columns from the source system.

6 Locate an existing exchange rate load mapping and copy the entire exchange rate mapping.

  Be sure to copy the entire set of objects—Business Components, and staging tables.

7 Reconfigure the Source Adapter MPLT_SA1_XRATES mapplet to map the data in the staging table to the ADI format.

8 Reconfigure the load mapping M_1_XRATES_LOAD with the new Source Adapter mapplet.

9 Validate any expressions or SQL statements, and then save the changes to your repository.

Configuring the Exchange Rate Type

This task is a step in the Process for Configuring Currencies on page 138.

The column XRATE_TYPE_CODE in the table IA_XRATES identifies a specific type of exchange rate. Data loaded from various sources have different types of exchange rates; thus, there can be different values for the XRATE_TYPE_CODE column.

If an exchange rate is not supplied to convert an amount to a different currency, then the ADI performs a lookup to retrieve an exchange rate from the IA_XRATES table. The lookup that is retrieved is based on the specified exchange rate type. Each load mapping has the default exchange rate type M. However, depending on the source system, the Siebel Customer-Centric Enterprise Warehouse has a prepackaged SQL statement in the session to extract a default exchange rate type that is applicable to the corresponding source. Thus, for Oracle 11i load sessions, there is a SQL statement that requests the exchange rate type Corporate instead of M.

If you would like to use an exchange rate type other than the prepackaged statement, then you must edit the SQL statement in the applicable session.

To override the default exchange rate type

1 In PowerCenter Workflow Manager, open the source system configuration folder.

2 Open the applicable session to open the Edit Tasks box.

3 In the Transformations tab, edit the SQL statement for the exchange rate lookups.

4 Change the exchange rate type by editing the following SQL statement fragment:

```
    WHERE XRATE_TYPE = ‘Corporate’
```

For a list of exchange rate type codes, see your source system’s options.

5 Click OK.

NOTE: You need to perform these steps for each fact load for which you wish to configure the exchange rate information.

Configuring Currency Code for Oracle 11i

This task is a step in the Process for Configuring Currencies on page 138.

Oracle 11i provides document and local currencies, but not a group currency. As a result, you need to supply the currency code by means of a text file. For more information, see the discussion on configuring the group currency code in Chapter 9, “Configuring Siebel Customer-Centric Enterprise Warehouse for Oracle 11i.”

Handling European Monetary Union (EMU) Currencies

For transactions that take place after the Euro effective date (January 01, 1999), the published exchange rates involving *individual* EMU member currencies may not be available. To address the need for conversions between the individual currency of an EMU member and any other currency, you may need to set up a mechanism for retrieving these rates. The Siebel Customer-Centric Enterprise Warehouse prepackages two methods for handling European Monetary Union currencies:

- Flat file (`file_xrates_emu.csv`)
- Expression transformation (`EXP_CURR_CONVERSION_TRANSFORM`).

Each of the two methods is discussed in the following sections.

Using the Flat File to Supply EMU Exchange Rates

The flat file contains currency conversion rates for all combinations of EMU currencies (including individual EMU member currencies as well as the joint EMU currency, the Euro). This file includes the following combinations:

- **EMU to EMU.** This is the conversion from the individual currency of one EMU member country to another.
- **EUR to EMU.** This is the conversion from the joint EMU currency, the Euro, to the individual currency of an EMU member.
- **EMU to EUR.** This is the conversion from the individual currency of an EMU member to the joint EMU currency, based on the Euro.

If you use this method, the Siebel Customer-Centric Enterprise Warehouse assumes that the source system supplies currency conversion rates between individual EMU member currencies and national currencies for countries outside of the EMU. In addition, this method does not conform to the rounding rules set by the European Monetary Union.

To use this method, you must upload the exchange rates from this flat file to the IA_XRATES table. After the exchanges rates are available in the IA_XRATES table, the ADI can look up these exchange rates from the IA_XRATES table and convert currency as required.

To load the individual EMU member currency exchange rates using the flat file

1. Before you load the exchange rates from the flat file, if necessary, change the default values in the flat file for the parameters SOURCE_ID, XRATE_TYPE_CODE, XRATES_TYPE_DESC, and KEY_ID.

    - The SOURCE_ID default is OAP11. If your source is not Oracle, then you can reconfigure this ID.
    - The XRATE_TYPE_CODE and XRATE_TYPE_DESC default is Corporate.
    - The KEY_ID uses the XRATE_TYPE_CODE as part of it definition. The Key ID is defined as the concatenation of the From Currency Code, To Currency Code, Exchange Rate Type Code, and the Effective From Date. Therefore, if you change the Exchange Rate Type Code, then you must also modify the Key ID accordingly.

2. Create the session for the M_F_XRATES_EXTRACT mapping, using the file_xrates_emu.csv flat file as the source system file.
3. Run the extract session to load the flat file exchange rates into the TF_XRATES staging table.
4. Create a session for the M_F_XRATES_LOAD mapping.
5. Run the load session to load the exchange rates into the IA_XRATES table.

About the EXP_CURR_CONVERSION_TRANSFORM Expression Transformation to Supply EMU Exchange Rates

The Expression transformation contains currency conversion logic for converting any combination of EMU currency (including individual EMU member currencies as well as the Euro) and other currencies (OTH) outside of the EMU. This logic applies to the following combinations:

- **EMU to EMU.** This is the conversion from the individual currency of one EMU member country to another.
- **EUR to EMU.** This is the conversion from the joint EMU currency, the Euro, to the individual currency of an EMU member.
- **EMU to EUR.** This is the conversion from the individual currency of an EMU member to the joint EMU currency, the Euro.
- **OTH to EUR.** This is the conversion from any other currency outside the EMU to the Euro.
- **EUR to OTH.** This is the conversion from the Euro to any other currency outside the EMU.

- **EMU to OTH.** This is the conversion from the individual currency of an EMU member country to another currency outside the EMU.
- **OTH to EMU.** This is the conversion from any other currency outside the EMU to the individual currency of an EMU member.

Unlike the flat file method, the transformation handles conversions between EMU currencies and currencies for countries outside the EMU. In addition, this method conforms to the rounding rules set by the European Monetary Union.

The transformation performs the conversion in different ways, depending on the types of input and output currencies. In the transformation, there are ten sets of amount input fields; each set consists of a document amount, local amount, and group amount field. In this discussion, document currency is referred to as the *From* currency and group and local currencies as the *To* currencies. Source-supplied document amounts are output as the same value. However, if the local and group amounts are not supplied, then a particular currency conversion process occurs. There are six different conversion processes that could occur. They are described in the list that follows:

- **Non-EMU to Non-EMU.** If the From Currency and To Currency are not EMU currencies, the following logic is used:
  - The supplied exchange rate is used to compute the new amount.
  - If the exchange rate is not supplied as an input to the transformation, the exchange rate is retrieved using a lookup to the `IA_XRATES` table. The new amount is then calculated using the looked up exchange rate and the From Currency amount.
- **EMU to EMU.** If the From Currency and To Currency are EMU currencies, the following logic is used:
  - Euro-triangulation is used to derive the new amount. Please note that the Euro-triangulation logic only applies if the exchange rate date is later than or equal to the Euro effective date.
  - If the exchange rate date is earlier than the Euro effective date, this case is treated as a Non EMU to Non EMU currency conversion case.

  The Euro-triangulation logic is used to convert one EMU currency to another EMU currency (EMU to EMU). Again, this logic is only applicable if the exchange rate date is equal to or later than the Euro effective date. The Euro-triangulation logic is as follows:

  - Convert from one national denomination (`EMU_DOC`) to its Euro equivalent (`EUR`). For example, `EUR = EMU_DOC/(EUR-TO-EMU_DOC` conversion rate).
  - Round the previous step's result to the nearest three decimal points. The default rounding is to three decimal points, but this is configurable.
  - Convert the Euro equivalent into the resulting national denomination (`EMU_LOC`). For example:

    ```
    EMU_LOC=(Result from previous step) * (EUR-TO-EMU_LOC conversion rate)
    ```
- **EMU to OTH.** If the From Currency is EMU and To Currency is OTH, the following logic is used in the given order:
  - The supplied exchange rate is used to compute the amount.

  - If the exchange rate is not supplied and the NONEMU_EMU_TRI_FLAG = Y, then the two-step conversion method is used to derive the amount. The two-step conversion method is defined as follows:
    - First, the From Currency amount is converted to the Euro Currency amount using the fixed EMU conversion rates supplied by the transformation’s logic.
    - Second, the Euro Currency amount is converted to the To Currency amount using the appropriate exchange rate.

  NOTE: The two-step conversion method is only used if the exchange rate date is greater than or equal to the Euro effective date and the NONEMU_EMU_TRI_FLAG is set to Y.

  - If the NONEMU_EMU_TRI_FLAG is set to N, this is treated as a Non EMU to Non EMU currency conversion case.

- **OTH to EMU.** If the From Currency is OTH and the To Currency is EMU, then the following logic is used:
  - The supplied exchange rate is used to compute the To Currency amount.
  - If the exchange rate is not supplied and the NONEMU_EMU_TRI_FLAG = Y, then the two-step conversion method is used to derive the amount. The two-step conversion consists of the following two steps:
    - First, the From Currency amount (OTH) is converted to the Euro Currency amount using the exchange rate from the IA_XRATES table.
    - Second, the Euro Currency amount is converted to the EMU Currency amount using the fixed Euro conversion rate available in the transformation. Please note that the two-step conversion method only applies if the NONEMU_EMU_TRI_FLAG = Y.
  - If the NONEMU_EMU_TRI_FLAG = N, this is treated as a Non EMU to Non EMU currency conversion case.
- **EUR to EMU.** If the From Currency is EUR and the To Currency is EMU, the following logic is used:
  - If the exchange rate date is greater than or equal to the Euro effective date, then the From Currency amount is converted to the To Currency amount using the fixed Euro conversion rates supplied by the transformation’s logic.
  - In all other cases, this is treated as a Non-EMU to Non-EMU currency conversion case.
- **EMU to EUR.** If the From Currency is EMU and the To Currency is EUR, the following logic is used:
  - If the exchange rate date is greater than or equal to the Euro effective date, the From Currency amount is converted to the To Currency amount using the fixed Euro conversion rates supplied by the transformation’s logic.
  - If the exchange rate date is less than the Euro effective date, this is treated as a Non EMU to Non EMU currency conversion case.

To use an Expression transformation, you must add the transformation to every load mapping that requires any of the previously listed types of currency conversions. See the M_F_EURO_TRIANG_EXP_USAGE_EXAMPLE mapping for an example of how to incorporate an Expression transformation. This mapping is located in the Configuration for Universal Source folder.

To add the transformation to a load mapping

1 Open the applicable source system configuration folder.

2 Select Tools > Transformation Developer.

3 Open the Siebel Business Analytics folder and drag and drop the EXP_CURR_CONVERSION_TRANSFORM Expression transformation to create a shortcut.

   You can place the shortcut in the Transformations folder contained in the applicable source system configuration folder.

4 Select Mapplet Designer.

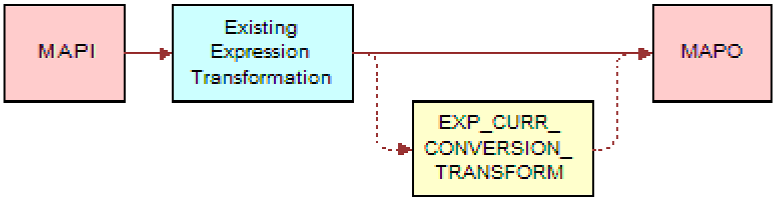
5 Open the applicable Source Adapter mapplet.

6 Add LKP_XRATES from the Transformations folder.

   NOTE: LKP_XRATES is a lookup to the IA_XRATES table. If the Transformations folder does not have a lookup to IA_XRATES, you must make a shortcut to LKP_XRATE in the Siebel Analytics Enterprise Applications folder.

7 Add the shortcut to the EXP_CURR_CONVERSION_TRANSFORM Expression transformation to the Source Adapter mapplet.

   Place the transformation between the load mapping's existing Expression transformation and MAPO, as shown in the following figure.


MAPI
Existing Expression Transformation
MAPO
EXP_CURR_ CONVERSION_ TRANSFORM

8 Reconnect the ports as necessary so that the data flows through the transformation and into the MAPO.

Table 21 lists all of the prepackaged exchange rates used by the transformation. Table 21 only lists exchange rates from EMU currency to Euro. Using these exchange rates, the transform can calculate any of the five scenarios (EMU to EMU, EUR to EMU, EMU to EUR, EMU to OTH, and OTH to EMU). For EMU to OTH, OTH to EMU, and EMU to EMU, the transformation converts the From Currency to the Euro and then from the Euro to the To Currency. For that reason, Table 21 only has EMU currency to the Euro conversion rates.

Table 21. Prepackaged Exchange Rates in Expression Transformation

Currency Code	Currency Name	Effective From Date	Conversion Rate for One Euro
BEF	Belgian Francs	01/01/1999	40.3399
DEM	German Marks	01/01/1999	1.95583
ESP	Spanish Pesetas	01/01/1999	166.386

Table 21. Prepackaged Exchange Rates in Expression Transformation

Currency Code	Currency Name	Effective From Date	Conversion Rate for One Euro
FRF	French Francs	01/01/1999	6.55957
IEP	Irish Púnt	01/01/1999	.787564
ITL	Italian Lira	01/01/1999	1936.27
LUF	Luxembourg Francs	01/01/1999	40.3399
NLG	Dutch Guilders	01/01/1999	2.20371
ATS	Austrian Schillings	01/01/1999	13.7603
PTE	Portuguese Escudo	01/01/1999	200.482
FIM	Finnish Markka	01/01/1999	5.94573

You can incorporate additional EMU currency conversion rates into the transformation. When doing so, you must consider whether or not the conversion rate is effective before January 1, 1999 or on/after January 1, 1999. (January 1, 1999 is the date the Euro was enacted.)

To add a new EMU currency with the same effectivity date (January 1, 1999)

- Edit the decode statement for the columns EMU_TO_EURO_CONV_FACT_DOC, EMU_TO_EURO_CONV_FACT_LOC, and EMU_TO_EURO_CONV_FACT_GRP, to add the new currency and its conversion rate.

  For example, if you are adding EMU1, where the Euro-to-EMU1 conversion rate is 2.34567, then you would modify the EMU_TO_EURO_CONV_FACT_DOC column definition as follows:

  ```
  EMU_TO_EURO_CONV_FACT_DOC = IIF (ISNULL (EXT_XRATE_LKP_DT) OR
  TO_NUMBER(TO_CHAR(EXT_XRATE_LKP_DT,'J')) >= TO_NUMBER(TO_CHAR(TO_DATE('01/01/
  1999', 'MM/DD/YYYY'),'J')), DECODE(EXT_DOC_CURR_CODE, 'BEF', 40.3399000000,
  'NLG', 2.2037100000, 'ATS', 13.7603000000, 'PTE', 200.4820000000, 'FIM',
  5.9457300000, 'IEP', 0.7875640000, 'LUF', 40.3399000000, 'FRF', 6.5595700000,
  'ITL',1936.2700000000, 'DEM', 1.9558300000, 'ESP', 166.3860000000,
  ‘EMU1’,2.3456700000, 'EUR', 1.0000000000, NULL), NULL)
  ```

To add a new EMU currency with a different effectivity date than January 1, 1999

- Edit the decode statement for the columns EMU_TO_EURO_CONV_FACT_DOC, EMU_TO_EURO_CONV_FACT_LOC, and EMU_TO_EURO_CONV_FACT_GRP, to add the new currency, its conversion rate, and the effectivity date logic.

  For example, if you are adding EMU2, where the Euro-to-EMU2 conversion rate is 3.45678, and the exchange rate effectivity date is 01/01/2001, then you would modify the EMU_TO_EURO_CONV_FACT_DOC column definition as follows:

  ```
  EMU_TO_EURO_CONV_FACT_DOC = IIF (ISNULL (EXT_XRATE_LKP_DT)
  OR(TO_NUMBER(TO_CHAR(EXT_XRATE_LKP_DT,'J')) >= TO_NUMBER(TO_CHAR(TO_DATE('01/01/
  1999', 'MM/DD/YYYY'),'J')) AND EXT_DOC_CURR_CODE!= 'EMU2') OR
  ```

```
TO_NUMBER(TO_CHAR(EXT_XRATE_LKP_DT,'J')) >= TO_NUMBER(TO_CHAR(TO_DATE('01/01/
2001', 'MM/DD/YYYY'),'J')),DECODE(EXT_DOC_CURR_CODE, 'BEF', 40.3399000000,
'NLG', 2.2037100000, 'ATS', 13.7603000000, 'PTE',200.4820000000, 'FIM',
5.9457300000, 'IEP', 0.7875640000, 'LUF', 40.3399000000, 'FRF', 6.5595700000,
'ITL',1936.2700000000, 'DEM', 1.9558300000, 'ESP',166.3860000000, ‘EMU2’,
3.4567800000, 'EUR', 1.0000000000, NULL), NULL)
```

About Stored Lookups

A lookup transformation allows you to specify a reference table, and then retrieve information such as code descriptions, exchange rates, and currency codes. The main types of preconfigured lookups within the Siebel Customer-Centric Enterprise Warehouse are:

- Codes Lookup on page 147.
- Dimension Key Lookups. For more information, see About Resolving Dimension Keys on page 149.

Codes Lookup

Some source systems use intelligent codes that are intuitively descriptive, such as HD for hard disks, while other systems use nonintelligent 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 Siebel Customer-Centric Enterprise Warehouse.

The code lookup in the ADI integrates both intelligent and nonintelligent codes by performing a separate extract for codes, and inserting the codes and their description into a codes table. The codes table provides the ADI with a resource from which it can automatically perform a lookup for code descriptions.

The Analytic Data Interface’s architecture uses components, as well as both fact and dimension tables, to facilitate lookup functionality. The following components and process are used in a lookup:

IA_CODES Table

The load control table IA_CODES consolidates all codes for future reference and assigns them a category and a single language for efficient lookup capability.

Codes Mappings

The Siebel Customer-Centric Enterprise Warehouse uses mappings designed to extract codes from source systems and populate the IA_CODES table in preparation for use by the ADI.

To understand how codes mappings function, it is helpful to first understand the columns within IA_CODES. Table 22 describes these columns.

Table 22. Columns in Code Mapplet

Column	Description
SOURCE_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 M_[SOURCE]_CODES_[CATEGORY]. Figure 30 shows an example of a code mapping in PowerCenter Mapping Designer.

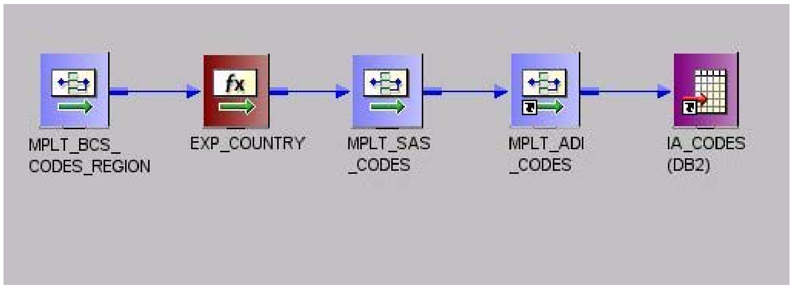


Figure 30. Sample Codes Mapping: M_S_CODES_REGION

Codes Mapplets

There are several mapplets that support the codes mappings in preparation for the source-independent ADI. They are as follows:

- **Source Adapter mapplets.** The Source Adapter mapplet connects the source-specific input attributes of CODES and the attributes from control or warehouse tables to the expression transform for mapping them. The naming convention for the Source Adapter codes mapplet is MPLT_SA[Source Abbreviation]_CODES.
- **Business Component mapplets.** The Business Component mapplet makes the source system attributes of CODES_CUST_CLASS available to the extract mapping. The naming convention for the Business Component codes mapplet is MPLT_BC[Source Abbreviation]_CODES_[CATEGORY].
- **ADI Mapplet.** The Analytic Data Interface (ADI) mapplet is source system independent and resolves the codes for the target table. The naming convention for the ADI codes mapplet is MPLT_ADI_CODES.

The ADI integrates multiple source system codes by designating one source system instance as a master in a mapping. All other source system codes are then mapped to the master. When the ADI encounters a code that requires definition, it references the load control lookup table to match the source system code to a Siebel Customer-Centric Enterprise Warehouse source-independent code, which retains all the source system codes' original functionality.

The following columns are used to designate a source system instance as the master source system:

- **`MASTER_ID`.** Code for the source system designated as master.
- **`SOURCE_ID`.** Unique identifier for the source system.

Configuring Extension Column Code Description Lookups

You can configure dimension and fact load sessions to perform specific lookups by editing the category of the data to be extracted from the `IA_CODES` table and loading the code information into a target table. If the code and code name do not exist in the `IA_CODES` table, then you must add them to the table. To configure the lookup, create a session override; do not modify the ADI in the load mapping.

To configure sessions for lookups

1. In PowerCenter Workflow Manager, open the applicable source system configuration folder.
2. Open the Edit Tasks box.
3. In the Transformations tab, edit the SQL statement for the lookup.

   For example, you may wish to edit the following lookup:

   ```
   MPLT_ADI_SUPPLIERS.LKP_SPLR_ATTR1
   ```

4. Edit the SQL statement to use the desired code category.
5. Edit the SQL statement from '`GENERIC`' to the category you wish to use for the lookup.

About Resolving Dimension Keys

By default, dimension key resolution is performed by the Siebel Customer-Centric Enterprise Warehouse in the ADI. The ADI uses prepackaged, reusable lookup transformations to provide prepackaged dimension key resolution. This section describes how dimension keys are looked up and resolved.

There are two commonly used methods for resolving dimension keys. The first method, which is the primary method used, is to perform a lookup for the dimension key. The second method is to supply the dimension key directly into the fact load mapping.

Resolving the Dimension Key Using Lookup

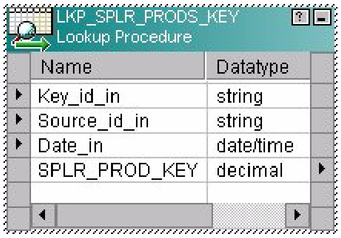
If the dimension key is not provided to the ADI through database joins, the ADI performs the lookup in the dimension table. The ADI does this using prepackaged lookup transformations.

The ADI uses the Dimension Key ID, the Source ID and Lookup date in looking up the dimension key. All these columns are necessary for the ADI to return the dimension key. The ports are described in Table 23.

Table 23. Columns Used in the ADI Dimension Key Lookup

Port	Description
Key ID	Uniquely identifies the dimension entity within its source system. Formed from the transaction in the Source Adapter of the fact table.
Source ID	Unique identifier of the source system instance.
Lookup Date	The primary date of the transaction; for example, receipt date, sales date, and so on.

In Figure 31, the Supplier Products Key Lookup transformation illustrates the three input columns needed for the ADI lookup—the Key ID, Source ID, and Date (lookup date). The transformation then outputs the Supplier Product key (the dimension key) to the data warehouse table IA_SPLR_PRODS.

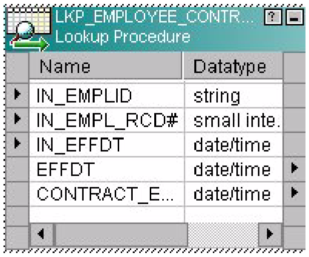


Name	Datatype
Key_id_in	string
Source_id_in	string
Date_in	date/time
SPLR_PROD_KEY	decimal

Figure 31. Example of Input Columns Required for the ADI

If Type II slowly changing dimensions are enabled, the ADI 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 lookup for Employee Contract Data shown in Figure 32, you can see the effective dates used to provide the effective period of employee contracts.



LKP_EMPLOYEE_CONTR...
Lookup Procedure

Name	Datatype
IN_EMPLID	string
IN_EMPL_RCD#	small inte.
IN_EFFDT	date/time
EFFDT	date/time
CONTRACT_E...	date/time

Figure 32. Effective Dates Used in Slowly Changing Dimension

Configuring the Dimension Key ID

There are many ways in which you can configure Dimension Key IDs. This section provides procedures on configuring dimension keys so that your dimension tables load data at an appropriate granular level and, also so that your keys get resolved.

Each Dimension Key ID has a default value, which you can configure. If you want to reset the value of a Dimension Key ID, you must modify the Key ID definition in the dimension extract mapping, in the Expression transformation, as well as in every fact load that uses the key. For example, if you want to modify the Key ID for the IA_GL_ACCOUNTS dimension table, then you must modify the Key ID's definition in the IA_GL_ACCOUNTS extract mapping's Expression transformation.

In addition, you have to modify any fact table load mapping that uses the key. For example, because the IA_SALES_IVCLNS fact table uses the Key ID of the IA_GL_ACCOUNTS dimension table, you must modify the Key ID definition in the IA_SALES_IVCLNS load mapping's Source Adapter mapplet. The following two procedures tell you how to accomplish both of these tasks.

To configure the Key ID in the dimension extract mapping

1. In PowerCenter Designer, open the applicable source system configuration folder.
2. Open the applicable extract mapping.
3. Double-click the Expression transformation to open the Edit Transformations box.
4. Edit the expression for the KEY_ID column.

   For example, if you redefine the KEY_ID column for the IA_GL_ACCOUNTS table, modify the default Key ID port in the M_I_GL_ACCOUNTS_EXTRACT mapping, which, by default, is set to the Set-of-Books ID ~ Code Combination ID (TO_CHAR(SOB_ID)||'~'||TO_CHAR(CC_ID)). You can reset the grain of this dimension table by setting the Key ID to something else, like Set-of-Books ID ~ GL account number.

   NOTE: Verify that any modified Key ID continues to uniquely identify each record in the table.

5. Validate and save your changes to the repository.

To configure the Key ID in the fact load mapping

1. In PowerCenter Designer, open the applicable source system configuration folder.
2. Open the applicable fact mapping's Source Adapter mapplet.
3. Double-click the Expression transformation to open the Edit Transformations box.
4. Edit the expression for the *_ID column.

   For example, if you want to redefine the IA_GL_ACCOUNTS Key ID in the IA_SALES_IVCLNS table, modify the default Key ID as shown in the following:

   ```
   TO_CHAR(INP_SETS_OF_BOOK_ID)||'~'||

   TO_CHAR(INP_CODE_COMBINATION_ID)
   ```

5. Validate and save your changes to the repository.
6. Repeat these steps for every fact table that is joined to the dimension in question.

Configuring Lookup in Additional or Custom Tables

The predefined extension dimension keys in the fact tables point to the IA_DIMENSION table. If you have custom dimension tables to add, or want to join additional dimension tables to a Siebel Customer-Centric Enterprise Warehouse fact table, you can link the dimension tables to the fact tables by modifying the SQL statement for the extension dimension keys in the fact tables.

Each fact table has at least three extension dimension keys, allowing you to store additional dimension tables. To join a new dimension table to a fact table, you need to modify the fact table's load mapping or session, which involves two tasks:

- Defining the Key ID in the Source Adapter mapplet of the fact table load.
- Modifying the session to perform a SQL statement for the lookup that is used to resolve the Key ID and redirect the lookup to the dimension table of your choice.

To reset the Dimension Key ID in the Source Adapter mapplet

1. In PowerCenter Designer, open the applicable source system configuration folder.
2. Open the applicable fact load mapping's Source Adapter mapplet.
3. Double-click the Expression transformation to open the Edit Transformations box.
4. Edit the EXT_[SUBJECT]_DIM*X*_KEY port.

   For example, if you want to join the IA_GL_ACCOUNTS dimension table to the IA_SALES_IVCLNS fact table, you could join it to the EXT_SIVL_DIM1_ID.

   NOTE: Make sure that the level at which you define the Dimension Key ID in the fact mapping is the same grain at which the Key ID is defined in the dimension table's extract mapping.

5 Validate and save your changes to the repository.

To resolve the extension column Dimension Key ID in the fact load

1 In PowerCenter Workflow Manager, open the applicable source system configuration folder.

2 Double-click the applicable session for the fact load mapping to open the Edit Tasks box.

3 In the Transformations tab, edit the Lookup SQL Override field for the dimension key lookup in the ADI mapplet.

   By default, the lookup points to the `IA_DIMENSIONS` table.

4 Point the lookup to the new dimension table.

   For example, If you are joining the `IA_GL_ACCOUNTS` table, then you would change the references from `IA_DIMENSIONS` to `IA_GL_ACCOUNTS`.

5 Click OK, and then click OK to exit the Edit Tasks dialog box.

Resolving the Dimension Key Using Database Joins

The ADI provides lookups for dimension keys by default. However, if the dimension table is large, you can provide the dimension key to the ADI by joining the dimension table and the fact table in the database.

If you supply the key, then the ADI does not perform the lookup and instead resolves the dimension key within the load mapping itself. In this case, you modify the SQL statement in PowerCenter to join the tables.

To load a dimension key using a database join

1 In PowerCenter Designer, open the applicable source system configuration folder.

2 Open the applicable load mapping.

3 Add the dimension table (the table that contains the dimension key) as a source system to your load mapping.

4 Drag and drop the surrogate key column from the dimension source system definition to the Source Qualifier.

5 Double-click the Source Qualifier transformation to open the Edit Transformations box.

6 In the Properties tab, edit the SQL statement to put in the join conditions between the dimension table and the fact extract table.

7 Drag and drop the surrogate key column from the Source Qualifier to an available EXT_*_KEY port in the Source Adapter mapplet.

8 Validate and save your changes to the repository.

In Figure 33, the dimension key resolution is performed in the Customer Dimension table in the database by joining the Customer Dimension table to the Sales Orders extract table. The dimension key is then passed to the ADI, and is then loaded into the Sales Orders Fact table.

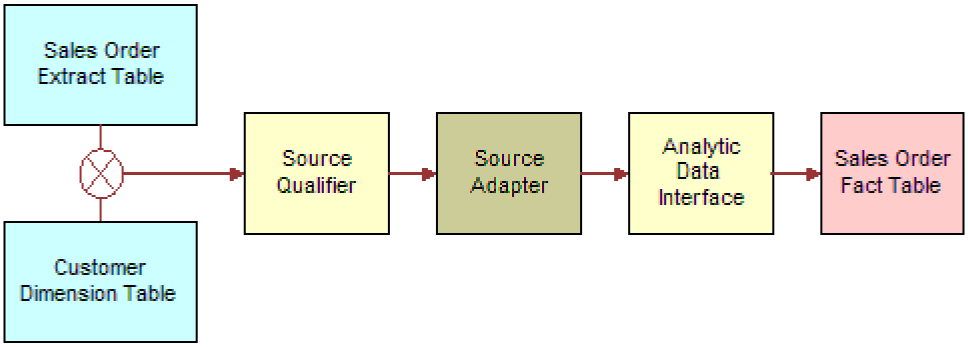


Figure 33. Dimension Key Resolution

About Domain Values

The Siebel Customer-Centric Enterprise 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 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 Siebel Customer-Centric Enterprise 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 Siebel Customer-Centric Enterprise Warehouse extracts data from both systems, it ports those source values through the extract mapping until the data reaches the IA_EVENT_GRP_CODE column in the TF_EVENT_TYPES Staging table.

The load mapping then ports the extracted source values (H and R from source system A, and 1 and 2 from source system B) into the Source Adapter mapplet. Within the Source Adapter, source values are translated into domain values (HIR and REH) based on a set of rules that are particular to your business practices.

Preparing to Define the Rules

You must define the rules so that the Source Adapter knows how to map your specific source values to the given set of domain values. Before you set up the rules you must:

1. Analyze all of your source values and how they map to the prepackaged domain values. You may find that you need to create additional domain values for particular columns. The result of this preparation work is a list of each source value and how it is mapped to a domain value.
2. Implement this logic in the applicable Source Adapter mapplet. To set up the logic, modify the Expression transformation in the Source Adapter mapplet for each affected column. For information on setting up the rules for domain values, see Configuring the Domain Value Set Using PowerCenter Designer on page 160.

Figure 34 illustrates how the source values are converted to the domain values—HIR and REH.

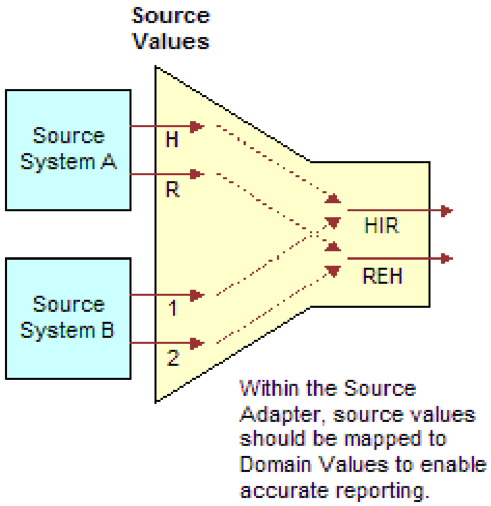

Source Values
Source System A
Source System B
H
R
1
2
HIR
REH
Within the Source Adapter, source values should be mapped to Domain Values to enable accurate reporting.

Figure 34. Source Values Translated to Domain Values

Figure 35 illustrates a different situation where the records may not contain a source value that flags the record as Hire or Rehire. In this case, the source system stores hires in one table and rehires in another table. To make this work, one possible solution is to modify the extract mappings to populate the IA_EVENT_GRP_CODE column with HIR or REH. If the field is populated in the extract mapping, you can then carry those same values through the Source Adapter mapplet.

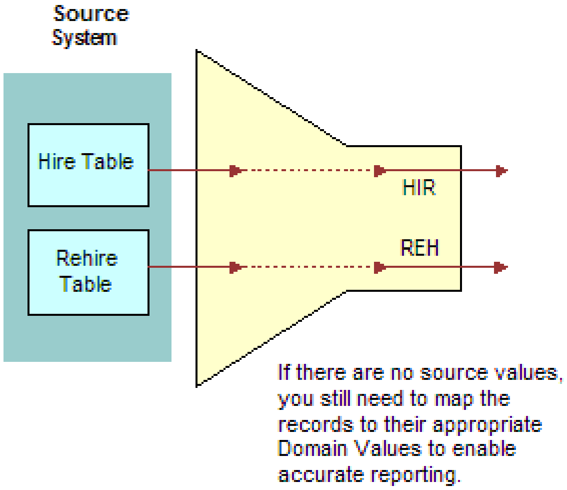

Source System
Hire Table
Rehire Table
HIR
REH
If there are no source values, you still need to map the records to their appropriate Domain Values to enable accurate reporting.

Figure 35. Records Mapped to Domain Values

After the Source Adapter mapplet converts the source-specific values to domain values, the domain values are inserted into a Siebel Customer-Centric Enterprise Warehouse table. In this example, the HIR and REH values populate the IA_EVENT_TYPES table, as illustrated in Figure 36.

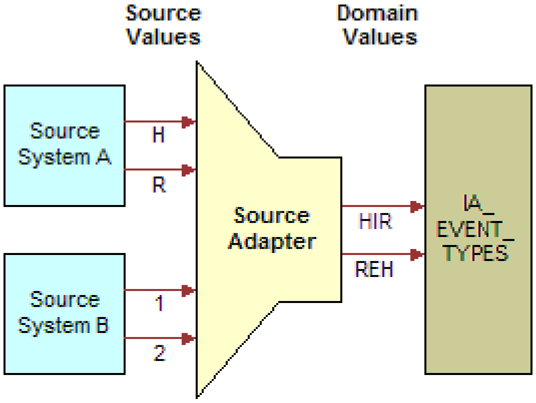


Figure 36. Domain Value Loading Siebel Customer-Centric Enterprise Warehouse Table

About the Importance of Domain Values

Values in the IA_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.IA_EVENT_GRP_CODE IN ('HIR','REH')) THEN EVNT.EVENT_CNT ELSE
0 END)
```

Rehires 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.IA_EVENT_GRP_CODE IN ('REH','HIR') THEN
EVNT.EVENT_CNT ELSE 0 END) = 0 THEN 0 ELSE SUM(CASE WHEN CMMNEVTP.IA_EVENT_GRP_CODE
IN ('REH') THEN EVNT.EVENT_CNT ELSE 0 END)/SUM(CASE WHEN CMMNEVTP.IA_EVENT_GRP_CODE
IN ('REH','HIR') THEN EVNT.EVENT_CNT ELSE 0 END) END
```

New Hire Count

This metric counts the headcount 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.IA_EVENT_GRP_CODE = 'HIR' OR CMMNEVTP.IA_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.IA_EVENT_GRP_CODE = 'HIR' OR CMMNEVTP.IA_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.IA_EVENT_GRP_CODE = 'HIR' OR CMMNEVTP.IA_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 Headcount - 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.IA_EVENT_GRP_CODE = 'HIR' OR CMMNEVTP.IA_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 Headcount - 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.IA_EVENT_GRP_CODE = 'HIR' OR CMMNEVTP.IA_EVENT_GRP_CODE = 'REH') AND
EVNT.EVENT_DK >= (CMMNDATE.DATE_KEY - 365) AND EVNT.EVENT_DK <= CMMNDATE.DATE_KEY
THEN EVNT.EVENT_CNT ELSE 0 END)
```

Each of these metric calculations is based on the domain values HIR and REH. All records whose source values are converted to one of these domain values are included in the metric calculations, as shown in Figure 37.

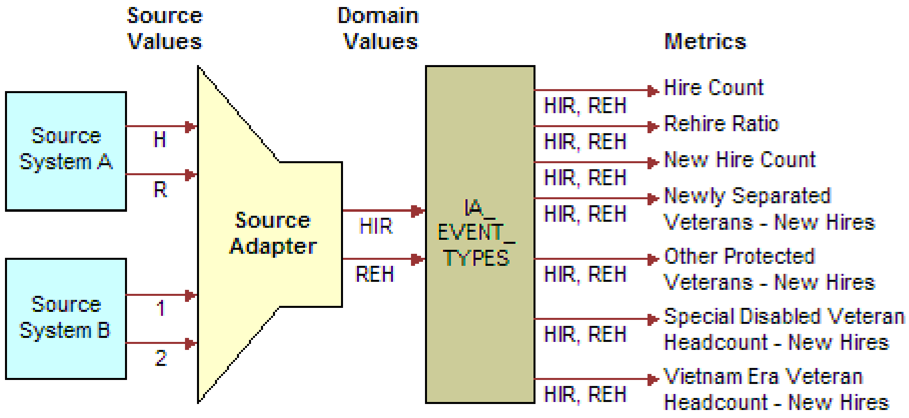

Source Values
Domain Values
Metrics
Source System A
H
R
Source System B
1
2
Source Adapter
HIR
REH
IA_ EVENT_ TYPES
HIR, REH
Hire Count
Rehire Ratio
New Hire Count
Newly Separated Veterans - New Hires
Other Protected Veterans - New Hires
Special Disabled Veteran Headcount - New Hires
Vietnam Era Veteran Headcount - New Hires

Figure 37. Domain Values Used in Metric Definitions

About Extending the Domain Value Set

The Siebel Customer-Centric Enterprise 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. To see which domain value sets you can modify, see the *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*. However, before you modify the domain value set for a particular column, you first perform impact analysis on existing metrics. For example, the Siebel Customer-Centric Enterprise Warehouse prepackages the following two events:

- **New Hire.** This event occurs when a new person is hired.

- **New Position.** This event occurs when a position is created, but an existing employee may be hired internally.

If you have an event that represents both a New Hire and a New Position, you may have to create a third event that depicts both. If you create this new event type domain value, you need to include it in the applicable metric definitions so as to account for all hires and positions.

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 Siebel Customer-Centric Enterprise Warehouse to allow you to create metric calculations independent of source system values. Siebel Customer-Centric Enterprise 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 Siebel Customer-Centric Enterprise 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 PowerCenter Designer. For more information on configuring domain values using PowerCenter Designer, see Configuring the Domain Value Set Using PowerCenter Designer on page 160.

For a list of CSV worksheet files and their domain values for your application, see your application configuration chapter.

To map source values to domain values using CSV worksheet files

1. Identify all the Siebel Customer-Centric Enterprise Warehouse table columns that use domain values.

   For a list of columns that use domain values, see the *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.

2. List all of your source values that qualify for conversion to one of the domain values.

3. Map each source value to a domain value.

   If any of your source system values do not map to a prepackaged domain value, and you may modify the list of domain values, then create a list of new domain values and map your orphaned source system values to your newly created domain values.

   You cannot modify all domain value sets. Also, you must check which metrics are affected by the modified domain value set. For more information, see the *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.

4 Open the CSV worksheet file in the `$Siebel\ETL\SrcFiles` folder.

5 Edit the file to map your source values to the existing domain values.

Alternately, if you want to add additional domain values, add them in this worksheet file.

6 Save and close the worksheet file.

Configuring the Domain Value Set Using PowerCenter Designer

If there are no worksheet files to map the source system values to the domain values, you need to modify the values using PowerCenter Designer. For more information on configuring the domain value set with CSV worksheet files, see Configuring the Domain Value Set with CSV Worksheet Files on page 159.

Configuring the domain value set for a particular column, using PowerCenter Designer, entails one or both of the following activities:

- Mapping source-specific values to domain values
- Adding more domain values to the prepackaged set of values

Regardless of which activity you choose, the configuration occurs in the Expression transformation of the applicable Source Adapter mapplet. The following procedure shows how to configure the Expression transformation to change the domain values.

To map source values to domain values using PowerCenter Designer

1 Identify all the Siebel Customer-Centric Enterprise Warehouse table columns that use domain values.

For a list of columns that use domain values, see the *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.

2 List all of your source values that qualify for conversion to one of the domain values.

3 Map each source value to a domain value.

If any of your source system values do not map to a prepackaged domain value, and you may modify the list of domain values, then create a list of new domain values and map your orphaned source system values to your newly created domain values.

You cannot modify all domain value sets. Also, you must check which metrics are affected by the modified domain value set. For more information, see the *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.

4 In PowerCenter Designer, open the applicable Source Adapter mapplet.

5 Open the Expression transformation.

6 Locate the applicable port’s expression so that you can modify it.

7. Edit the port's expression to map your source values to the existing domain values.

   Alternately, if you want to add additional domain values, add them in this same expression.

8. Save and validate your changes to the repository.

Configuring Conformed Dimensions

This section provides procedures on configuring objects that apply to more than one application.

Configuring Conformed Dimensions for Universal Source

This section provides configuration procedures for modifying dimensions that are loaded using a universal business adapter.

Product Effective Dates in the Products Dimension

The Siebel Customer-Centric Enterprise Warehouse stores product Effective To (SRC_EFF_TO_DT) and From (SRC_EFF_FROM_DT) dates in the Products dimension table, IA_PRODUCTS. In addition, the Products dimension stores a Support Withdrawn Date column, SPRT_WITHDRAWN_DT.

By default, the Support Withdrawn Date takes precedence over the product Effective To Date. This prioritization means that if you supply a value for the Support Withdrawn Date column in your flat file upload, the Siebel Customer-Centric Enterprise Warehouse uses that value as the product Effective To value as well, overwriting anything in the SRC_EFF_TO_DT column. You can change this default behavior by modifying the Products Expression in the Universal Source Products Extract mapping.

To modify the product Effective To Date logic for a flat file extract

1. In PowerCenter Designer, open the Configuration for Universal Source folder.
2. In the M_F_PRODUCTS_EXTRACT mapping, open the EXP_PRODUCTS expression.
3. Edit the logic for the SRC_EFF_TO_DT_OUT port.
4. Validate and save your changes.

Defining Services at the Package or Component Level

The Service Provisions dimension, `IA_SRVC_PRVSNS`, stores information about service or account packages, as well as the components that comprise these packages. When the Siebel Customer-Centric Enterprise Warehouse reports your product or service count, by default, they report it at the lower level of granularity—the individual service level. Therefore, if you have two service packages, Package1 and Package2, each consisting of ten components, your product count is twenty (at the component level), rather than only two (at the package level). You can change this logic to the higher level of granularity—the service package level.

To configure the granularity of the your services definition, use the Package Product ID instead of the Component Product ID to populate the Product ID column of your Service Provisions dimension.

To define the granularity of the Service Provisions dimension

1. Configure the contents of the `file_srvc_prvsns.csv` flat file that is used in the `M_F_SRVC_PRVSNS_EXTRACT` mapping.

   The `M_F_SRVC_PRVSNS_EXTRACT` mapping is found in the Configuration for Universal Source folder in PowerCenter Designer.

2. If you want the Service Provisions dimension to be configured at the package level of granularity, use the value of the `PACKAGE_PROD_ID` to populate the `PRODUCT_ID` column.

About Cross-Referencing Entities in the Siebel Customer-Centric Enterprise Warehouse

The Siebel Customer-Centric Enterprise Warehouse contains several dimension tables that store information about *entities,* which are companies and individuals who come in contact with your business. Because one entity can play multiple roles in your business, the Siebel Customer-Centric Enterprise Warehouse includes a cross-reference table, `IA_XRF_ENTITIES`, to maintain cross-reference information.

For example, assume that a vendor, who is defined in your Suppliers dimension table, `IA_SUPPLIERS`, also plays the role of a customer (and is therefore also defined in your Customers dimension table, `IA_CUSTOMERS`). The multiple roles of this customer or vendor would be tracked in the Cross Reference Entities table, `IA_XRF_ENTITIES`.

Table 24 provides the columns in IA_XRF_ENTITIES that are updated by means of a flat files. As new references come in from a flat file, the ENTITY_KEY column is updated, while the ORIG_ENTITY_KEY column retains its original value.

Table 24. Cross Reference Entities Table: IA_XRF_ENTITIES

IA_XRF_ENTITIES		
DIM_KEY_ID	ENTITY_KEY	ORIG_ENTITY_KEY
Customer1	Entity1	Entity1
Supplier1	Entity2	Entity2
Employee1	Entity3	Entity3
WebVisitor1	Entity4	Entity4

About Flat Files That Update Cross-Reference Entities Tables

Table 25 depicts the flat file columns that are used to update the IA_XRF_ENTITIES table. The sample shows that Supplier1 is the same entity as Customer1, and that the same is true for Employee1, indicating three separate roles for one individual. The sample flat file is structured correctly because the Reference Dimension ID column uses Customer as its reference all three times. If there were an additional record with EmployeeX as the Base Dimension and SupplierY as the Reference Dimension, that record would go into a separate file for upload so that every uploaded file uses only one dimension in its Reference Dimension ID column. This consistency allows the IA_XRF_ENTITIES table to update correctly.

Table 25. Flat File to Update the Cross-Reference Entities Table IA_XRF_ENTITIES

Flat File Source Base Dimension ID	Reference Dimension ID
Supplier1	Customer1
Employee1	Customer1
WebVisitor7	Customer23

Limiting the Entities Cross-Referenced in IA_XRF_ENTITIES

By default, the cross-reference entities load mappings load every entity from the following four dimension tables:

- Customers dimension: IA_CUSTOMERS
- Suppliers dimension: IA_SUPPLIERS
- Employees dimension: IA_EMPLOYEES
- Web Visitors dimension: IA_WEB_VISITORS

To limit which entities are loaded into the cross-reference table, edit the SQL statement in the Source Qualifier of the appropriate PLP cross-reference entities load mapping.

To limit the entities loaded into IA_XRF_ENTITIES

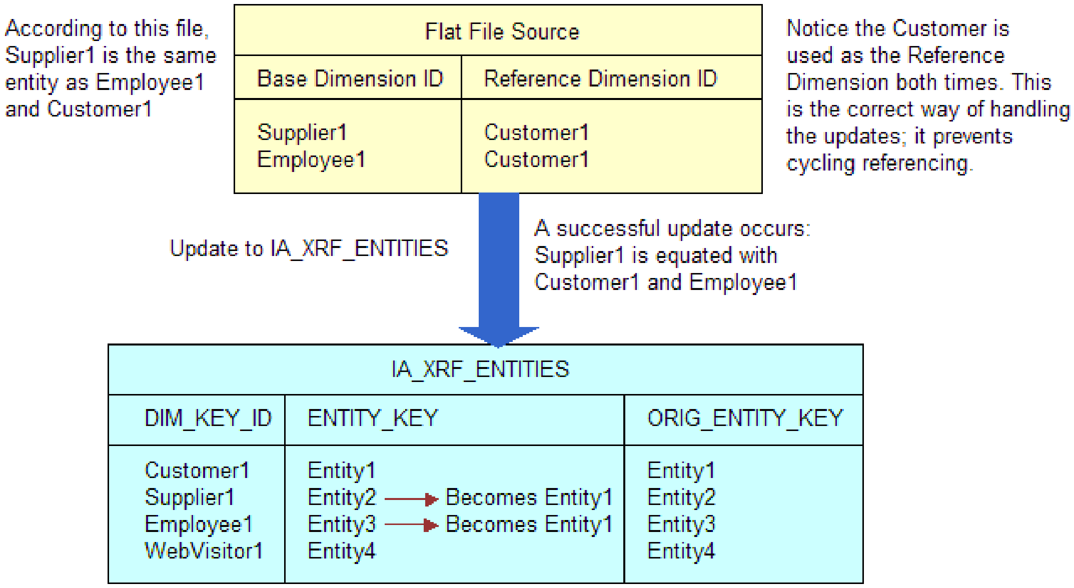
1. In PowerCenter Designer, open the Configuration for Post Load Processing folder.
2. Open the appropriate PLP cross-reference entities load mapping and edit the Source Qualifier for the type of record you want to limit on the cross-reference entities table:
   - **Customers.** Open the M_PLP_XRF_ENTITIES_CUSTOMERS_LOAD mapping, and edit the SQ_XRF_ENTITIES_CUSTOMERS_LOAD Source Qualifier.
   - S**uppliers.** Open the M_PLP_XRF_ENTITIES_SUPPLIERS_LOAD mapping, and edit the SQ_XRF_ENTITIES_SUPPLIERS_LOAD Source Qualifier.
   - **Employees.** Open the M_PLP_XRF_ENTITIES_EMPLOYEES_LOAD mapping, and edit the SQ_XRF_ENTITIES_EMPLOYEES_LOAD Source Qualifier.
   - **Web visitors.** Open the M_PLP_XRF_ENTITIES_WEB_VISITORS_LOAD mapping, and edit the SQ_XRF_ENTITIES_WEB_VISITORS_LOAD Source Qualifier.
3. Edit the SQL statement in the Source Qualifier to select only the desired set of entities.
4. Validate and save your changes to the repository.

About Specifying Correct Entity Cross-References in Flat File Uploads

It is recommended that you specify an entity type for the Reference Dimension ID when updating the Cross Reference Entities table (IA_XRF_ENTITIES) from a flat file. If you use random dimensions as the Reference Dimension ID, you risk making a cyclic reference in IA_XRF_ENTITIES, in which case the M_PLP_XRF_ENTITIES_CONSOLIDATION mapping creates a partial or incorrect cross-reference.

Figure 38 depicts a correctly structured flat file that matches the Base Dimension ID to the Reference Dimension ID, enabling a proper update to the IA_XRF_ENTITIES table. In the first row, the flat file shows that Supplier1 is the same entity as Customer1. The corresponding update to the IA_XRF_ENTITIES table updates the second row, changing Supplier1’s ENTITY_KEY to be the same as Customer1. Supplier1 and Customer1 now share the same ENTITY_KEY— Entity1.

The next row in the flat file indicates that Employee1 is also the same entity as Customer1 (and, therefore, as Supplier1). The corresponding update to the IA_XRF_ENTITIES table updates the third row, changing Employee1's ENTITY_KEY value to Entity1, matching that of Customer1 and Supplier1.



According to this file, Supplier1 is the same entity as Employee1 and Customer1

Flat File Source

Base Dimension ID	Reference Dimension ID
Supplier1	Customer1
Employee1	Customer1

Notice the Customer is used as the Reference Dimension both times. This is the correct way of handling the updates; it prevents cycling referencing.

Update to IA_XRF_ENTITIES

A successful update occurs: Supplier1 is equated with Customer1 and Employee1

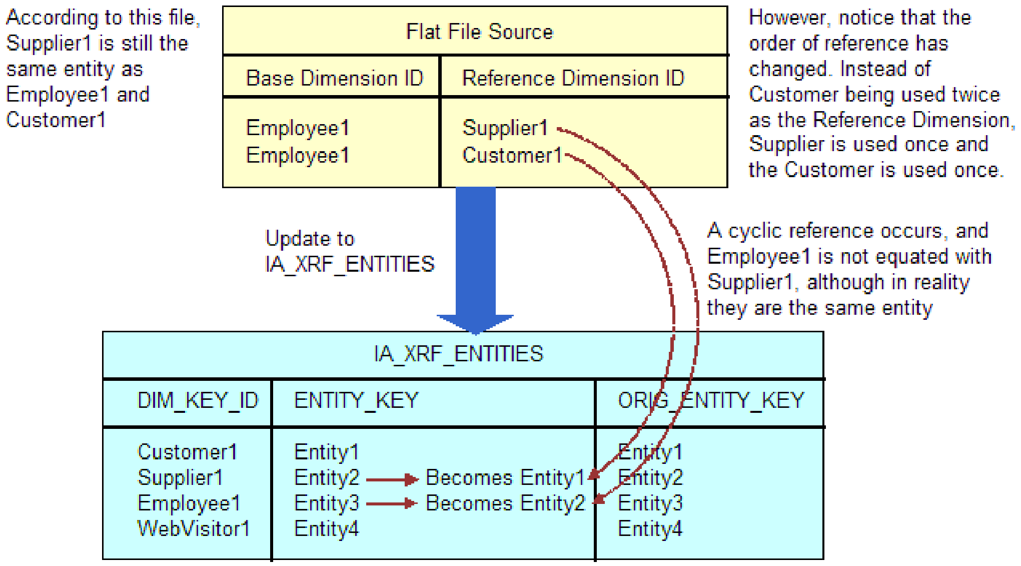
IA_XRF_ENTITIES

DIM_KEY_ID	ENTITY_KEY	ORIG_ENTITY_KEY
Customer1	Entity1	Entity1
Supplier1	Entity2 → Becomes Entity1	Entity2
Employee1	Entity3 → Becomes Entity1	Entity3
WebVisitor1	Entity4	Entity4

Figure 38. Successful Update of IA_XRF_ENTITIES Using the Same Entity Type as the Reference Dimension

Figure 39 depicts what happens if the entities used as Reference Dimension IDs in the flat file are cyclic from the definition. Supplier1, Employee1, and Customer1 are all still the same entity, as they were in the Figure 38. However, in the previous example, the Customer entity was consistently used as the Reference Dimension, whereas in Figure 39, the Supplier entity is used once and the Customer entity is also used only once.

The first row of the flat file links Employee1 to Supplier1. The corresponding update to the IA_XRF_ENTITIES table changes the ENTITY_KEY of Employee1 to Entity2, matching the ENTITY_KEY value of Supplier1. The second row of the flat file links Supplier1 to Customer1. The corresponding update to the IA_XRF_ENTITIES table changes the ENTITY_KEY column of Supplier1 to the value of Entity1. The cross-reference, then, is only partial—it identifies Supplier1 and Customer1 as the same entity, but fails to indicate that Employee1 is yet another role played by that very same entity.



According to this file, Supplier1 is still the same entity as Employee1 and Customer1

Flat File Source	
Base Dimension ID	Reference Dimension ID
Employee1 Employee1	Supplier1 Customer1

However, notice that the order of reference has changed. Instead of Customer being used twice as the Reference Dimension, Supplier is used once and the Customer is used once.

A cyclic reference occurs, and Employee1 is not equated with Supplier1, although in reality they are the same entity

IA_XRF_ENTITIES		
DIM_KEY_ID	ENTITY_KEY	ORIG_ENTITY_KEY
Customer1 Supplier1 Employee1 WebVisitor1	Entity1 Entity2 → Becomes Entity1 Entity3 → Becomes Entity2 Entity4	Entity1 Entity2 Entity3 Entity4

Figure 39. Unsuccessful Update of IA_XRF_ENTITIES Caused by Cyclic Reference

NOTE: After loading cross-references, make sure all relationships are accurately defined.

9 Configuring Siebel Customer-Centric Enterprise Warehouse for Oracle 11i

Regardless of which application you are implementing, there is some general configuration information that is specific to Oracle 11i. In this chapter, you learn about each of these points.

This chapter contains the following topics:

Configuring the Group Currency Code for Oracle 11i

Oracle 11i provides document and local currencies, but not a group currency. Therefore, the Siebel Customer-Centric Enterprise Warehouse uses a flat file called `grp_curr.csv` that contains a group currency code, which is loaded into the data warehouse during the initialization phase. It is initially set as USD for United States dollars. However, if your group currency is different, you must edit the currency code in the flat file before you load it. For example, if your group currency is the European Euro, then you might set the currency code to EUR.

To configure the group currency code for Oracle 11i

1. Open the `grp_curr.csv` file using Microsoft WordPad or Notepad in the $pmserver\srcfiles folder.
2. Edit the group currency code.
3. Save and close the file.

About Adding Oracle Flex Fields to the Data Model

Oracle Applications have two kinds of flex columns:

- **Key flex fields.** Key flex fields are key fields that are required by Oracle 11i. You can modify their definitions when configuring Oracle Applications.

- **Descriptive flex fields.** Descriptive flex fields are extension fields in the Oracle Applications database.

Key Flex Fields

The Siebel Customer-Centric Enterprise Warehouse has prepackaged mappings to extract some of the key flex field data. However, if you configure any of these key flex fields other than the default configuration prepackaged by the Siebel Customer-Centric Enterprise Warehouse, you must modify the mappings. The following are the key flex fields preconfigured for extraction:

- **GL Account, account segment only.** If you want to incorporate any other segment, you must modify the extract and load mappings.
- **Territory, segments 1, 2, and 3 only.** These key flex fields are used for Business Organizations Sales Geographical hierarchies only. If you want to incorporate any other segments, you need to modify the extract and load mappings.
- **Product Category, segments 1 and 2 only.** These key flex fields are used for classification only. If you want to incorporate any other segments, you need to modify the extract and load mappings.

If you wish to add other key flex field data to the data model, it is recommended that you use the extension columns available in the tables. For more information on using extension columns, see Overview of Storing, Extracting, and Loading Additional Data on page 175.

Descriptive Flex Fields

The Siebel Customer-Centric Enterprise Warehouse prepackages different types of extension columns that you can use to hold any additional data. For more information on using extension columns, see Overview of Storing, Extracting, and Loading Additional Data on page 175.

Setting the Organization ID

The Organization ID for Oracle Applications defaults to 204 as the invoice information in Oracle Applications does not have an Organization ID for inventory organizations. This column must be configured with the correct Organization ID or the ADI is not able to resolve dimension keys, such as the Product key and Sales Product Key, for the applicable fact load.

For example, by default the Sales Order Lines table supplies the load mapping with the Product key. However, if it does not, the ADI performs a lookup to retrieve the Product key from IA_PRODUCTS. The lookup uses the Organization ID to help resolve the Product key. The ADI uses the SOURCE_ID, CREATED_ON_DT, and the PRODUCT_ID, where the PRODUCT_ID is defined as the INP_INV_ITEM_ID concatenated with the ORGANIZATION_ID. If the wrong Organization ID is provided, then the Product ID is not defined correctly, which results in a failed lookup for the Product key, and the Product key is not loaded into the applicable fact table during the load.

If your business has multiple Organization IDs for inventory organizations, you can use the most commonly used Organization ID, or the master Organization ID as defined in your Oracle Applications instance.

To reset the Organization ID

1 In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.

2 Edit all applicable Source Adapter mapplets.

The following is a list of applicable mapplets for Oracle 11i:

- MPLT_SAI_SALES_IVCLNS
- MPLT_SAI_GL_REVENUE
- MPLT_SAI_GL_REVENUE_UPDATE

3 Double-click the Expression transformation to open the Edit Transformation box.

4 In the Ports tab, edit the expression for the VAR_ORGANIZATION_ID.

By default, it is set to 204. Edit it by setting it to the master organization ID.

5 Validate and save your changes to the repository.

Mapping Source Customer Hierarchies to the Customers Dimension Table

Customer hierarchies are typically custom-defined in Oracle Applications. If you want to include this data in the data warehouse, you must perform two customization processes:

- Import the hierarchies into the TI_CUSTOMERS staging table for Oracle 11i.
- Redefine the category lookup so that the new category data is loaded into the Siebel Customer-Centric Enterprise Warehouse.

The two processes are illustrated in Figure 40.

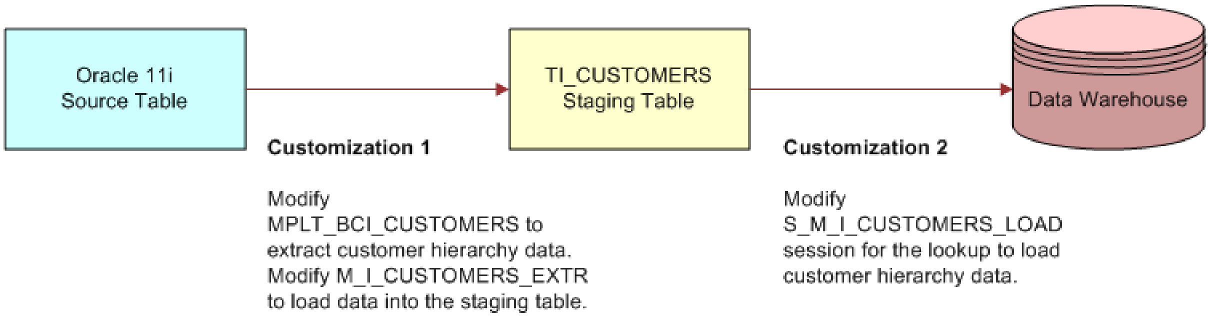


Figure 40. Oracle Applications: Customization Processes for Custom Customer Hierarchies Load

To load the source-defined customer hierarchies into the TI_CUSTOMERS staging table for Oracle 11i, you must first edit the MPLT_BCI_CUSTOMERS Business Component mapplet to extract the hierarchy in addition to the customer information. After your Business Component is set up to extract the customer hierarchies, you must verify that the M_I_CUSTOMERS_EXTRACT extract mapping outputs this data to the data warehouse.

To map Oracle-defined customer hierarchies to the Customers dimension table

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. Open the `MPLT_BCI_CUSTOMERS` Business Component mapplet for Oracle 11i.

   Modify the mapplet to extract the customer hierarchy columns.

   If the hierarchies and customers are maintained in the same Oracle Applications source table, then load these columns into the SQL Source qualifier and map them to the Business Component output.

   However, if the hierarchies and customers are stored in two different tables, then the Business Component must be modified to include both source tables so that it can include both sets of information.

   - Modify the `M_I_CUSTOMERS_EXTRACT` extract mapping, to map the source customer hierarchy columns to the extension hierarchy columns in the `TI_CUSTOMERS` staging table.

   If the source table has hierarchy codes, but no descriptions associated with these codes, map the Oracle Applications codes to both the hierarchy name columns and the hierarchy code columns. Hierarchy name columns are named as `CUST_HIER`*`X`*`_NAME`, where X denotes the level of the customer hierarchy.

   If the source table has code values, but the corresponding descriptions are in a different source table, you must build new codes mappings that load the data into the `IA_CODES` table.
3. Save your changes to the repository.

   NOTE: After you complete the previous process, you must modify a hierarchy lookup in the customer dimension so that the system extracts the new categories.

To configure the category lookup

1. In PowerCenter Workflow Manager, open the Configuration for Oracle Applications v11i folder.
2. Open the `S_M_I_CUSTOMERS_LOAD` session for Oracle 11i, to open the Edit Tasks box.
3. In the Transformations tab, modify the lookup in the `MPLT_ADI_CUSTOMERS.LKP_CUST_HIER`*`X`* field in the `IA_CODES` table by adding the new category in the SQL statement.

   Select the arrow to edit the `WHERE` clause.

   You can use the following statement as a sample of how to structure your SQL statement:

   ```
   WHERE IA_CODES.CATEGORY = 'GENERIC'AND IA_CODES.LANGUAGE = 'E'
   ```

   For example, if you have mapped something to the `CUST_HIER1_CODE,` then the SQL to the `IA_CODES` table is now a new category code in place of `GENERIC`.
4. Validate and save your changes to the repository.

Configuring Product Categories

As initially configured, the Siebel Customer-Centric Enterprise 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 categories. Therefore, you must reconfigure your product categories by making two customizations:

- Identify and extract only the categories that you want to report against.
- Properly position the data so that it loads into the Siebel Customer-Centric Enterprise Warehouse.

There are two dimension tables that have built-in product hierarchies—the Product and Sales Product dimension tables. These dimension tables share one category staging table. ETL extracts the Product and Sales Product staging tables separately, and then joins these tables with the shared category staging table to load hierarchies. The category extract mapping controls the category sets that are used in the Product and Sales Product dimensions. The load mappings for the Product and Sales Product dimensions specifies which Category Set is used to load to the hierarchy columns.

To configure product category extract from Oracle Applications

1. Identify the categories that need to be mapped to the extension hierarchy columns.

   These categories are extracted from the source and placed in the hierarchy column specified.

2. In PowerCenter Workflow Manager, open the Configuration for Oracle Applications v11i.
3. Open the S_M_I_CATEGORY_EXTRACT session with the Task Developer.
4. In the Mappings tab, click MPLT_BCI_CATEGORY.SQ_MTL_CATEGORIES.
5. In the right pane, scroll down and click SQL Query to edit session SQL override.
6. Click the arrow to edit the WHERE clause.

   The following statement is an example of how to structure the WHERE clause:

   ```
   WHERE...MTL_CATEGORY_SETS_B.CATEGORY_SET_ID IN (27,2)
   ```

   In this example, the WHERE clause extracts categories where the Category Set ID is 27 or 2.

7. Click OK, and then click OK to close the Edit Tasks box.

To modify the default Category Set

1. Open the file_parameters_ora11i.csv file in the $pmserver\srcfiles folder
2. Replace the default Category Set ID (27) with your new value.

   The following table is an example of the file_parameters_ora11i.csv file.

S_M_I_SALES_PRODS_LOAD:CATEGORY_SET_ID	0	0	0	0	S	27
S_M_I_PRODUCTS_LOAD:CATEGORY_SET_ID	0	0	0	0	S	27

3 Save and close the file.

To reconfigure the product hierarchy loads

1 In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder, and expand the Mapplets folder.

2 Open the MPLT_SAI_PRODUCTS mapplet for Oracle 11i.

3 Double-click the EXP_PRODUCTS expression transformation to open the Edit Transformation box.

4 In the Ports tab, scroll down to find the hierarchy code port.

Hierarchy levels are named with the following convention EXT_PROD_HIER*X*_CODE, where X denotes the level within the hierarchy. For example, if you want to edit the first level of your hierarchy, you must edit the definition for EXT_PROD_HIER1_CODE port. The first two levels are preconfigured as follows:

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

INP_SEGMENT1||'~'||INP_SEGMENT2)

EXT_PROD_HIER2_CODE = INP_SEGMENT1
```

5 Modify the expression that defines your hierarchy code.

6 Validate and save your changes to the repository.

NOTE: You can configure six hierarchy extension columns in the Siebel Customer-Centric Enterprise Warehouse. To resolve the names for each level, you need to extend the IA_CODES table with the correct codes when configuring the new hierarchy levels.

Modifying the Source ID

The Source ID is set to OAP11i in the mappings corresponding to the Oracle sessions. The `file_parameters_ora11i.csv` file contains the value for the $$SOURCE_ID field for each of the sessions.

After you have modified the file, the next time you run the session, a new Source ID is loaded.

To modify the Source ID

1 Open the `file_parameters_ora11i.csv` file using Microsoft WordPad or Notepad in the `$pmserver\srcfiles` folder.

2 Replace the default Source ID with the new Source ID.

Perform this action for all applicable sessions.

3 Save and close the file.

Modifying the Last Extract Date

By default, the $$LAST_EXTRACT_DATE is set to 01/01/1970 for Oracle 11i. However, if you need to modify this date, you can do so by modifying the INFA_PARAMETERS.PAR file. After you have modified the file, the next time you run the session, the new Last Extract date is populated.

To modify the Last Extract date

1. Open the `file_parameters_ora11i.csv` file.
2. Replace the default Last Extract date with the new Last Extract date.

   Perform this action for all applicable sessions.
3. Save and close the file.

10 Storing, Extracting, and Loading Additional Data

This chapter discusses the methodology for storing additional data in the data warehouse. In addition, it gives general procedures for extracting and loading new data.

This chapter includes the following topics:

Overview of Storing, Extracting, and Loading Additional Data

Siebel Customer-Centric Enterprise Warehouse contains an open (source-independent) data model. When you set up your data warehouse, you find that the packaged mappings load most of your source data. However, you may have additional data that does not have a storage place. To accommodate this scenario, Siebel Customer-Centric Enterprise Warehouse has set aside extension columns in most warehouse tables that act as placeholders for additional data.

Extension columns make it possible to extend any fact or dimension table without changing the schematic structure of the Siebel Customer-Centric Enterprise Warehouse data model, or making modifications to the load mapping, because load mappings already include the extension column loading logic. In addition, extension columns have far less impact on the database size than building entirely new tables, which could also have implications when upgrading. Each of the following sections provides you with information on the preferred methods of integrating different types of data without affecting database size or impeding functionality.

TIP: It is recommended that you use the extension columns, instead of changing the data model structure. Using extension columns, you can use data warehousing practices such as change control, updates, generating surrogate keys, and slowly changing dimensions.

About Classifying Additional Data

There are three types of data that you might want to store in the Siebel Customer-Centric Enterprise Warehouse:

- **Attributes.** Attributes include descriptive data that allow you to look at metrics under different circumstances. Examples of attributes are product name, product description, and sales order number.

- **Amounts and Quantities.** Amounts and quantities are two types of metrics, which are sometimes referred to as facts. Amounts are monetary values, such as costs, revenues, profits, and so on. Quantities are counts of items, such as the number of sales orders, number of products sold, and number of backlogged sales orders.
- **Dates.** Dates are self-explanatory.

Depending on the type of data that you want to store in the data warehouse, you may choose a particular type of table, as well as an extension column with a particular data type. Siebel Customer-Centric Enterprise Warehouse prepackages extension columns in fact tables, dimension tables, and class tables, where each type of table contains different types of extension columns.

The following sections suggest the types of tables and columns to use when storing particular types of data in your data warehouse.

Types of Extension Columns

Each fact and dimension table has a unique set of extension columns. The following sections outline the types of extension columns for each type of table.

There are two types of dimension tables. Table 26 provides descriptions of the types of extension columns present in both types of dimension tables. The only difference between the two types is that one type of dimension table has more columns than the other.

Table 26. Extension Columns in Dimension Tables

Type of Extension Column	Description	Number of Columns
Code	You can use these columns to store codes.	3 to 10
Code Name	You can use these columns to store code names. Code names are looked up to decipher cryptic codes.	3 to 10
Text	You can use these columns to store text. This is a pass-through column.	3 to 10

Fact tables generally have more types of extension columns than dimension tables. This is due to the nature of a star schema. Table 27 provides descriptions of the extension columns present in the two types of fact tables. The only difference between the two types is that one type of fact table has more columns than the other.

Table 27. Extension Columns in Fact Tables

Type of Extension Column	Description	Number of Columns
Code	You can use these columns to store codes.	3
Code Name	You can use these columns to store code names. Code names are looked up to decipher cryptic codes.	3
Text	You can use these columns to store text. This is a pass-through column.	3 or 8
Date Key	You can use these columns to store date keys. The ADI transforms any dates into Julian dates.	3
Amount	You can use these columns to store monetary values. Amounts come in triplets—document, local, and group currencies. For information on the three types of currencies, see About Document, Local, and Group Currencies on page 136.	9 (3 Document, 3 Group, 3 Local)
Quantity	You can use these columns to store units.	3 or 6
Dimension Key	You can use these columns to store dimension keys. You can use this to join the fact table to a new dimension table.	3 or 5

In the following sections, you learn how to incorporate attributes, metrics, and dates into fact, dimension, and class tables.

About Adding More Attributes to the Data Model

There are two different types of attributes—domained and free text. Domained attributes are restricted to a set of values. An example of a domained attribute is a State column that requests a state within the United States. The State column’s domain of values consists of all the states in the United States.

NOTE: Siebel Customer-Centric Enterprise Warehouse domain values are different from domained attributes. Domained attributes are source values, not values necessarily loaded into Siebel Customer-Centric Enterprise Warehouse; they are a set of values that are used for a particular field. On the other hand, domain values are Siebel Customer-Centric Enterprise Warehouse values for particular fields that are used to create metrics. These values are called Siebel Customer-Centric Enterprise Warehouse domain values, which are generically referred to as domain values. For more information on domain values, see About Domain Values on page 154. For a list of domain values, see the *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.

In addition to domained attributes, your source system may also provide free text attributes, which can have any value. Unlike domained attributes, where there is a select set of values, there are no restrictions on the value of free text attributes. An example of a free text attribute is a description column in which users enter a description. Descriptions can vary widely; there is no standard set of values for descriptions.

Depending on the type of attribute you want to store, domained or free text, there are different recommendations on how to incorporate them into your data warehouse. The following sections describe how to store each type of attribute.

Determining the Type of Table to Use for Attributes

Before you can store domained or free text attributes in the data warehouse, you must first decide what table the data goes in. The following sections describe the options available to you; the options are listed in the order that they are recommended.

First Recommendation—Store Additional Attributes in an Existing Dimension or Fact Table

The preferred option is to store additional attributes is an existing dimension table that primarily stores this type of data. There are a few requirements that must be met before you can do this:

- The attribute stored in the dimension table must be at the same base grain as the table. Changing to a lower base grain may negatively affect joins to other tables, and it is therefore recommended that you do not change the base grain of the table.

- The relationship between the dimension table base grain to the attribute can be one-to-one, or many-to-one, but not one-to-many. For example, assume that the IA_PRODUCTS table’s grain is the product number. So, for example, you cannot incorporate a store location code column, which takes the base grain of product, and matches it with several store attributes, changing the grain from products code to store code. However, you can add a color column to track that attribute of the product without changing the grain of the table.

- Given the limited number of extension columns, you must be selective when choosing data that you want to incorporate into the data warehouse. If you require more extension columns than are provided, keep attributes that are the most closely associated with the dimension table in that table, and place all other attributes in the other tables. For example, if you had a dimension table that covered the attributes of storage capacity for your warehouse, and you had both additional storage and location attributes to incorporate, you would choose to create a new location table, rather than split the storage capacity information. For information on creating new tables, see Table Formats on page 191.

Second Recommendation—Store Additional Attributes in a New Dimension Table and Extension Tables to Existing Dimensions

If you have a group of related attributes to load, but they cannot be incorporated into an existing dimension table or class table, then your final option is to create a new dimension table. For example, if you have a group of attributes that all relate to Profit Centers, then you can create a Profit Center dimension table. It is recommended that you create dimension tables for attributes that are grouped by a business area. For information on creating new tables see Table Formats on page 191.

Do not create a dimension table to store a disparate set of attributes. If you decide to create a new dimension table, use the same structure and naming conventions as the prepackaged dimension tables. Structurally, the new dimension table must contain columns such as primary key, Source ID, Key ID, fact keys, dimension keys, and so on. In addition, with each of these columns, there are naming conventions. For information on naming conventions, see the *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.

If you need to add more columns to a dimension table and if all the extension fields of the existing dimension are used, create an extension table to the dimension table with the same Surrogate Key, Key ID, source if and effective from, and two dates to populate these extra fields.

Determining the Type of Extension Column to Use for Attributes

When you have determined which type of table to store the attribute in, you then have to decide what type of extension column to use. The extension column you use depends on the type of attribute. There are two types of attributes data—free-text and domained.

Free Text Attributes

It is recommended that you use a Text extension column as a placeholder for free text attribute data. Text extension columns are available in fact, dimension, and class tables. The Text data type extension column offers the most character space; they are varchar (254).

Domained (Codes and Code Names) Attributes

It is recommended that you use Code Name and Code extension columns as a placeholder for domained attribute data. Referring back to the sample domained attribute State, the State column's domain of values is restricted to the states in the United States, which can be denoted by a code. For example, California can be denoted by CA. In this case, the code name is California and the code is CA. With domained attributes, you can issue codes to each possible value, because you know the possible values for this variable. However, do not use codes for columns that have too many values, because it could significantly slow the processing time. For example, even though you might use a Code and Code Name extension column for a state, you might not use it for cities.

If a code does not exist in the IA_CODES table and you want to use a lookup for the code name, then you must add the code to the IA_CODES table. Three possible scenarios occur when working with codes and code names. Each of the following scenarios are also described in Figure 42. The three scenarios are described in the sections that follow.

Storing Codes and Code Names in the Siebel Customer-Centric Enterprise Warehouse

If the code and code name pair reside in the IA_CODES table and you load only the code column, the ADI resolves the code name through a lookup. (See Figure 41.) This is the way to enforce domained values. For information on adding new codes to the IA_CODES table, see Creating a Codes Mapping on page 210.

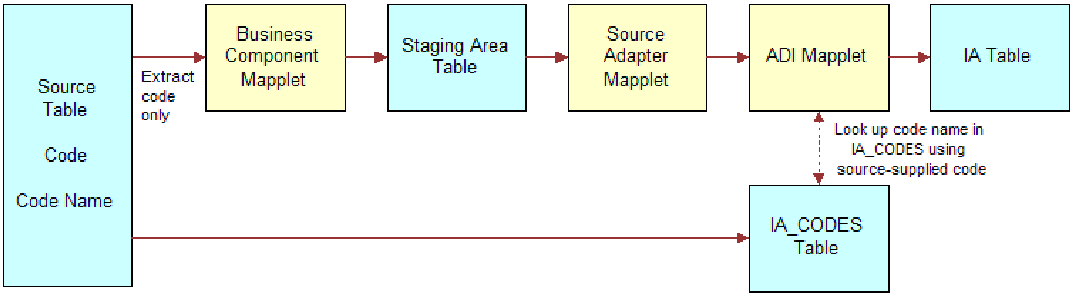


Figure 41. Storing Codes and Code Names in an IA Table

Storing Codes Without Code Names in the Siebel Customer-Centric Enterprise Warehouse

If the source supplies only the code without the code name, and the code and code name pair do not exist in the IA_CODES table, then the ADI tries to resolve the code name by a lookup. However, because the code and code name do not exist in the IA_CODES table, no data is retrieved. As a result, the ADI loads only the code into the IA table. This method does not enforce domained values.

Storing Code Names Without Codes in the Siebel Customer-Centric Enterprise Warehouse

If the source supplies the code name without the code, then the ADI loads only the code name into the IA table. (See Figure 42.) This method does not enforce domained values.

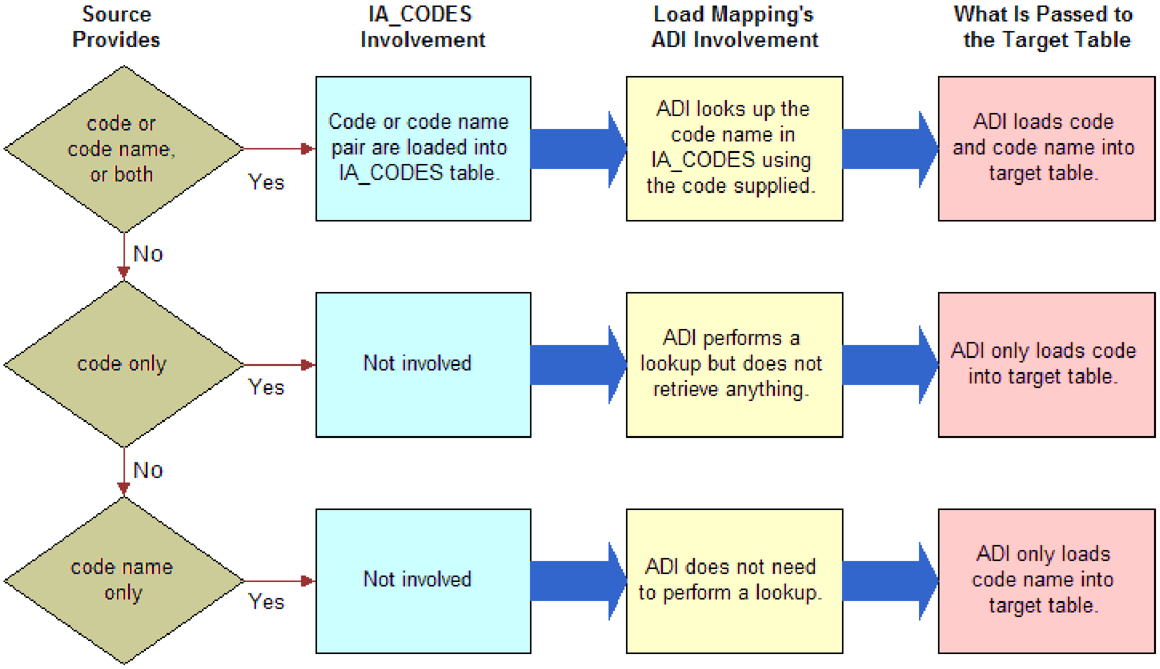

Source Provides
IA_CODES Involvement
Load Mapping's ADI Involvement
What Is Passed to the Target Table
code or code name, or both
Yes
Code or code name pair are loaded into IA_CODES table.
ADI looks up the code name in IA_CODES using the code supplied.
ADI loads code and code name into target table.
No
code only
Yes
Not involved
ADI performs a lookup but does not retrieve anything.
ADI only loads code into target table.
No
code name only
Yes
Not involved
ADI does not need to perform a lookup.
ADI only loads code name into target table.

Figure 42. ADIs Handling of Code and Code Names

Storing Additional Attributes in the Data Warehouse

Siebel Customer-Centric Enterprise Warehouse comes with several prepackaged attributes. Because you may want to incorporate additional attributes to the data warehouse, this section provides detailed instructions for storing free text and domained attributes.

There are a variety of scenarios that occur when storing a new attribute. Depending on where the data resides, you can take particular steps to incorporate new attributes and make them available in the data warehouse and your front-end schema. Generally speaking, there are three major areas where data resides—Source database, staging table, and Siebel Customer-Centric Enterprise Warehouse. Figure 43 illustrates the three scenarios and the components that are affected when trying to store the new data.

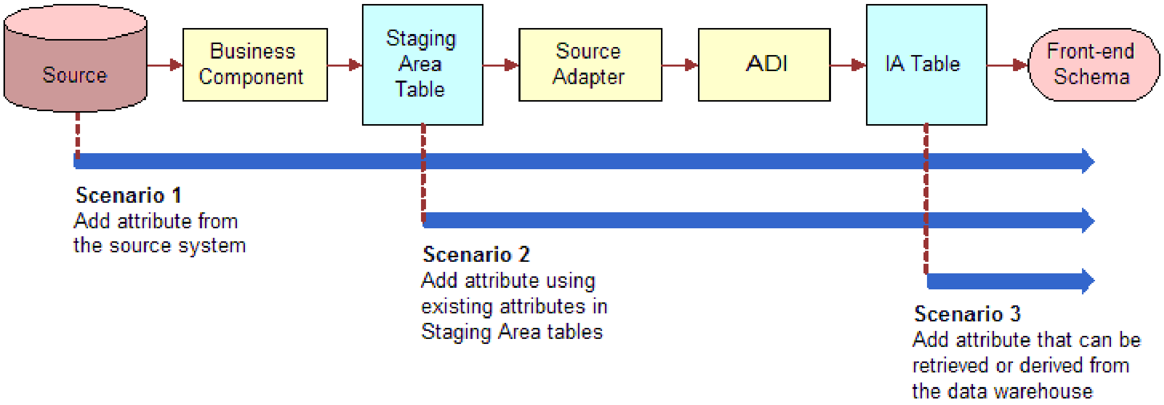

Source
Business Component
Staging Area Table
Source Adapter
ADI
IA Table
Front-end Schema
Scenario 1
Add attribute from the source system
Scenario 2
Add attribute using existing attributes in Staging Area tables
Scenario 3
Add attribute that can be retrieved or derived from the data warehouse

Figure 43. Scenarios for Storing New Attributes

In the sections that follow, you can find procedures for storing new attributes for each of the three scenarios. Each of the steps in the procedure corresponds to larger topics described in later sections of this chapter. The steps provide you with a high-level overview, and the larger topics provide the details.

To store additional attributes from the source system

1 If applicable, load the code and code name into the IA_CODES table.

   This step applies only if you are storing domained attributes. In addition, perform this step only if the code and code name are not already loaded into the IA_CODES table. For information on creating a codes mapping, see Creating a Codes Mapping on page 210.

2 Modify the existing extract mapping or create a new extract mapping to extract this new information from the source system.

For information on modifying the existing Business Component mapplet of an extract mapping, see Process of Creating and Modifying Business Adapters on page 213. For information on creating a new extract mapping, see Creating an Extract Mapping on page 203.

TIP: You can perform calculations in the Business Component mapplet of the extract mapping or in the Source Adapter mapplet of the load mapping. However, it is not recommended that you perform performance-expensive calculations in the extract. This protects you from interfering with your source transaction system. For these types of calculations, it is recommended that you perform them in the Source Adapter mapplet in the load mapping. One of the major reasons why Siebel Customer-Centric Enterprise Warehouse splits the extract process from the load process into two separate mappings is to minimize the amount of time tying up your source transaction system.

3 Modify the existing load mapping to take the data from the staging table and load it into the data warehouse.

For information on modifying the existing Source Adapter mapplet of a load mapping, see Process of Creating and Modifying Business Adapters on page 213. For information on creating a new load mapping, see Creating a Load Mapping on page 206.

TIP: If you map the data to a staging table's extension column, you must determine the type of extension column to use. For information on the type of extension column to use for attribute data, see Determining the Type of Extension Column to Use for Attributes on page 179.

4 Incorporate the new attribute into your front-end schema.

To store additional attributes derived from staging area objects

1 Modify the existing load mapping to take the data from the staging table and load it into the data warehouse.

For information on modifying the existing Source Adapter mapplet of a load mapping, see Process of Creating and Modifying Business Adapters on page 213. For information on creating a new load mapping, see Creating a Load Mapping on page 206.

TIP: If you map the data to a staging table's extension column, you must determine the type of extension column to use. For explicit information on the type of extension column to use for attribute data, see Determining the Type of Extension Column to Use for Attributes on page 179.

2 Incorporate the new attribute into your front-end schema.

NOTE: This procedure can also be used to store additional metrics. For more about storing metrics, see Storing Additional Metrics in the Data Warehouse on page 184.

About Adding More Metrics to the Data Model

There are two different types of metrics—quantities and amounts. Quantity metrics are basically counts of items. For example, the number of dining tables ordered is a quantity. Amount metrics are monetary values. For example, the cost of a dining table is an amount metric.

Determining the Type of Table to Use for Metrics

Regardless of the type of metric, it is recommended that you load metric data into fact tables.

However, before loading a metric into a fact table, perform the following checks:

- You must make sure that the grain of the table is the same grain as the new metric. If the table and the data are not at the same grain level, it is strongly recommended that you do not change the grain of a table to match the grain of the metric data. By doing so, you may negatively affect joins to other tables, as well as reports.
- Determine the type of extension column you use to store the metric. Different extension columns are provided for both the quantity and amount type of metric. Extension columns for quantity are identified by the QTY suffix, and extension columns for amount are identified by the AMT suffix.
- The metric data must be associated with the subject area of the fact table.

In addition to determining the appropriate fact table, you must also determine which type of extension column to load the metric data into. The following section suggests the type of column to use for each type of metric data.

Determining the Type of Extension Column to Use for Metrics

It is recommended that you use a Quantity extension column (*_QTY) to store quantity metric data. As well, it is recommended that you use an Amount extension column (*_AMT) to store amount metric data.

Storing Additional Metrics in the Data Warehouse

Siebel Customer-Centric Enterprise Warehouse comes with several prepackaged metrics. Because you may want to store additional metrics to the data warehouse, this section provides instructions for storing Quantity and Amount metrics.

There are a variety of scenarios that occur when incorporating a new metric. Depending on where the data resides, you can take particular steps to incorporate the new metrics and make them available in your data warehouse and front-end schema. Generally speaking, there are three major areas where data resides—source database, staging table, and Siebel Customer-Centric Enterprise Warehouse. Figure 44 illustrates the three scenarios and the components that are affected when trying to store the new data.

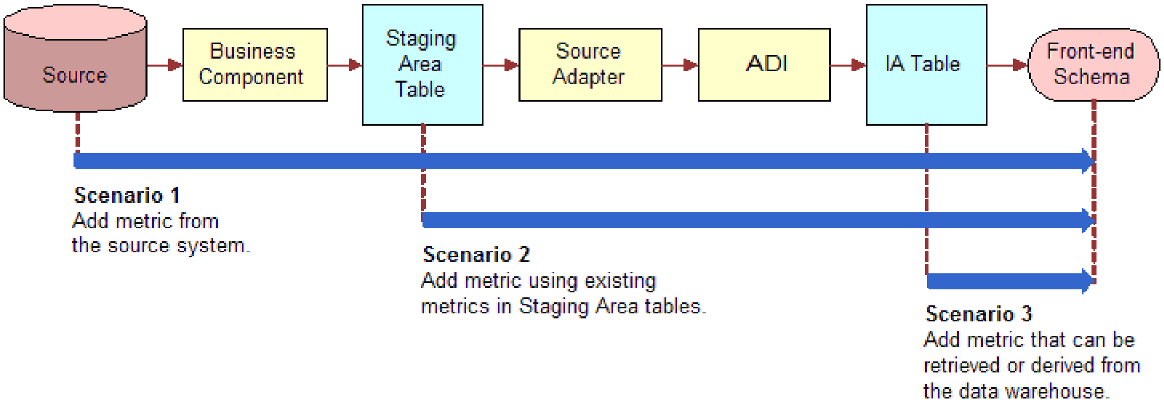

Source
Business Component
Staging Area Table
Source Adapter
ADI
IA Table
Front-end Schema
Scenario 1
Add metric from the source system.
Scenario 2
Add metric using existing metrics in Staging Area tables.
Scenario 3
Add metric that can be retrieved or derived from the data warehouse.

Figure 44. Scenarios for Storing New Metrics

In the sections that follow, you can find procedures for storing new metrics given each of these three scenarios. Each of the steps in the procedure correspond to larger topics discussed in later sections of this chapter. The steps provide you with a high-level overview, and the larger topics provide the details.

To store additional metrics from the source system

1 Modify the existing extract mapping or create a new extract mapping to extract this new information from the source system.

  For information on modifying the existing Business Component mapplet of an extract mapping, see Process of Creating and Modifying Business Adapters on page 213. For information on creating a new extract mapping, see Creating an Extract Mapping on page 203.

  TIP: You can perform calculations in the Business Component mapplet of the extract mapping or in the Source Adapter mapplet of the load mapping. However, it is not recommended that you perform large calculations in the extract. This protects you from interfering with your source transaction system. For these types of calculations, it is recommended that you perform them in the Source Adapter mapplet in the load mapping. One of the major reasons why Siebel Customer-Centric Enterprise Warehouse splits the extract process from the load process into two separate mappings is to minimize the amount of time tying up the source transaction system.

2 Modify the existing load mapping to take the data from the staging table and load it into the data warehouse.

For information on modifying the existing Source Adapter mapplet of a load mapping, see Process of Creating and Modifying Business Adapters on page 213. For information on creating a new load mapping, see Creating a Load Mapping on page 206.

TIP: If you map the data to a staging table's extension column, you must determine the type of extension column to use. For explicit information on the type of extension column to use for attribute data, see Determining the Type of Extension Column to Use for Metrics on page 184.

3 Incorporate the new metric into your front-end schema.

NOTE: You can also store additional metrics derived from other staging area tables or from data warehouse objects. The steps for these tasks are identical to those for storing new attributes. See the procedures in Storing Additional Attributes in the Data Warehouse on page 181.

Adding More Dates to the Data Model

There are a number of dates that are populated when you initialize Siebel Customer-Centric Enterprise Warehouse. For example, purchase orders use an ordered-on-date, a requested-date, a required-on-date, and so on. However, for date types that are not captured by the prepackaged extract and load processes, you can use extension date columns.

To map an additional date column to an extension date column

1 Map the additional date from the source into one of the extension columns in the staging area.

To do this, you must modify an extract mapping. Within the Business Component:

a Include the source table that supplies the date as a source table in the extract mapping.

b Map the data from the source table to the Source Qualifier transformation.

c Map the data from the Source Qualifier to the Mapplet Output Object (MAPO).

d Map the data from the Business Component to the Expression transformation. If your date is not in the proper date format, you can convert it using this Expression transformation.

e Map the data from the Expression transformation to the staging table.

After this is complete, the corresponding load mapping pulls the date data and loads it into the corresponding IA table.

2 You also must modify the front-end schema to report on this date.

About Integrating Data from Source Systems Without Prepackaged Business Adapters

Siebel Customer-Centric Enterprise Warehouse provides universal business adapters for source systems that have no prepackaged business adapter. These source systems are referred to as universal sources. The universal business adapter extracts, transforms, and loads data from these sources. However, unlike the prepackaged business adapters for particular source systems, the universal business adapters are prepackaged to perform general transformations, such as enabling Type II Slowly Changing Dimensions, date data type conversion logic, and deletion logic. The universal source architecture extracts and loads data from any source system. As a result, you may need to configure the data or prepackaged extract, or both, and load mappings to incorporate your unique transformation logic. For more information on the universal source data requirements, and an architectural overview of the ETL processes, see the following sections.

About Extracting Universal Source Data

All universal source data goes through a similar extract mapping that is made up of a flat file template, Source Qualifier, Expression transformation, and staging table. The basic extract mapping for universal source flat files is shown in Figure 45.

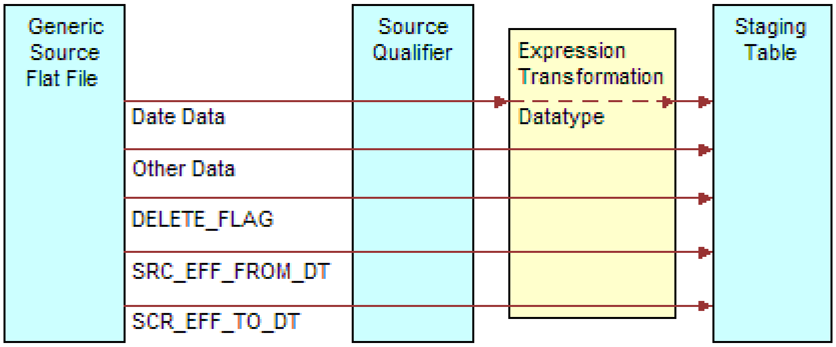


Figure 45. Basic Extract Mapping for Universal Source Data

NOTE: Unlike other adapters, the universal adapter requires values for the DELETE_FLAG, SRC_EFF_FROM_DT, and SRC_EFF_TO_DT.

Each of the extract mapping components is described in the following sections.

Universal Source Flat File Template

Given the vast differences in source data, Siebel Customer-Centric Enterprise Warehouse does not prepackage transformation logic other than date transformations. Therefore, there are a few requirements on how to present source data when using the universal ETL components. Siebel Customer-Centric Enterprise Warehouse provides a flat file template that delineates what data to provide, as well as the format in which the data must be presented. Each flat file template corresponds to exactly one data warehouse table. The following are the specifications for the source flat file:

- Supply all the universal source data in a comma delimited flat file (*.csv). Because flat files have only number and string data types, all dates must be provided in the String (14) data type. For example, you can use 200112310000 for December 31, 2001.
- Specify the values for the Key ID and Source ID in the flat file. Unlike prepackaged sources, these values are not formed in the Source Adapter for universal sources.
- Preset the record deletion flags. The delete flag can have the values Y or N.
- If applicable, supply the source Effective To and Effective From dates in the flat file. If the effective dates are not supplied, Siebel Customer-Centric Enterprise Warehouse inserts default values—January 01, 1899 for the Effective From date and January 01, 3714 for the Effective To date.
- Each flat file template has 10 system columns that are not used. These columns are named RESERVED_1, RESERVED_2, and so on. These are not customization columns for your use; Siebel Business Analytics reserves these columns for future development.

  NOTE: The IA_COPYRIGHT column is populated during an initial load (M_Z_INITIAL_LOAD) from one of the prepackaged CSV files.

Each of the extract mapping components performs the following functions:

- **Source Qualifier.** The Source Qualifier provides the means for extracting the data from the source.

  NOTE: You can use the Source Qualifier to join multiple source flat files from the same database platform. However, if you are sourcing from multiple sources that belong to different database platforms you cannot use a Source Qualifier; you must use a Joiner Transformation to join the sources.

- **Business Component.** The Business Component is omitted from the extract mapping for universal sources as there are no transformations to perform other than the date data type change. Therefore, you can either transform the data prior to it being extracted by the universal adapter or incorporate transformation logic within the universal adapter.
- **Expression Transformation.** As previously mentioned, the Expression transformation only converts dates with the string(14) data type into a timestamp data type, which is the standard data type used in the data warehouse tables.
- **Staging Table.** The staging tables are the target tables for the extract mappings. Each staging table mirrors its corresponding load control table structure. However, it does not have control columns, such as CURR_KEY, IA_INSERT_DT, IA_UPDATE_DT, and other CURR_* columns. If a load control table is not available, the staging table mirrors the IA table, except that instead of *_KEY and *_DT columns, it uses *_ID and *_DT columns.

After data is extracted, the data then goes through a load mapping, described in the following section.

Load Process for Universal Source Data

All universal source data goes through a similar load mapping that is made up of a staging table, Source Adapter mapplet, ADI mapplet, and data warehouse table. The basic load mapping for universal source flat files is shown in Figure 46.

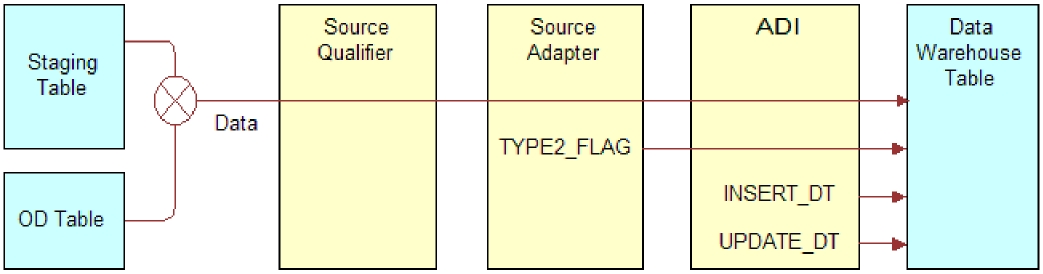


Figure 46. Basic Load Mapping for Universal Source Data

Each of the load mapping components perform as follows:

- **Staging Table.** The staging table serves as the source table for the load mapping. Each staging table mirrors its corresponding load control table structure. However, it does not have control columns, such as CURR_KEY, IA_INSERT_DT, IA_UPDATE_DT, and other CURR_* columns. If a load control table is not available, the staging table mirrors the IA table, except that instead of *_KEY and *_DT columns, it uses *_ID and *_DT columns.
- **OD Table.** The load control table determines if the source records loaded in the staging table are new, updated, or unchanged. New records are inserted into the data warehouse table. Updates to existing records overwrite the existing records if Type I Slowly Changing Dimensions are enabled. Updates to existing records are inserted if Type II Slowly Changing Dimensions are enabled. Unchanged records are rejected from the data warehouse because they already exist.
- **Source Qualifier.** The Source Qualifier joins the staging table and load control table and provides the Source ID and Key ID.
- **Source Adapter.** The Source Adapter mapplet adds the Type II Flag to the sourced data. If you want to incorporate additional data transformation logic, add it within this object.
- **ADI.** The ADI mapplet creates values for the INSERT_DT and UPDATE_DT columns. In addition, the ADI also performs the update strategy that either inserts new records, updates existing records, or rejects records from the load.
- **Data Warehouse Table.** The resulting data warehouse table (IA) stores the data for end user querying purposes. Although not illustrated in Figure 46, each time the IA table is loaded, the ADI also reloads the OD table after truncating it. Only the most recent snapshot of all column values are stored in OD tables; they do not store historical information.

11 Integrating Additional Data

This chapter provides procedural information for creating and modifying the ETL components to populate data in the data warehouse tables.

It contains the following topics:

Overview of Integrating Additional Data

As you perform gap analysis, you may find that you need to create new components to either extract, load, transform, or store additional data. This chapter discusses how to perform each of these tasks, including use of the table extension columns.

As you begin creating new components, you need to be consistent in the way you name your objects. For all custom-built objects, the naming convention may be prefixed with Z_. For example, if you create a new profile table for customers, then you might name it as Z_CUSTOMER_PROFILE. For a list of naming conventions used for objects prepackaged by Siebel Customer-Centric Enterprise Warehouse, see the *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.

Table Formats

You can create additional tables in Siebel Customer-Centric Enterprise Warehouse to meet your business requirements. This section lists the specific formats for various tables that are commonly used standards.

Although there are exceptions to the rules, these are the formats you need to follow when creating new tables. All tables, with the exception of new staging tables, need to be created in the Siebel Business Analytics folder in the PowerCenter repository. Only staging tables are created within the prepackaged, source-specific configuration folders, such as Configuration for Oracle 11i and Configuration for PeopleSoft.

The following topics discuss additional table formats you can create in Siebel Customer-Centric Enterprise Warehouse. They describe how to create an entirely new table, as well as how to make a new table using a copy of an existing table:

- Fact Table Format on page 192

Each of the table types in Siebel Customer-Centric Enterprise Warehouse has a specific format and naming convention that must be followed to create tables.

You can also create Profile tables with additional table formats. Profile tables are covered separately in the topic Creating a Profile Table Using Domain Values on page 202.

Fact Table Format

A fact table that is linked to other tables requires a surrogate key. If the fact table is used in other mappings the surrogate key becomes the primary key. If the table is not used in other mappings, the primary key is a composite of the KEY_ID and SOURCE_ID. The naming convention for fact tables is IA_[SUBJECT], for example, IA_AP_XACTS.

The following is the order of the columns, data type, and precision for fact tables:

1. **Surrogate key.** The data type is decimal (10,0) or (15,0) (for example, in the fact table for Accounts Payable transactions, IA_AP_XACTS, the surrogate key is AP_XACTS_KEY).

   NOTE: This column does not apply to all fact tables.

2. **Dimensions keys.** The data type is decimal (10,0) or (15,0) (for example, the GL_ACCOUNT_KEY in IA_AP_XACTS).
3. **Date Key columns.** The data type is decimal (15,0) in Julian format with a DK suffix (for example, CREATED_ON_DK is a date key in the IA_AP_XACTS fact table).
4. **Amount columns.** The data type is decimal (28,10) ordered by currency type and with an AMT suffix (for example, AP_GRP_AMT in IA_AP_XACTS).
5. **Quantity columns.** The data type is decimal (18,3) with a QTY suffix (for example, XACT_QTY).
6. **Code columns.** The data type of all codes is varchar (30) with a CODE suffix (for example, UOM_CODE). There are two kinds of codes:
   - The first code type contains units of measure and currency.
   - The second code type is code-name pairs where the code and the code name are both stored, such as states where there is both CA and California.
7. **Other fact attributes.** The data type of the attribute is subjective to the kind of attribute. For example, for ACCT_DOC_NUM, the data type is varchar (30), while for ACCT_DOC_ITEM the data type is decimal (15,0).
8. **Description columns.** The data type for the description columns is varchar (254) or varchar (255) with a DESC suffix (for example, GL_ACCOUNT_DESC).

9 **Name columns.** The data type of the name columns is varchar (254) or varchar (255) with a NAME suffix (for example, GL_ACCOUNT_NAME).

10 **Extension columns.** Extension columns have the same data type and precision as their column type (for example, amount, quantity, or code). The order for the extension columns is as follows:

- The dimensional key naming convention is [FACT TABLE ABBREVIATION]_DIM[SEQUENTIAL NUMBER]_KEY (for example, APXT_DIM1_KEY in IA_AP_XACTS).
- The format for data keys is [FACT TABLE NAME]_DATE[SEQUENTIAL NUMBER]_DK (for example, APXT_DATE1_DK in IA_AP_XACTS).
- The quantity naming convention is [FACT TABLE NAME]_[SEQUENTIAL NUMBER]_QTY (for example, APXT_1_QTY in IA_AP_XACTS).
- The amount naming convention is [FACT TABLE NAME]_DOC[SEQUENTIAL NUMBER]_AMT (for example, APXT_DOC1_AMT in IA_AP_XACTS).
- The code and name pair naming convention is [FACT TABLE NAME]_ATTR[SEQUENTIAL NUMBER]_CODE and [FACT TABLE NAME]_ATTR[SEQUENTIAL NUMBER]_NAME (for example, APXT_ATTR1_CODE and APXT_ATTR1_NAME in IA_AP_XACTS).
- The text column naming convention is [FACT TABLE NAME]_ATTR[SEQUENTIAL NUMBER]_TEXT (for example, APXT_ATTR1_TEXT in IA_AP_XACTS). (The data type for text columns is varchar (254).)

11 **Control columns.** The data type and precision of the control columns varies. KEY_ID is data type varchar (80), SOURCE_ID is data type varchar (30), and IA_COPYRIGHT is data type varchar (254). IA_INSERT_DT and IA_UPDATE_DT are dependent on the database; datetime (26, 6) for SQL Server; date (26,6) for Oracle; and timestamp (26,6) for DB2.

Load Control Table Format for Fact Tables

There are different load control table formats for both fact tables and dimension tables. This section focuses specifically on load control tables for fact tables. The first column in a load control table is the Current Key (CURR_KEY). It is the Current Key that links the load control table with the surrogate key in the fact table. The naming convention for a load control table is OD_[SUBJECT], for example, OD_AP_XACTS.

The following is the order of the columns, data type, and precision for load control tables for facts:

1 **Current key (CURR_KEY).** The data type for the current key is decimal (15, 0).

2 **ID columns.** The data type for ID columns is varchar (80), with a suffix of ID, which corresponds to the *_KEY columns in the fact (IA) table (for example, SALES_ORDLN_ID in the OD_SALES_PCKLNS load control table corresponds to the SALES_ORDLN_KEY).

NOTE: These columns only exist in an OD_* load control table if the corresponding IA_* warehouse table contains a surrogate key. For example, the IA_SALES_ORDLNS warehouse table uses a surrogate key for SALES_ORDLNS_KEY column; therefore, OD_SALES_ORDLNS table has a SALES_ORDLN_ID column.

3 **Date columns.** The data type of date columns is datetime (26, 6) for SQL Server; date (26,6) for Oracle; and timestamp (26,6) for DB2, with a suffix of DT (for example, CREATED_ON_DT). This column corresponds to the *_DK columns in IA fact tables.

4 **Amount columns.** The data type for the amount columns is decimal (28, 10) with a suffix of AMT (for example, NET_GRP_AMT). As with the amount columns in the fact table, they are in order of currency—group, local, and document.

5 **Codes.** The data type of all codes is varchar (30), with a suffix of CODE (for example, UOM_CODE). The code columns correspond with the fact table, therefore if the fact table has both units of measure and currency as well as code-name pairs, the load control table has these values too.

6 **Other attributes.** The data type varies to suit the attribute, but corresponds to what is found in the fact table.

7 **Extension columns.** As found in the fact table and in the same order:

- Dimensional IDs with a format of [FACT_TABLE_NAME]_DIM[SEQUENTIAL_NUMBER]_ID, such as APXT_DIM1_ID.
- Date with a format of [FACT_TABLE_NAME]_DATE[SEQUENTIAL_NUMBER]_DT, such as APXT_DATE1_DT.
- Quantity columns with a format of [FACT_TABLE_NAME]_[SEQUENTIAL_NUMBER]_QTY, such as APXT_1_QTY. Amount columns with a format of [FACT_TABLE_NAME]_DOC[SEQUENTIAL_NUMBER]_AMT, such as APXT_DOC1_AMT.
- Code and Name pairs with a format of [FACT_TABLE_NAME]_ATTR[SEQUENTIAL_NUMBER]_CODE, or [FACT_TABLE_NAME]_ATTR[SEQUENTIAL_NUMBER]_NAME, such as APXT_ATTR1_CODE and APXT_ATTR1_NAME.
- Text columns with a format of [FACT_TABLE_NAME]_ATTR[SEQUENTIAL_NUMBER]_TEXT, such as APXT_ATTR1_TEXT. The data type for text columns is varchar (254).

8 **Control columns.** The data type and precision of the control columns varies. KEY_ID is varchar (80), SOURCE_ID is data type varchar (30), and IA_COPYRIGHT is data type varchar (254). IA_INSERT_DT and IA_UPDATE_DT are dependent on the database; datetime (26, 6) for SQL Server; date (26,6) for Oracle; and timestamp (26,6) for DB2.

Dimension Table Format

Dimension tables always start off with a surrogate key. The surrogate key is always the primary key for the table. The naming convention for dimension tables is the same as fact tables, IA_[SUBJECT], for example, IA_PRODUCTS.

The following is the order of the columns, data type, and precision for dimension tables:

1 **Surrogate key.** The data type is decimal (10, 0) or decimal (15, 0) (for example, PRODUCT_KEY).

2 **Dimension keys.** The data type for the dimension key is decimal (15,0) (for example, the VISITOR_KEY in the IA_CUSTOMERS dimension table).

3. **Date keys**. The data type is decimal (15, 0) in Julian format and the suffix DK (for example, CREATED_ON_DK).
4. **Attribute columns.** Attribute columns can be of several different types, including descriptive and code, and the data type varies depending on the type. If it is a descriptive type such as name, the data type is varchar (254); if it is a code, it is varchar (30); all others, including number types, are either varchar (80) or varchar (30).
5. **Hierarchy columns.** The data type for codes is varchar (30), and for names the data type is varchar (254). The number of the hierarchy columns is based on the number of nodes in the hierarchy and there is no specific limit. Hierarchy columns are entered in code or name pairs, with the naming convention of [DIMENSION TABLE ABBREVIATION]_HIER[SEQUENTIAL NUMBER]_CODE, or [DIMENSION TABLE ABBREVIATION]_HIER[SEQUENTIAL NUMBER]_NAME. Each pair uses the same number, corresponding to its level in the hierarchy, for example, PROD_HIER1_CODE and PROD_HIER1_NAME in the IA_PRODUCTS dimension table are at the first level.

   NOTE: This column is only applicable to a few dimension tables.

6. **Code and Description columns.** The data type for codes is varchar (30), and the data type for descriptions is varchar (254), with respective CODE and DESC suffixes (for example, DIVISION_CODE and DIVISION_DESC).
7. **Extension columns.** There are principally two kinds of extension columns in the following order:
   - Extension columns for additional code or name pair attributes:
     - Naming conventions of [DIMENSION TABLE ABBREVIATION]_ATTR[SEQUENTIAL NUMBER]_CODE and [DIMENSION TABLE ABBREVIATION]_ATTR[SEQUENTIAL NUMBER]_NAME.
     - Data types varchar (30) and (254).
     - CODE and NAME suffixes (for example, PROD_ATTR1_CODE and PROD_ATTR1_NAME).
   - Extension columns for additional textual information have a naming convention of [DIMENSION TABLE ABBREVIATION]_ATTR[SEQUENTIAL NUMBER]_TEXT, with data type varchar (254), and TEXT suffix (for example, PROD_ATTR1_TEXT and PROD_ATTR1_NAME).
8. **Control columns.** Control columns include CURRENT_FLAG and DELETE_FLAG varchar (1), KEY_ID varchar (80), SOURCE_ID varchar (30), and IA_COPYRIGHT varchar (254), IA_INSERT_DT, IA_UPDATE_DT, EFFECTIVE_FROM_DT and EFFECTIVE_TO_DT.where EFFECTIVE_FROM_DT and EFFECTIVE_TO_DT are added to handle slowly changing dimensions and are dependent on the database; datetime (26, 6) for SQL Server; date (26,6) for Oracle; and timestamp (26,6) for DB2.

Load Control Table Format for Dimension Tables

The first key is the current key, which points to the surrogate key in the dimension IA table. The naming convention for load control tables is OD_[SUBJECT], for example, OD_GL_ACCOUNTS.

The following is the order of the columns, data type, and precision for load control tables for dimension tables:

1 **Current key (CURR_KEY).** The data type for the current key is decimal (15, 0).

2 **Dimension ID keys.** The data type for dimension ID keys corresponding with the IA dimension table is varchar (80).

3 **Date.** The data type of date columns is datetime (26, 6) for SQL Server; date (26,6) for Oracle; and timestamp (26,6) for DB2, with a suffix of DT (for example, CREATED_ON_DT). This column corresponds to the *_DK columns in IA fact tables.

4 **Other attributes.** The data type, length and precision is as found in the corresponding dimension table (IA table).

5 **Hierarchy columns.** The data type for codes is varchar (30,0) and for name the data type is varchar (254). The sequential number of the hierarchy column matches those code or name pairs found in the IA dimension table.

6 **Code and Description columns.** The data type for codes is varchar (30) and for descriptions the data type is varchar (254). The code-name pairs are found in the corresponding IA dimension tables.

7 **Extension columns.** Extension columns corresponds with those found in the IA dimension table.

8 **Control columns.** The data types correspond with those control columns found in the dimension table, including IA_COPYRIGHT and a composite primary key composed of the KEY_ID, SOURCE_ID, and SRC_EFF_FROM_DT. IA_INSERT_DT and IA_UPDATE_DT are dependent on the database; datetime (26, 6) for SQL Server; date (26,6) for Oracle; and timestamp (26,6) for DB2.

    NOTE: The SRC_EFF_FROM_DT column is not available in all dimension tables.

Aggregate Table Format

Aggregate tables aggregate facts across a dimension or set of dimensions. You can have several aggregate tables linking to the same tables, but aggregating data for different time periods. for example, there are three contact center aggregate tables (IA_CC_REP_A1, IA_CC_REP_A2, and IA_CC_REP_A3) that aggregate various contact statistics by hour, day, and month respectively.

The first keys in an aggregate table are the Dimension keys that link the table to the dimension tables for which it is aggregating data. As illustrated in the IA_CC_REP_A1 example, the naming convention for aggregate tables is IA_[SUBJECT]_A[SEQUENTIAL NUMBER].

The following is the order of the columns, data type, and precision for IA aggregate tables.

1 **Dimension keys.** The data type is decimal (10, 0) or decimal (15, 0). Dimension keys link the aggregate table to the dimension tables that could be used to analyze the aggregate table. For example, SUPERVISOR_KEY links IA_CC_REP_A1 with the Employee dimension table. There can be several dimension keys drawing information from several different dimension tables.

2 **Date columns.** The data type is decimal (10,0) or decimal (15,0), with a DK suffix (for example, PERIOD_START_DK).

3 **Amount columns.** The data type is decimal (28,10), with an AMT suffix (for example, TRUNK_COST_AMT).

4 **Quantity columns.** The data type of (18,3), with QTY suffix (for example, PRA1_1_QTY).

5. **Currency Code columns**. The data type is varchar (30), with a code suffix (for example, LOC_CURR_CODE).
6. **Extension columns.** Format for Extension columns varies and is as follows:
   - Dimensional keys with a format of [FACT TABLE NAME]_DIM[SEQUENTIAL NUMBER]_KEY, such as APXT_DIM1_KEY.
   - Date keys with a format of [FACT TABLE NAME]_DATE[SEQUENTIAL NUMBER]_DK, such as APXT_DATE1_DK.
   - Quantity columns with a format of [FACT TABLE NAME]_[SEQUENTIAL NUMBER]_QTY, such as APXT_1_QTY.
   - Amount columns with a format of [FACT TABLE NAME]_DOC[SEQUENTIAL NUMBER]_AMT, such as APXT_DOC1_AMT.
   - Code and Name pairs with a format of [FACT TABLE NAME]_ATTR[SEQUENTIAL NUMBER]_CODE, or [FACT TABLE NAME]_ATTR[SEQUENTIAL NUMBER]_NAME, such as APXT_ATTR1_CODE and APXT_ATTR1_NAME.
   - Text columns with a format of [FACT TABLE NAME]_ATTR[SEQUENTIAL NUMBER]_TEXT, such as APXT_ATTR1_TEXT.

   NOTE: The data type for text columns is varchar (254).
7. **Control columns.** Control keys include CREATED_BY_KEY, CHANGED_BY_KEY, CREATED_ON_DK, CHANGED_ON_DK, DELETE_FLAG, IA_COPYRIGHT, and a composite primary key containing the KEY_ID and SOURCE_ID. IA_INSERT_DT and IA_UPDATE_DT are dependent on the database; datetime (26, 6) for SQL Server; date (26,6) for Oracle; and timestamp (26,6) for DB2.

Incremental Table Format

Incremental tables follow the same exact format of the corresponding fact or load control table from which they source information. Therefore, to create an incremental table, you must first create or locate the fact or load control table that is being used as the source. The fact or load control table can then be copied and renamed, as shown in Using Existing Tables to Create New Tables on page 200. Set the session parameter so that the table gets truncated before each load. The naming convention for incremental tables is NU_[SUBJECT], for example, NU_GL_ACTIVITY.

Staging Table Format

Unlike the other tables that are created in the Siebel Business Analytics folder, the staging tables are source-dependent and are created in the prepackaged source-specific configuration folders in the PowerCenter Repository. These tables have the following requirements:

- The default database for staging tables is DB2, however you may change this value as necessary. The primary key for staging tables is a composite of the KEY_ID and SOURCE_ID.
- The naming convention for staging tables is T[SOURCE]_[NAME].

- For Oracle 11i, the staging table is TI_USERS.

The purpose of the staging tables is to speed data extraction by temporarily storing extracted source data that are transformed in an Expression transformation in preparation for the source-independent Siebel Customer-Centric Enterprise Warehouse. Therefore, the order of the columns, and contents is determined by the corresponding source data.

For example, in the staging table TO_USERS, the first eight columns contain attributes about user information, some of which include CREATED_BY, LAST_UPDATED_BY, USER_NAME, and EMAIL_ADDR, and so on, and which are then followed by extension columns and the last two control columns that comprise the primary key, the KEY_ID and the SOURCE_ID.

In contrast, TS_USERS, which is the staging table for user information in SAP, has several columns containing attribute information about users including DEPARTMENT_CODE, DELEGATEE_NAME, REGION_CODE, and LANGUAGE_CODE. The only significant similarities are that the attributes are followed by extension columns and that the primary key is usually a composition of KEY_ID and SOURCE_ID columns. Sometimes, however, the primary key also includes the effective date for the record.

Creating New Tables

After reviewing the required format and naming convention for the tables you want to create, you can create the actual table itself. The process of creating the table is the same regardless of the kind of tables you want to create. The important difference is the naming convention and to maintain their respective formats. The following procedure provides information on creating a target table. If the table is used as a source as well, you can copy the table into the Sources folder and modify its properties as necessary.

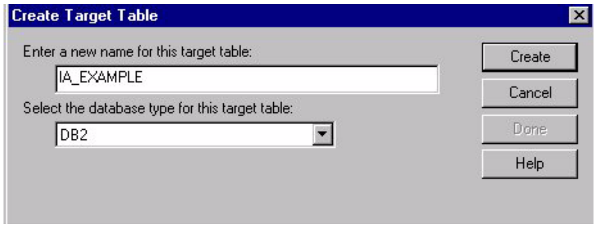
To create new target tables

1. Open PowerCenter repository and select the Siebel Business Analytics folder, if you are creating any kind of table except a staging table.

   If you are creating a staging table, open the corresponding prepackaged source specific configuration folder. The rest of the procedure remains the same.

2. Select Tools > Warehouse Designer.
3. Select Targets > Create to open the Create Target Table window.
4. Following the appropriate naming convention for the type of table, enter a new name for the table.

5 Select the database type from the list window and click Create, as shown in the following figure.



Enter a new name for this target table: IA_EXAMPLE

Select the database type for this target table: DB2

NOTE: The required database type for all tables is DB2.

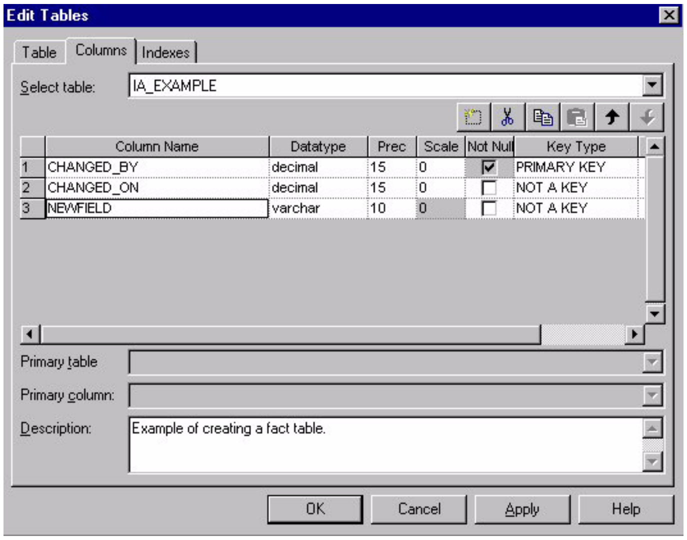
6 When the table appears in the Warehouse Designer, click Done to close the Create Target Table window.

7 Double-click the newly created table to open the Edit Table window.

8 Enter a description in the Description window on the Table tab.

9 Click the Columns tab to add columns to the table.

Click the Add Column icon and add columns, data type, and set precision as required, as shown in the following figure.



Select table: IA_EXAMPLE

	Column Name	Datatype	Prec	Scale	Not Null	Key Type
1	CHANGED_BY	decimal	15	0	☑	PRIMARY KEY
2	CHANGED_ON	decimal	15	0	□	NOT A KEY
3	NEWFIELD	varchar	10	0	□	NOT A KEY

Primary table:

Primary column:

Description: Example of creating a fact table.

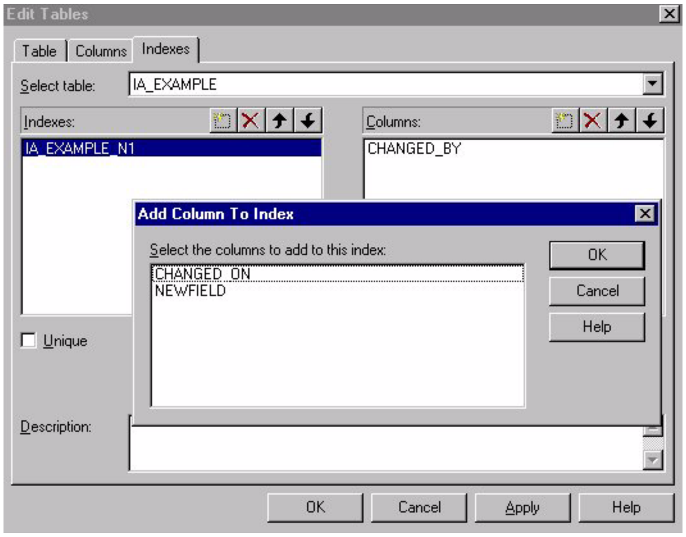
10 Click the Indexes tab, and then click the New Insert button to enter the table name with the appropriate index suffix in the Indexes window.

NOTE: The format for the index is [TABLE NAME]_N[SEQUENTIAL NUMBER] for nonunique indexes, or [TABLE NAME]_U[SEQUENTIAL NUMBER] for unique indexes. For example, IA_EXAMPLE_N1 or IA_EXAMPLE_U1. Although indexes are not required, they help speed processing time by connecting nonunique tables.

11 Click the New Insert button in the Columns window to open the Add Column to Index dialog box:

- Highlight the column from the list that you want the index to look locate, and click OK.

- You must repeat clicking the New Insert button, highlight the appropriate column, and then click OK for each column you want to add, as shown in the following figure.



Select table: IA_EXAMPLE

Indexes: IA_EXAMPLE_N1

Columns: CHANGED_BY

Select the columns to add to this index:
CHANGED_ON
NEWFIELD

12 If the index is unique, select the Unique check box.

Do not check the box for nonunique indexes.

13 Click OK to return to the Warehouse Designer.

The new table is automatically added to the repository. Save changes before exiting PowerCenter Designer.

Using Existing Tables to Create New Tables

You can use existing tables in Siebel Customer-Centric Enterprise Warehouse to create new tables by making a copy of the table that most closely corresponds with your needs. This is the simplest way to make sure the table format is correct, and it saves time if some of the required columns are already present. The following procedure provides information on creating a target table. If the table is used as a source as well, you can copy the table into the Sources folder and modify its properties as necessary.

To create new target tables by copying existing tables

1 In PowerCenter Designer, open the Siebel Business Analytics folder, unless you are making a new staging table, in which case select the specific source configuration folder.

2 On the Tools menu, click Warehouse Designer.

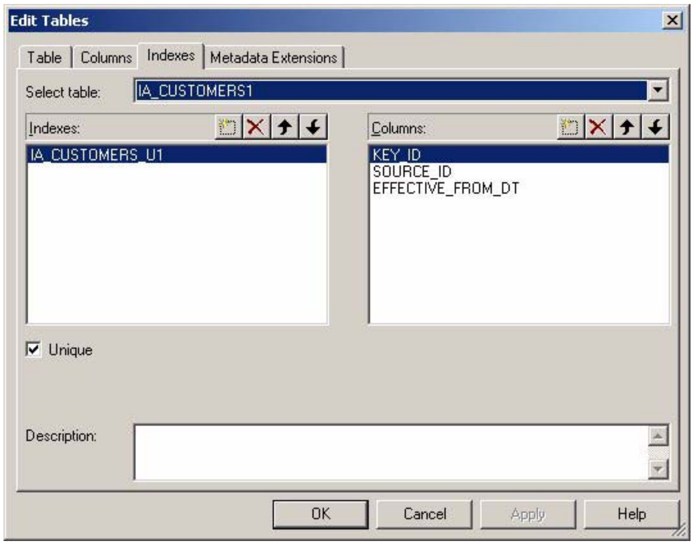
3 Open the Targets folder, copy the table most closely representing the table you want to create, and drag it into the Warehouse Designer.

- If you use the copy and paste functions in the Edit menu and you paste them back into the Siebel Business Analytics folder, you are prompted to rename the table because it already exists.
- By default all tables in Siebel Customer-Centric Enterprise Warehouse are created with a database type of DB2. If you want to select a different database type, select the Database Type list window and select from the list.

4. Open the new table in Warehouse Designer to edit its properties.
5. Click the Columns tab and add, delete, or modify columns as necessary.

   Be sure to conform column to format for data type, length, and precision.
6. Click the Indexes tab, and then click the New Insert button to enter the table name with the appropriate index suffix.

   NOTE: The format for the index is [TABLE_NAME]_N[SEQUENTIAL_NUMBER] for nonunique indexes or [TABLE_NAME]_U[SEQUENTIAL_NUMBER] for unique indexes. For example, IA_BANK_LOCAL_N1 or IA_BANK_LOCAL_U1. Although indexes are not required, they help speed processing time by connecting the nonunique tables.
7. Click the New Insert button in the Columns window to open the Add Column to Index dialog box.
   - Highlight the column from the list that you want the index to look for and click OK.
   - You must repeat clicking the New Insert button, highlighting the appropriate column and then clicking OK for each column you want to add, as shown in the following figure.



Select table: IA_CUSTOMERS1

Indexes:
IA_CUSTOMERS_U1

Columns:
KEY_ID
SOURCE_ID
EFFECTIVE_FROM_DT

☑ Unique

Description:

8. If the index is unique, select the Unique check box.

   Do not check the box for nonunique tables.
9. Click OK to return to the Warehouse Designer.

   The new table is automatically added to the repository. Save changes before exiting PowerCenter Designer.

Creating a Profile Table Using Domain Values

This section describes how to create new profile tables. A *profile table* is a table that holds all possible combinations of specific multistate columns. You can create an unlimited number of profile tables, containing all possible combinations of domains and their values or the cross-product of the contents of IA_DOMAINS.

A profile table is built with the help of two prepackaged tables—the Domains table and the Profile Specifications table.

- **The Domains table**. IA_DOMAINS, stores predefined values for columns that can only hold certain values, as shown in Table 28.
- **The Profile Specifications table**. IA_PROFILE_SPECS, is used to join to the Domains table to generate all the possible combinations of values you may want to use in a profile table, as shown in Table 29.

Table 28. Join IA_DOMAINS

Domain Name	Domain Position	Domain Value	Domain Flag	Language Code
REGN	0	Not Performed	Y	E
REGN	1	Started	N	E
REGN	2	Completed	N	E
RESEARCH	0	Not Performed	Y	E
RESEARCH	1	Started	N	E

Table 29. Join IA_PROFILE_SPECS

PROFILE NAME	Language Code	Domain Name - 1	Domain Name - 2	Domain Name - 3
WEB_SESS_CTXT	E	REGN	RESEARCH	E
NEW PROFILE	E	Domain-1	Domain-2	E

You can create profile tables using the already loaded data in IA_DOMAINS and IA_PROFILE_SPECS. For this purpose, Siebel Customer-Centric Enterprise Warehouse includes a Business Component mapplet, which you can replicate and configure for every profile table you want to build. The Business Component joins up to ten instances of the IA_DOMAINS table to IA_PROFILE_SPECS using the DOMAIN_NAME column.

NOTE: You can build a profile table with a maximum of ten domains, if the first domain holds no more than three values and the other nine hold no more than ten each. If you build a profile table with fewer than ten domains, each domain can hold a maximum of ten values.

To build a new profile table using IA_DOMAINS and IA_PROFILE_SPECS

1. In PowerCenter Designer, open the Siebel Business Analytics folder.
2. Select Tools > Mapplet Designer.
3. Open Mapplets folder, and drag and drop MPLT_BCZ_PROFILE mapplet into Mapplet Designer.
4. Double-click the SQ_PROFILE Source Qualifier to open the Edit Transformations window.
5. Click the Properties tab, and then click the down arrow to access the SQL statement.
6. Edit the SQL statement by entering your new profile table name (for example, IA_SALES_PROFILE) in the PROFILE_NAME column and the Domain Names that comprise it in the DOMAIN_NAME_* columns.
7. Select Generate SQL, and then click OK.
8. Validate and save your changes to the repository.

   NOTE: Fact tables in Siebel Enterprise Sales Analytics and Siebel Enterprise Contact Center Analytics contain foreign keys to profile tables. For all other tables that do not contain a profile key, use one of the dimension key extension columns packaged in the table.

Creating New Mappings

This section describes the following configuration procedures:

- Creating an Extract Mapping on page 203
- Creating a Load Mapping on page 206
- Creating a Codes Mapping on page 210
- Creating a Derive Mapping on page 212

Creating an Extract Mapping

Extract mappings are used to extract source data and store it in the staging table. They usually contain the following components:

- Business Component mapplet
- Expression transformations
- Staging tables

NOTE: Universal Source extract mappings do not contain business components. For information on Universal Source mappings, see About Integrating Data from Source Systems Without Prepackaged Business Adapters on page 187.

About the Basic Structure of an Extract Mapping

M_I_PURCH_ORDERS_EXTRACT, shown in Figure 47, is an example of a basic extract mapping.

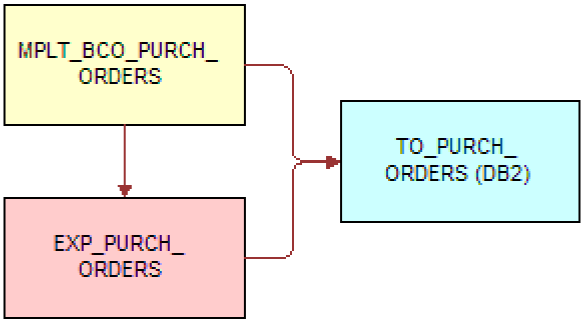


Figure 47. Basic Structure of an Extract Mapping

The Business Component mapplet extracts data from the source tables. For more information on creating a new Business Component, see the discussion on creating a new Business Component in Process of Creating and Modifying Business Adapters on page 213. After the Business Component mapplet extracts the source data, it then passes the data to at least one Expression transformation, which configures the Source ID and the Key ID. The data can pass through as many different types of Expression transformations as necessary to transform the data into a usable state.

After the transformations occur, the data is passed to the staging area target table. When creating a new extract mapping, create a new staging table as well; do not use any existing staging tables as it may impact performance of other mappings using that same table. For information on creating a new staging table, see the discussion on the staging table format in Staging Table Format on page 197. For information on working with extension columns, see Overview of Integrating Additional Data on page 191.

About Integrating Data from Source Systems Without Prepackaged Business Adapters

If you are adding generic source data—that is data that is extracted from source systems other than the prepackaged sources, such as Oracle, PeopleSoft, and SAP—then you do not need a Business Component mapplet in the extract mapping. You can omit this step, because generic sources must be in a ready-to-load state and do not need business components. Transformations are not prepackaged in the Universal Adapter mappings that extract, transform, and load this type of data. For more information on adding generic source data, see About Integrating Data from Source Systems Without Prepackaged Business Adapters on page 187.

NOTE: If a new staging table is created, you must create a new load mapping to move the data into the data warehouse. For information on creating a new load mapping, see Creating a Load Mapping on page 206. If an existing staging table's extension columns are used, the load mapping is prepackaged to move any data from the Staging table's extension columns into the Siebel Customer-Centric Enterprise Warehouse's extension columns.

To create an extract mapping

1. In PowerCenter Designer, open the applicable source configuration folder.

2 Select Tools > Mapping Designer.

3 Select Mappings > Create to create the mapping.

4 Enter the mapping name.

For a list of naming conventions, see the *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.

5 From the Repository Navigator, open the mapplet folder and drag the Business Component mapplet you require into Mapping Designer.

6 Open the transformation folder, and drag the reusable Expression transformation EXP_SOURCE_ID_FORMATION into Mapping Designer.

NOTE: If no reusable Expression transformation is available, proceed to Step 10.

7 Double-click on the Expression transformation to open the Edit Transformations window, and select Rename.

a Enter a new name for the transformation and click OK.

The naming convention for Expression transformations is EXP_[SUBJECT], where Subject is the subject that identifies the entity such as Suppliers or Customers.

b Enter a Description of the new Expression transformation for future reference, and click OK.

8 Drag the output ports from the Business Component mapplet to the Expression transformation input ports to connect them.

9 Create a KEY_ID column in the Expression transformation, and then:

a On the Ports tab, edit the definition of the KEY_ID.

The Key ID uniquely identifies records within a source. The KEY_ID port has a data type of string(80). This port is an output only port—select output (O) flag.

b In the Expression column, enter the definition for the Key ID.

You must include all columns that make the records unique.

NOTE: While forming the Key ID, do explicit type conversions inside the Expression Editor box for those ports that are not of string data type. After you create your new KEY_ID port, you may view it in the Expression transformation box.

TIP: It is recommended you keep the columns having the greatest number of different values first and the columns with the least distinct values last, separating each column name with a tilde (~).

10 If a reusable transformation was available as defined in Step 9, continue directly to Step 12.

a If a reusable transformation is not available to provide a Source ID, you must create a new one by selecting Transformation > Create.

b Select Expression from the Select Transformation Type list window.

Enter a new name for the transformation and select Create, then Done.

c Drag the output ports from Copy the Business Component mapplet to the Expression transformation input ports to connect them.

11 Double-click the transformation to open the Edit Transformation window and create the Source ID.

NOTE: In addition to creating a Key ID, you must also create the Source ID.

a Edit the Expression transformation to add another column called SOURCE_ID, with data type of string, 30.

b On the Ports tab, edit the definition of the SOURCE_ID to an output port only.

c Enter an abbreviation for the source.

The abbreviations for preconfigured source systems are shown in the following table.

Source	Source Abbreviation for SOURCE_ID
Oracle 11i	OAP11i
PeopleSoft	PSFT75
SAP R/3	SAPR3

NOTE: You may change the default Source ID if you use multiple instances of the same source. For example, you may be running multiple instances of SAP and, therefore, want separate Source IDs for each instance. For example, you might call the first instance SAPR3_1, and the second instance SAPR3_2, and so on.

After you create your new SOURCE_ID port, you may view it in the Expression transformation box.

12 Open the Target folder, and drag and drop the staging table into Mapping Designer.

13 Connect the ports from the Expression transformation to the staging area target table.

The target table may have extra ports or extension columns created for later customization. Leave these empty if there is no corresponding port coming in from the Expression transformation.

14 Validate the mapping and save your changes to the repository.

15 In PowerCenter Workflow Manager, you must now create a session for this mapping.

Be sure to set the target load properties to truncate the staging table by selecting the Truncate check box.

Creating a Load Mapping

Load mappings transform data from a staging table and load it into the IA and OD target tables. They usually contain the following components:

- Staging table
- Shortcut to a load control area (OD_*) table or IA_* table
- Source Adapter mapplet

- Shortcut to ADI mapplet
- Shortcut to IA and OD target tables

The staging table provides temporary storage for extracted source data. The load mapping compares extracted source data in the staging table against the load control table for updates to existing data.

If there is no load control table, use a shortcut to the IA table. The IA table and the staging table pass the data to the Source Adapter. The Source Adapter contains most of the business logic used in preparing the data for the Analytic Data Interface (ADI). The Source Adapter uses an Expression transformation to handle data type conversions, source-specific lookups, and to create control columns used in loading.

The load mapping prepares data for the ADI, matching its output with the input of the ADI mapplet. The ADI mapplet contains source-independent transformation logic, such as:

- Lookups to resolve keys and codes for dimension table loads
- General calculations, such as currency conversions for fact table loads

The mapping, M_I_PURCH_ORDERS_LOAD, which is shown in Figure 48, provides a sample of a typical fact load mapping.

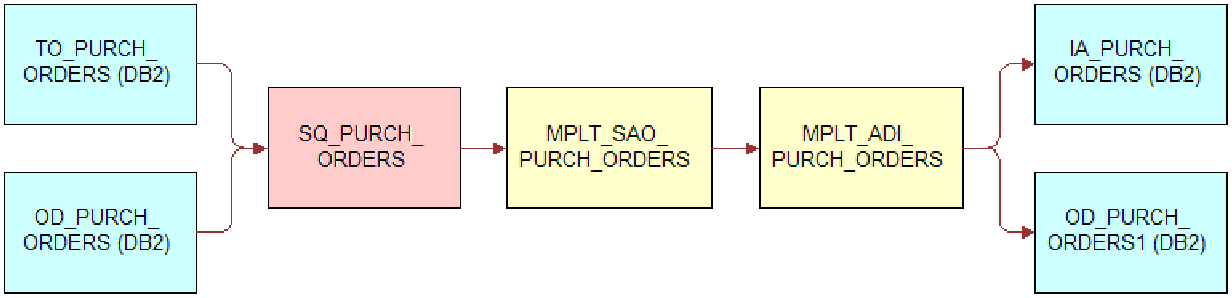


Figure 48. Basic Structure of a Fact Load Mapping

The mapping, M_I_CUSTOMRS_LOAD, which is shown in Figure 49, provides a sample of a typical dimension load mapping.

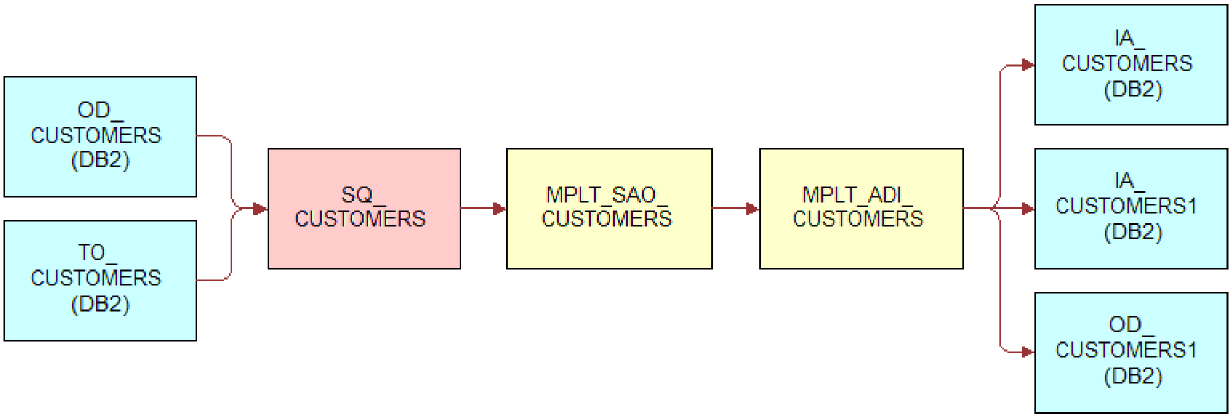


Figure 49. Basic Structure of a Dimension Load Mapping

To create a load mapping

1 In PowerCenter Designer, open the applicable source configuration folder.

2 Select Tools > Mapping Designer.

3 Select Mappings > Create to create the mapping.

4 Enter the mapping name, and click OK.

   For a list of naming conventions, see the *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.

5 From the Repository Navigator window, drag and drop the following items into Mapping Designer:

   - Staging table founds in IA_[SOURCE]_STAGE. For example, IA_ORA_STAGE for Oracle Applications.
   - Shortcut to Control table (OD table or IA table) found in IA_[SOURCE]_CONTROL. For example, IA_ORA_CONTROL for Oracle Applications.
   - Source Adapter mapplet from the Mapplet folder. For example, MPLT_SAO_AR_XACTS in Oracle Applications.
   - Shortcut to ADI mapplet from the Mapplet folder. For example, MPLT_ADI_AR_XACTS.
   - Shortcut to IA and OD target tables from the Target folder. For example, IA_AR_ACTIVITY.

   NOTE: If you are going to require Type II slowly changing dimension support, you must import two instances of the IA target table into your dimension load mappings. For more information on Type II slowly changing dimension, see Type I and Type II Slowly Changing Dimensions on page 131.

6 Delete any unnecessary Source Qualifier transformations.

If set up to do so, PowerCenter Designer creates separate Source Qualifier transformations for each source table you drag into a mapping.

For example, if you drag in a staging table and a control table (or even multiple instances of the same table) as the sources for your mapping, you have one Source Qualifier for each of them. Because only one Source Qualifier is required, delete all other Source Qualifier transformations.

7 After deleting any unnecessary Source Qualifier transformations, connect all ports to the one remaining Source Qualifier:

NOTE: In the previous example, the control table ports and the staging table ports are both connected to the same Source Qualifier transformation.

a Connect the Source Qualifier output ports to the Source Adapter mapplet input ports.

b Connect the Source Adapter mapplet output ports to the ADI mapplet input ports.

c Connect the first set of ADI mapplet output ports, classified under the MAPO_[SUBJECT]_IA1 heading, to one of the IA table instances.

NOTE: If you require only Type I changing dimension support, there is only one instance of the IA table and you can move directly to Step 9. For more information on Type I slowly changing dimension, see Type I and Type II Slowly Changing Dimensions on page 131.

8 If it is a dimension load mapping with Type II support, connect the second set of mapplet output ports, classified under the MAPO_[SUBJECT]_IA2 heading, to the second instance of the IA table.

This set is different from the main set of ports for the IA table; it only has the surrogate key and the control ports.

9 Connect the third set of mapplet output ports, classified under the MAPO_[SUBJECT]_OD heading, to the control table. If the dimension does not have a control table, then you only need to connect to the IA tables.

10 Double-click the Source Qualifier to open the Edit Transformations box.

You need to modify the User Defined Join and the SQL Query ports.

11 Click the Properties tab, and then click the small arrow in the Value column by User Defined Join to open the SQL Editor.

12 Edit the SQL statement for the User Defined Join port, as shown in the following figure, and enter the join condition between the source tables in this port.

Generally, the join is based only on the KEY_ID and SOURCE_ID. (Sometimes, SRC_EFF_FROM_DT, in addition to the KEY_ID and SOURCE_ID, is used in the join condition.) The join condition is an outer join between the staging area and the OD table. All staging area records are selected, whether or not they are present in the OD table.

13 Click the small arrow in the Value column by SQL Query to open the SQL Editor.

14 Edit the SQL Query statement.

Generate the SQL statement by selecting the Generate SQL button.

15 Validate the query by clicking the Validate button.

TIP: Before validating the query, you must remove the WHERE clause from the SQL statement, and temporarily store the text until you validate the SQL statement. PowerCenter uses a database-independent outer join syntax, and the WHERE clause is one of the database independent outer join syntaxes not understood by the SQL statement. After validation, replace the WHERE clause.

16 Save your changes to the repository.

17 In PowerCenter Workflow Manager, you must now create a session for this mapping.

For information on how to resolve dimension keys, see About Resolving Dimension Keys on page 149.

Creating a Codes Mapping

To write new codes to the IA_CODES table, you need to design a codes mapping to extract data from the source and load it into the IA_CODES table. Codes mappings have the following five objects, as shown in Figure 50:

- Business Component mapplet (MPLT_BC[SOURCE_VARIABLE]_CODES_[SOURCE])
- Expression transformation (EXP_CODES_[SUBJECT])
- Source Adapter mapplet (MPLT_SA[SOURCE_VARIABLE]_CODES)
- ADI mapplet (MPLT_ADI_CODES)
- IA table (IA_CODES)

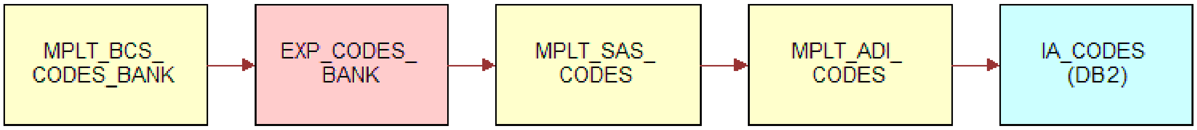


Figure 50. Codes Mapping

All codes mappings have the following three objects in common:

- Source Adapter mapplet
- ADI mapplet
- IA table

What makes each codes mapping unique are the remaining two objects—the Business Components mapplet and the Expression transformation. Therefore, when creating your new codes mapping, you can copy any codes mapping that comes from the same source and then change the Business Component mapplet and the Expression transformation.

NOTE: Not all codes mappings require an Expression transformation.

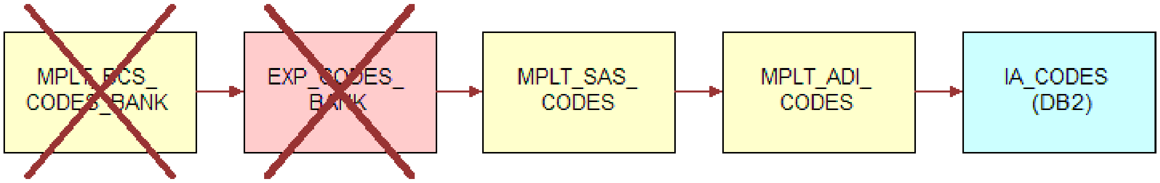
To create a new codes mapping

1. In PowerCenter Designer, open the applicable source configuration folder.
2. Select Tools > Mapping Designer.
3. Copy any existing codes mapping into Mapping Designer that has the same source as the one you want to create.

   By using an existing codes mapping you can avoid recreating the Source Adapter. Rename the codes mapping using the appropriate naming convention. For a list of naming conventions, see the *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.

4. Delete the mapping's existing Business Component mapplet and, if applicable, Expression transformation.

   You need to create a new Business Component mapplet and Expression transformation for your new mapping.



5. Create a new Business Component mapplet that includes a source definition, Source Qualifier, and mapplet output object (MAPO).

   For more information on creating a new Business Component, see the discussion on creating a new Business Component in Process of Creating and Modifying Business Adapters on page 213.

6. Drag the new Business Component mapplet into the mapping in Mapping Designer.
7. In the new codes mapping, create an Expression transformation for the Business Component mapplet.

   In this expression, you need to set the new category.

8. This category is the descriptive name of the type of code you are trying to create.

   a. Connect the detached ports in the new Codes mapping by first connecting the Business Component mapplet's output ports to the Expression input ports.

   b. Then, connect the Expression transformation output ports to the Source Adapter input ports from the mapping you copied in Step 3.

   The remaining output ports of the Source Adapter, as well as the input and output ports of the ADI, are already connected from this copied mapping.

9. Save your changes to the repository.

Creating a Derive Mapping

Derive mappings provide additional logic between the extract and load processes. They are primarily used to modify the format of data, or to merge data from two target tables and move it into a third. The naming convention for derive mappings is M_[SOURCE]_[SUBJECT]_DERIVE.

Derive mappings are only used in special cases. However, there may be a situation that requires additional features in a mapping. In this case, you need to build an additional mapping based on your needs.

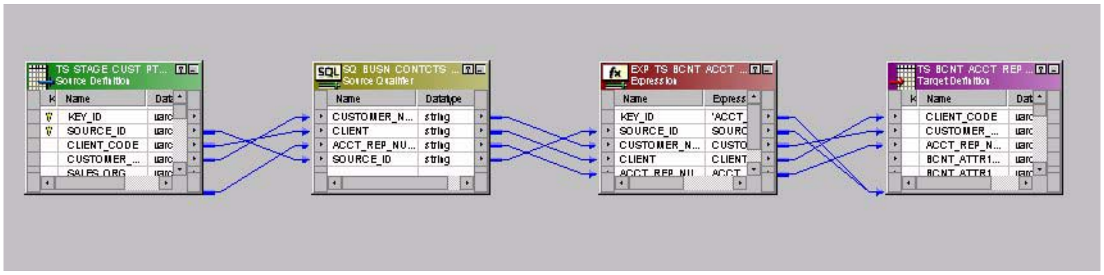
The procedure for building a derive mapping is the same as building a regular mapping. The structure of the mapping depends on the tables from which you want to derive data. The following are examples of derive mappings for SAP R/3:

- M_S_SALES_ORDLNS_LN_DERIVE

  Derives sales order line related partner and business data.

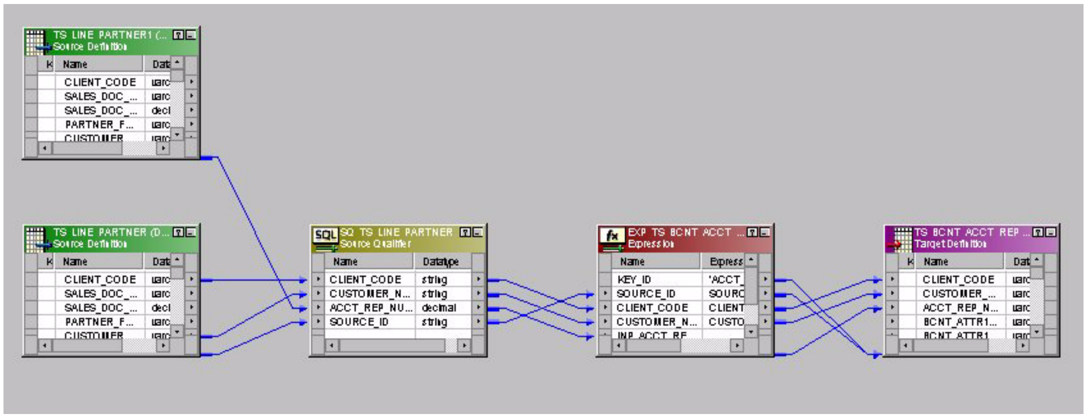
- M_S_BUSN_CONTCTS_ACCOUNT_REP_DERIVE_MASTER

  Extracts Sales representative information from SAP R/3, as shown in the following figure.



- M_S_BUSN_CONTCTS_ACCOUNT_REP_DERIVE_ORDLNS

  Extracts Account representative (order line level) information from SAP R/4, as shown in the following figure.



About Creating and Modifying Business Adapters

Business adapters consist of a Business Component mapplet and a Source Adapter mapplet. A single business adapter extracts a specific type of data (for example, sales order data) from a specific source (for example, SAP R/3), translates it into understandable business language where required, and prepares it for the source-independent Analytic Data Interface (ADI).

Because business adapters are source-specific, only certain business adapters for prepackaged sources are included in Siebel Customer-Centric Enterprise Warehouse. If you are extracting data from a source for which no prepackaged business adapters are available, you can feed data into Siebel Customer-Centric Enterprise Warehouse through a flat file, using Siebel Business Analytics' universal business adapters. You can also build your own business adapters.

As you begin creating new mappings, you may also need to create, or modify, mapplets, transformations, and so on. The following sections provide instructions for modifying the Business Component mapplet and the Source Adapter mapplet. Within each procedure, you can also find how to modify the transformations contained in the mapplets.

Process of Creating and Modifying Business Adapters

This section describes configuration procedures for creating a new Business Component and modifying an existing Business Component. It also describes configuration procedures for creating a new source adapter and modifying existing source adapter expression transformations.

To create and modify Business Adapters, perform the following tasks:

- Creating a New Business Component on page 213
- Modifying a Business Component on page 214
- Creating a New Source Adapter on page 215
- Modifying Source Adapter Expression Transformations on page 215

Creating a New Business Component

This task is a step in the Process of Creating and Modifying Business Adapters on page 213.

Business components reside in the configuration folder for each source and exist as mapplets. You can add new Business Component mapplets for a source.

To create a new Business Component

1. In PowerCenter Designer, open the applicable source configuration folder.
2. Select Mapplets > Create.

3 At the Mapplets Name prompt, enter a name for your new Business Component mapplet.

Siebel Customer-Centric Enterprise Warehouse naming conventions are described in the *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.

4 Expand the Sources folder.

Drag and drop the source table into Mapplet Designer.

5 Add Source Qualifiers as appropriate.

Depending on your PowerCenter settings, the Source Qualifier may be created automatically when you drag in the Source table.

6 Select Transformations > Create.

a At the Create Transformation prompt, select Mapplet Output from the drop down list window.

b Name the transformation using the MAPO_* naming convention.

Select Create, and then click Done.

7 Drag and drop ports from the Source Qualifier to the Output transformation.

Do not forget to link new data to the appropriate extension column. For information on the type of extension column to use, see Types of Extension Columns on page 176.

8 Edit the SQL statement in the Source Qualifiers.

9 Generate the SQL statement and save your changes to the repository.

Modifying a Business Component

This task is a step in the Process of Creating and Modifying Business Adapters on page 213.

When modifying a Business Component mapplet, you need to identify what areas of the mapplet you wish to modify. The following procedure contains instructions for adding a new source definition, connecting the ports to the Source Qualifier, editing the Source Qualifier, connecting the ports to the Output transformation, and editing the Output transformation. First, define what areas of the Business Component mapplet you wish to modify, and then follow the procedure only for those specific areas.

To modify an existing Business Component mapplet

1 In PowerCenter Designer, open the applicable source configuration folder.

2 Select Tools > Mapplet Designer.

3 Open the applicable Business Component mapplet.

4 Expand the Sources folder, and copy a source table into your new mapplet by dragging and dropping the table into Mapplet Designer.

5 Drag and drop required columns from the new source definition to the Source Qualifier.

6 Double-click the Source Qualifier to open the Edit Transformations box, and then:

   a. Click the Ports tab, and make any changes to the new ports as necessary.
   b. Click the Properties tab to make changes to the SQL statement as necessary.
   c. Click OK.

7. Drag and drop ports from the Source Qualifier to the Output transformation.
8. Save your changes to the repository.

Creating a New Source Adapter

This task is a step in the Process of Creating and Modifying Business Adapters on page 213.

Source Adapter mapplets are source-specific objects. Therefore, when creating them, you must make sure you put them in their applicable folder.

To create a new Source Adapter

1. In PowerCenter Designer, open the configuration folder where the Source Adapter mapplet is stored.

   For example, if the Source Adapter mapplet is for Oracle 11i, then open the Configuration for Oracle Applications v11i folder.

2. Select the generic mapplet of your choice from the navigator panel and select Edit > Copy.
3. Expand the configuration folder for the source for which you wish to create a new Source Adapter.
4. Highlight the mapplets node and select Edit > Paste.

   At the prompt, Copy the mapplet...?, click Yes.

5. Rename the Source Adapter mapplet to a name different than the one being copied.
6. Open your new mapplet, select Mapplets > Edit, and rename it to reflect its new function using the Siebel Customer-Centric Enterprise Warehouse naming conventions.

   Click OK.

7. Save your changes to the repository.

Modifying Source Adapter Expression Transformations

This task is a step in the Process of Creating and Modifying Business Adapters on page 213.

When you edit a Source Adapter mapplet, you add the desired input port (named INP_*) to the input (MAPI) side of the mapplet, as shown in Figure 51. You then copy the new input port from the MAPI to the Expression transformation and then link the two ports. In the Expression, link the port to the closest existing extension (EXT_*) port. Step-by-step procedures follow.

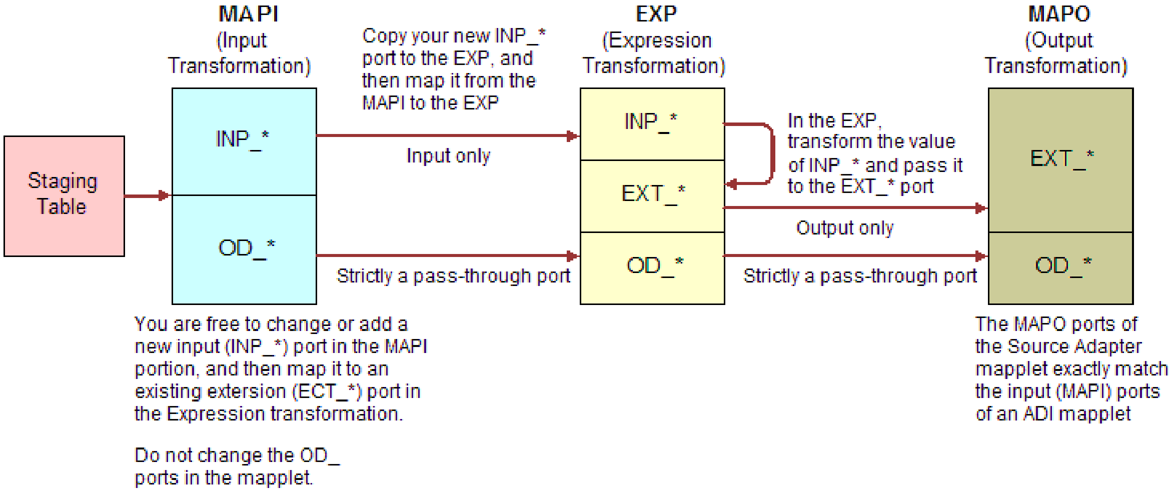


Figure 51. Input, Expression, and Output Ports of a Source Adapter Mapplet

To edit a Source Adapter Expression transformation

1 In PowerCenter Designer, open the applicable source configuration folder.

2 Select Tools > Mapplet Designer, and open the applicable Source Adapter mapplet.

3 Add a new input port to the MAPI side of the mapplet, following the INP_* naming convention.

4 Copy this new input port and add it to the Expression portion of the mapplet.

5 Link the new port from the MAPI to its new counterpart in the Expression.

6 In the Expression, clear Output indicator for the new port.

   You are mapping this port to an existing extension (EXT_*) port in Step 7.

7 From the available existing extension ports (EXT_*), select the closest appropriate port and double-click to open the Edit Transformations box.

8 On the Ports tab, find the chosen extension port, and select its Expression to open the Expression Editor.

   a In the Expression Editor, select your new input (INP_*) port.

      The extension column now contains the value of the new port you added.

   b Validate the new expression.

   c Edit any required ports in the Source Adapter.

      These ports vary depending on the dimension table.

9. If necessary, map the extension port you have configured to the corresponding EXT_* port in the Output transformation (MAPO).

   In Figure 51, the ports in the Output transformation of the mapplet match the input ports of the ADI mapplet exactly. This exact match makes sure that whatever comes out of the Output transformation directly feeds into the ADI.

10. Save your changes to the repository.

12 Checklist for Configuring Siebel Customer-Centric Enterprise Warehouse Applications

This chapter describes how to configure certain objects for particular sources to meet your business needs.

It contains the following topics:

Checklist for Initializing Siebel Customer-Centric Enterprise Warehouse

This section contains Siebel Customer-Centric Enterprise Warehouse mandatory configuration points for Oracle 11i, SAP R/3, universal source, and PeopleSoft 8.4:

- **Common Initialization Workflow files.** It is important that the date-related files are in the `$pmserver\SrcFiles` folder. These files reflect your data warehouse time span. For more information on the Common Initialization Workflow files, see Initialization Workflow Files on page 57.
- **Modifying session parameters for initial and incremental loads.** Set up the parameter files correctly for Oracle 11i, SAP R/3, and PeopleSoft 8.4. For more information on modifying session parameters for initial and incremental loads, see About Modifying Session Parameters for Initial and Incremental Loads on page 61.
- **Incremental loads.** Modify the parameter files correctly for incremental loads. For more information on modifying parameter files, see About Modifying Session Parameters for Initial and Incremental Loads on page 61.
- **Configuring the database parameter for the source system.** It is important that you have correctly configured the database parameter for your source system. For more information on configuring the database parameter for the source system, see Configuring the Database Parameter for the Source System on page 64.
- **Generating ABAP Code for SAP R/3.** If you are using an application from SAP R/3, make sure you have generated all the required ABAP codes. For more information on generating the ABAP codes, see Determining Configuration Requirements on page 31.

- **Deploying stored procedures for PeopleSoft and Oracle 11i.** If you are using PeopleSoft or Oracle 11i, deploy the appropriate stored procedures. For more information on deploying stored procedures, see Deploying Stored Procedures on page 49.
- **Table Analyze Utility.** The Siebel Customer-Centric Enterprise Warehouse uses the Table Analyze Utility to analyze tables after they are loaded. For more information on the Table Analyze Utility, see About the Table Analyze Utility on page 68 and Process of Configuring the Table Analyze Utility on page 68.

Checklist for Configuring the Siebel Enterprise Contact Center Analytics

This section contains the mandatory and optional configuration points for the Siebel Enterprise Contact Center Analytics application for Universal Source and post-load processing (PLP).

Checklist for Configuring the Siebel Enterprise Contact Center Analytics for Universal Source

This section contains the Siebel Enterprise Contact Center Analytics application configuration points that are specific to universal source.

The following section contains mandatory Siebel Enterprise Contact Center Analytics configuration points:

- **Dimension Key Resolution.** When creating the source files, the data and the format of the reference to the Dimension data (foreign key ID column) must match the KEY_ID in the corresponding dimensional source file. This is critical for accurate resolution of the foreign keys on the fact record when loading the data warehouse. For more information on Dimension key resolution, see About the Dimension Key Resolution Process for Universal Source on page 234.
- **Representative Activities spanning a date boundary.** If a representative's activity spans across the date boundary, then you must provide two different records for that representative for the same activity with different activity start dates and times. For more information on dates and setting up the Representative Activities table, see Setting Up the Representative Activities Table on page 238.
- **Setting the effective date ranges for Benchmark and Targets**. When configuring the source files for the Benchmarks and Targets fact table, you need to make sure that the targets and benchmarks cover the time periods for the capture of the Contact Center Performance and Representative Activity data. For more information on setting the effective date ranges for Benchmark and Targets, see Setting Up the Benchmarks and Targets Table on page 241.

The following section contains optional Siebel Enterprise Contact Center Analytics configuration points.

- **Configuring flags.** Many of the fact and dimension tables within the Siebel Enterprise Contact Center Analytics application use flag fields to provide value-added information pertaining to a contact or contact representative. These flag fields have logic to assign a default value to the flag while reading from the source files. For more information on configuring flags, see Configuring Flags for Siebel Enterprise Contact Center Analytics on page 242.

- **Excluding calls from the Answered Contact Count.** You may choose not to count calls which are completed in the IVR as an answered call. To do this you need to flag such calls with a new contact status. For more information excluding calls from the Answered Contact Count, see Setting Up the Contact Representative Snapshot Table on page 240

Checklist for Configuring the Siebel Enterprise Contact Center Analytics for post-load processing

The following section contains an optional Siebel Enterprise Contact Center Analytics application configuration point that is specific to post-load processing.

Excluding representative data from the Contact Representative aggregate tables. For more information on excluding representative data from the Contact Representative aggregate tables, see Excluding Representative Data from the Contact Representative Aggregate Tables for Post-Load Processing on page 244.

Checklist for Configuring the Siebel Enterprise Sales Analytics

This section contains the mandatory and optional configuration points for the Siebel Enterprise Sales Analytics application for SAP R/3, Oracle 11i, and post-load processing.

Checklist for Configuring the Siebel Enterprise Sales Analytics for SAP R/3

This section contains optional Siebel Enterprise Sales Analytics application configuration points that are specific to SAP R/3:

- **Configuring the Booking Flag Calculation in the Sales Order Lines Table.** To configure the booking flag calculation in the Sales Order Lines table, see Configuring the Booking Flag Calculation in the Sales Order Lines Table on page 248.
- **Tracking multiple attribute changes in Booking Lines.** If you want to track changes based on more than one attribute, in the SQL statement you must concatenate the attribute column IDs in the VAR_BOOKING_ID column. For more information on tracking multiple attribute changes in Booking Lines, see Tracking Multiple Attribute Changes in Bookings on page 251.
- **Configuring the Booking Flag calculation in the Sales Schedule Lines Level.** To configure bookings at the Sales Schedule Lines level, see Configuring the Booking Flag Calculation in the Sales Schedule Lines Level on page 252.
- **Setting up early and late tolerances for shipping.** To define early or late shipments with reference to the scheduled pick date, you need further configurations. For more information on configuring early and late tolerances for shipments, see Configuring Early and Late Tolerances for Shipments on page 254.

- **Including incomplete Sales Invoices.** By default, the Siebel Enterprise Sales Analytics application is configured to extract completed sales invoices when performing the Sales Invoice data extract. To extract incomplete sales invoices you would need further configurations. For more information on configuring Sales Invoice Lines data storage, see Configuring the Sales Invoice Extract on page 255.

- **Configuring Order Types for Backlog calculations.** To edit the Backlog flag, see Configuring Order Types for Backlog Calculations on page 255.

- **Setting the negative sign for the Order and Invoice Lines.** By default, the Siebel Customer-Centric Enterprise Warehouse does not use negative values in the quantity or amount columns for the IA_SALES_IVCLNS table or the IA_SALES_ORDLNS table. You can configure mapplets to account for negative values. For more information on accounting for negative values for Orders, Invoices, and Picks, see Accounting for Negative Values in Orders, Invoices, and Picks on page 267.

- **Domain Values.** For a list of CSV worksheet files and domain values for Siebel Enterprise Sales Analytics for SAP R/3, see Domain Values and CSV Worksheet Files for Siebel Enterprise Sales Analytics on page 287.

Checklist for Configuring the Siebel Enterprise Sales Analytics for Oracle 11i

This section contains optional Siebel Enterprise Sales Analytics application configuration points that are specific to Oracle 11i:

- **Including nonbooked lines in the Sales Booking Line fact table.** By default, only booked lines are brought over to the Sales Booking Line fact table. You can change this to include Nonbooked Lines. For more information on configuring the handling of Booked and Nonbooked Orders in the Order Lines and Bookings Table, see Configuring Sales Order Lines Data Storage on page 258.

- **Tracking changes in Booking Lines.** By default, only the changes in Order Amount, Quantity, Line ID, and Warehouse are tracked in the Booking Lines table (IA_SALES_BKGLNS). If you want to track other changes you can do so. For example, you may want to track changes to the sales representative who is handling the order. For more information on viewing the Data Warehouse changes by Salesperson ID, see About Tracking Attribute Changes in Bookings on page 259.

- **Tracking changes to dimensional attributes in Booking Lines.** If you want additional lines to be created to track changes to other dimensional attributes, you would need further configurations. For more information on how to track dimensional attribute changes in bookings, see Tracking Multiple Attribute Changes in Bookings on page 261.

- **Tracking hold information.** This step must be done in case you want to track holds. The configuration procedure requires that you map your source values to the set of domain values to translate the values. The Siebel Customer-Centric Enterprise Warehouse supports storage of up to nine different types of hold. For more information on assigning Sales Order hold types, see Configuring Sales Schedule Lines Data Storage on page 262.

- **Loading bookings at the Schedule Line Level instead of the Sales Order Line level.** You can configure the load process to load bookings at the Sales Schedule Line level instead of the Sales Order Line level. For more information on loading bookings at the Schedule Line level, see Configuring Sales Schedule Lines Data Storage on page 262.

- **Setting up early and late tolerances for shipping.** To define early or late shipments with reference to the scheduled pick date, you need further configurations. For more information on defining early and late tolerances for shipments, see Configuring Early and Late Tolerances for Shipments on page 264.

- **Including incomplete Sales Invoices.** By default, the Siebel Enterprise Sales Analytics application is configured to extract completed sales invoices when performing the Sales Invoice data extract. To extract incomplete sales invoices you would need further configurations. For more information on configuring Sales Invoice Lines data storage, see Configuring Sales Invoice Lines Data Storage on page 265.

- **Including closed orders for backlog calculations.** Be default, only orders with status Open are included in the backlog calculations. To include the closed orders, you would need further configurations. For more information on adding closed orders to backlog calculations, see Configuring Different Types of Backlog Calculations on page 265.

- **Configuring order types for backlog calculations.** By default, all order types are included in the backlog calculations. To exclude certain order types you would need further configurations. For more information on configuring order types for backlog calculations, see Configuring Different Types of Backlog Calculations on page 265.

- **Configuring the backlog history period date.** This configuration allows you to change the default monthly backlog snapshot to a different grain—for example weekly, daily, and so on. For more information on configuring the backlog period date, see Configuring Order Types for Backlog Calculations on page 266.

- **Setting the negative sign for the Order and Invoice Lines.** By default, the Siebel Customer-Centric Enterprise Warehouse does not use negative values in the quantity or amount columns for the `IA_SALES_IVCLNS` table or the `IA_SALES_ORDLNS` table. You can configure mapplets to account for negative values. For more information on accounting for negative values for Orders, Invoices, and Picks, see Accounting for Negative Values in Orders, Invoices, and Picks on page 267.

- **Domain Values.** For a list of CSV worksheet files and domain values for Siebel Enterprise Sales Analytics for Oracle 11i, see Domain Values and CSV Worksheet Files for Siebel Enterprise Sales Analytics on page 287.

Checklist for Configuring the Siebel Enterprise Sales Analytics for post-load processing

This section contains optional Siebel Enterprise Sales Analytics configuration points that are specific to post-load processing.

- **Aggregating Siebel Enterprise Sales Analytics tables.** To aggregate the Sales Invoice Lines and Sales Order Lines tables, see Process of Aggregating Siebel Enterprise Sales Analytics Tables on page 268.

- **Tracking multiple products sold as one package.** This configuration allows the user to set up the Order Line Key ID to reference all products sold in a bundle. For more information on tracking multiple products sold as one package, see About Tracking Multiple Products for Siebel Enterprise Sales Analytics on page 278.

- **Adding dates to the Order Cycle Time table.** To add more dates, you need to understand how the Order Cycle Times table is populated. For more information on adding dates to the Cycle Time table load, see Adding Dates to the Order Cycle Time Table for Post-Load Processing on page 279.

Checklist for Configuring the Siebel Enterprise Workforce Analytics

This section contains the mandatory and optional configuration points for the Siebel Enterprise Workforce Analytics application for Oracle 11i and post-load processing.

Checklist for Configuring the Siebel Enterprise Workforce Analytics for Oracle 11i

This section contains the Siebel Enterprise Workforce Analytics configuration points that are specific to Oracle 11i.

The following section contains mandatory Siebel Enterprise Workforce Analytics configuration points.

- **Configuring Workforce Operations.** For more information on configuring the Workforce Operations, see Process of Configuring Workforce Operations for Oracle 11i on page 292.
- **Configuring domain values and CSV worksheet files.** You need to configure the CSV files in Siebel Enterprise Workforce Analytics by mapping domain values to columns. For more information on Configuring Domain Values and CSV Worksheet Files, see Configuring Domain Values and CSV Worksheet Files for Siebel Enterprise Workforce Analytics on page 305.

The following section contains an optional Siebel Enterprise Workforce Analytics configuration point.

Configuring Workforce Payroll. You can modify the Workforce Payroll Filters and improve ETL performance for Workforce Payroll. For more information on configuring the Workforce Payroll, see Process of Configuring Workforce Payroll for Oracle 11i on page 301.

Checklist for Configuring the Siebel Enterprise Workforce Analytics for Post-load Processing

This section contains the Siebel Enterprise Workforce Analytics configuration points that are specific to post-load processing.

The following section contain an optional Siebel Enterprise Workforce Analytics configuration point.

Aggregating the Payroll table. You can aggregate the Payroll table in Siebel Enterprise Workforce Analytics. For more information on aggregate the Payroll table, see Aggregating the Payroll Table for Siebel Enterprise Workforce Analytics on page 303.

Checklist for Configuring the Siebel Financial Analytics

This section contains the mandatory and optional configuration points for the Financial applications for Oracle 11i, SAP R/3, and PeopleSoft 8.4. The Financial applications consist of Siebel General Ledger Analytics, Siebel Payables Analytics, Siebel Receivables Analytics, and Siebel Profitability Analytics.

Checklist for Configuring the Siebel Financial Analytics for Oracle 11i

This section contains the Siebel Financial Analytics applications configuration points that are specific to Oracle 11i.

The following section contains mandatory Siebel Financial Analytics configuration points:

- **Extracting Data Posted at the Detail-Level.** By default, the Siebel Customer-Centric Enterprise Warehouse assumes that the posting from your journal to your General Ledger is done at summary level and that references are maintained in the General Ledger for AP and AR subledgers. Configuration is required if import references are not maintained in the General Ledger and the posting from subledgers is at the detail level. For more information on extracting data posted at the detail-level, see Extracting Data Posted at the Detail-Level for Oracle 11i on page 311.
- **Mapping General Ledger Analytics account numbers to group account numbers.** You need to map General Ledger Analytics account numbers to group account numbers. For more information on mapping General Ledger accounts numbers to group account number, see Mapping Siebel General Ledger Analytics Account Numbers to Group Account Numbers on page 312.

The following section contains optional Siebel Financial Analytics configuration points:

- **Configuring the Set of Books ID.** By default, the Siebel Customer-Centric Enterprise Warehouse extracts data for all set of books. Configuration is required to extract data for a certain set of books only. For more information on filtering extracts based on Set of Book ID, see Filtering Extracts Based on Set of Books ID for Siebel General Ledger Analytics on page 313.
- **Configuring posted or unposted transactions.** By default, Siebel Customer-Centric Enterprise Warehouse extracts posted transactions. You can configure the Siebel Customer-Centric Enterprise Warehouse to extract unposted transactions as well as posted transactions. For more information on configuring the General Ledger transaction extracts and COGS extract, see Configuring Siebel General Ledger Analytics Transaction Extracts on page 314 and Configuring General Ledger Cost Of Goods Extract on page 315.
- **Configuring General Ledger account hierarchies.** Additional configuration is required to make sure the General Ledger hierarchical information is correct and to load the IA_HIERARCHIES table. For more information on configuring the General Ledger account hierarchies and loading hierarchies into the IA_HIERARCHIES table, see Configuring the General Ledger Account Hierarchies on page 315 and Loading Hierarchies for Siebel General Ledger Analytics on page 316.

- **Configuring the General Ledger Balance ID.** By default, the General Ledger Balance ID is maintained at the Set of Books and GL Code Combination ID level. If you want to maintain your General Ledger Balance at a different grain, you can redefine the GL Balance ID. For more information on configuring the General Ledger Balance ID, see Configuring the General Ledger Balance ID on page 319.

- **Configuring the AP Balance ID.** If you want to maintain your AP balance at a different grain, you can redefine the Balance ID value in the applicable mapplets. For more information on configuring the AP Balance ID, see Configuring AP Balance ID for Siebel Payables Analytics on page 320.

- **Configuring the AR Balance ID.** If you want to maintain your AR balance at a different grain, you can redefine the Balance ID value in the applicable mapplets. For more information on configuring the AR Balance ID, see Configuring AR Balance ID for Siebel Receivables Analytics and Siebel Profitability Analytics on page 320.

- **Configuring the AR Adjustments Extract.** If you want to extract additional types of AR adjustment entries, you can remove the filter in the Business Component mapplet. For more information on configuring the AR adjustments extract, see Configuring the AR Adjustments Extract for Siebel Receivables Analytics on page 321.

- **Configuring the AR Schedules Extract.** If you want to extract additional types of AR schedule entries, you must remove the filter in the Business Component mapplet. For more information on configuring the AR schedules extract, see Configuring the AR Schedules Extract on page 322.

- **Configuring the AR Cash Receipt Application Extract.** If you want to extract additional types of cash-receipt application entries, you can remove the filter in the Business Component mapplet. For more information on configuring the AR receipt application extract, see Configuring the AR Cash Receipt Application Extract for Siebel Receivables Analytics on page 322.

- **Configuring the AR Credit-Memo Application Extract.** If you want to extract additional types of credit-memo application entries, you can remove the filter. For more information on configuring the AR receipt application extract, see Configuring the AR Credit-Memo Application Extract for Siebel Receivables Analytics on page 323.

- **Configuring the Customer Costs and Product Costs Fact Tables.** In Siebel Profitability Analytics, the Customer Costs and Product Costs fact tables store the costing and expenses for the Profitability functional area. You need to use these tables with General Ledger Revenue and General Ledger COGS fact tables. For more information on configuring the Customer Costs and Product Costs Fact Tables, see Configuring the Customer Costs Lines and Product Costs Lines Tables for Siebel Profitability Analytics on page 324.

Checklist for Configuring the Siebel Financial Analytics for SAP R/3

This section contains the Siebel Financial Analytics configuration points that are specific to SAP R/3. The following section contains mandatory Siebel Financial Analytics configuration points:

- **Extracting Data Posted at the Header Level.** By default, the Siebel General Ledger Analytics application extracts sales information posted to the General Ledger at the detail level. However, you can configure the extraction if your installation of SAP R/3 is configured to store data at the header level. For more information on extracting data posted at the header level, see Extracting Data Posted at the Header Level for SAP R/3 on page 329.

- **Mapping General Ledger Analytics account numbers to group account numbers.** You need to map General Ledger Analytics account numbers to group account numbers. For more information on mapping General Ledger accounts numbers to group account number, see Configuring the Group Account Number Categorization for Siebel General Ledger Analytics on page 330.

The following section contains optional Siebel Financial Analytics configuration points:

- **Configuring General Ledger account hierarchies.** Additional configuration is required to make sure the General Ledger hierarchical information is correct and to load the IA_HIERARCHIES table. For more information on configuring the General Ledger account hierarchies and loading hierarchies into the IA_HIERARCHIES table, see Configuring the General Ledger Account Hierarchies on page 332 and Configuring Hierarchy ID in Source Adapter for Siebel General Ledger Analytics on page 334.
- **Configuring General Ledger Balance Extract.** By default, this is maintained at GL account—Company—Business Area—Client level. The Client is used to distinguish source instances if you have multiple source instances for your data warehouse. To change the grain at which you accumulate the General Ledger balance, modify the Balance ID or Key ID definition. For more information on configuring General Ledger Balance Extract, see Configuring the Siebel General Ledger Analytics Balance Extract on page 335.
- **Configuring the transaction types.** You can configure the transaction type by editing the xact_type_code_sap.csv file. For more information on configuring Configuring the transaction types, see Configuring the Transaction Types for Siebel Financial Analytics on page 331.
- **Configuring the AP Balance ID.** If you want to maintain your AP balance at a different grain, you can redefine the Balance ID value in the applicable mapplets. For more information on configuring the AP Balance ID, see Configuring the Siebel Payables Analytics Balance Extract on page 336.
- **Configuring the AR Balance ID.** If you want to maintain your AR balance at a different grain, you can redefine the Balance ID value in the applicable mapplets. For more information on configuring the AR Balance ID, see Configuring the Siebel Receivables Analytics Balance Extract on page 338.
- **Configuring the Customer Costs and Product Costs Fact Tables.** In Siebel Profitability Analytics, the Customer Costs and Product Costs fact tables store the costing and expenses for the Profitability functional area. You need to use these tables with General Ledger Revenue and General Ledger COGS fact tables. For more information on configuring the Customer Costs and Product Costs Fact Tables, see Configuring the Customer Costs Lines and Product Costs Lines Tables for Siebel Profitability Analytics on page 337.

Checklist for Configuring the Siebel Financial Analytics for PeopleSoft 8.4

This section contains the Siebel Financial Analytics configuration points that are specific to PeopleSoft 8.4.

The following section contains mandatory Siebel Financial Analytics configuration points:

- **Mapping General Ledger Analytics account numbers to group account numbers.** You need to map General Ledger Analytics account numbers to group account numbers. For more information on mapping General Ledger accounts numbers to group account number, see Configuring the Primary Ledger Name for Siebel General Ledger Analytics on page 343.
- **Configuring the Primary Ledger name.** By default, the name of the Primary Ledger is set to LOCAL for PeopleSoft. However, if the name of your Primary Ledger is not LOCAL, you can change this value by modifying the `file_parameters_psft84.csv` file. For more information on configuring the Primary Ledger name, see Process of Configuring Siebel Financial Analytics for PeopleSoft 8.4 on page 340.
- **Configuring PeopleSoft Trees.** For PeopleSoft, the Siebel Financial Analytics application sources data from a data structure, called PeopleSoft Trees, to get information about the organization's General Ledger hierarchies, and so on. For a PeopleSoft environment with different tree names to the Siebel Financial Analytics application, you need to import these into the PowerCenter repository, and replace the old tree names with the new tree names. For more on PeopleSoft Trees, see About PeopleSoft Trees in Siebel Financial Analytics on page 339, Customizing the PeopleSoft Tree Names on page 340, and Importing PeopleSoft Trees Into the PowerCenter Repository on page 341.

Checklist for Configuring the Siebel Financial Analytics for Post-Load Processing

This section contains the Siebel Financial Analytics configuration points that are specific to post-load processing.

The following section contains optional Siebel Financial Analytics configuration points:

- **Configuring Aging Buckets.** You need to configure the values for the first three bucket start and bucket end days. For more information on configuring Aging Buckets, see Configuring Aging Buckets for Siebel Receivables Analytics on page 345 or Configuring Aging Buckets for Siebel Payables Analytics on page 346.
- **Configuring the History Period for the Invoice Level.** You configure the history period value to match your business requirements. For more information on configuring the History Period for the Invoice Level, see Configuring the History Period for the Invoice Level for Siebel Receivables Analytics on page 346 or Configuring the History Period for the Invoice Level for Siebel Payables Analytics on page 347.

Checklist for Configuring the Siebel Strategic Sourcing Analytics

This section contains the mandatory and optional configuration points for the Siebel Strategic Sourcing Analytics for Oracle 11i, universal source, SAP R/3, and post-load processing.

Checklist for Configuring the Siebel Strategic Sourcing Analytics for Oracle 11i

This section contains the Siebel Strategic Sourcing Analytics configuration points that are specific to Oracle11i.

The following section contains optional Siebel Strategic Sourcing Analytics configuration points:

- **Configuring the Region Name definition.** This configuration allows you to load specific Region Names into the IA_CODES table. For more information on configuring the Region Name definition, see Configuring the Region Name on page 350.
- **Configuring the State Name definition.** This configuration allows you to load specific State Names into the IA_CODES table. For more information on configuring the State Name definition, see Configuring the State Name on page 351.
- **Configuring the Country Names definition.** This configuration allows you to load specific Country Names into the IA_CODES table. For more information on configuring the Country Names definition, see Configuring the Country Name on page 352.
- **Configuring the Make-Buy Indicator.** Your organization may require different indicator codes. If so, you can modify the indicator logic by reconfiguring the condition in the MPLT_SAI_PRODUCTS mapplet. For more information on configuring the Make-Buy Indicator, see Configuring the Make-Buy Indicator on page 352.
- **Extracting particular purchase order records.** By default, the filter condition is set to BLANKET or STANDARD. However, you can change this value to some conditional statement that only allows particular types of records to be extracted. For more information on extracting particular purchase order records, see Extracting Particular Purchase Order Records on page 353.
- **Configure the Purchase Organization hierarchy.** The product allows a ten-level hierarchy for Purchasing Organizations. By default, the first three levels are set to Organization ID, Legal Entity ID, and Set of Books ID. You may configure the remaining seven levels. For more information on configuring the Purchase Organization hierarchy, see Configuring the Purchase Organization Hierarchy on page 354.
- **Configuring the Siebel Business Analytics Repository.** You can map the department segment as a cost center in the Siebel Business Analytics Repository. For more information on configuring the Siebel Business Analytics Repository for Siebel Strategic Sourcing Analytics, see Configuring the Siebel Business Analytics Repository for Siebel Strategic Sourcing Analytics on page 354.

Checklist for Configuring the Siebel Strategic Sourcing Analytics for SAP R/3

This section contains the Siebel Strategic Sourcing Analytics configuration points that are specific to SAP R/3

The following section contains mandatory Siebel Strategic Sourcing Analytics configuration points:

- **Configuring the Siebel Business Analytics Repository.** The Requisition Cost and Purchase Cost fact tables are not loaded for SAP R/3. You need to disable these tables in the Siebel Business Analytics Repository. For more information on configuring the Siebel Business Analytics Repository for SAP R/3, see Domain Values and CSV Worksheet Files for Siebel Strategic Sourcing Analytics on page 375.
- **Configuring the date parameters for the SAP R/3 parameter file.** You need to set the PARM_NVALUE_1 value in the file_parameters_sapr3.csv file to the number of days that you expect your orders to be open. For more information configuring the date parameters for the SAP R/3 parameter file, see Configuring the Date Parameters for the SAP R/3 Parameter File on page 356.

Checklist for Configuring the Siebel Strategic Sourcing Analytics for Universal Source

This section contains the Siebel Strategic Sourcing Analytics configuration points that are specific to Universal Source.

The following section contains a mandatory Siebel Strategic Sourcing Analytics configuration point:

Configuring Expense Payment Types. The various expense types in the source data are mapped to Reimbursable Expenses (E), Expenses Prepaid (P), and Cash Advance (C). For more information on configuring Expense Payment types, see Configuring Expense Payment Types on page 360.

The following section contains optional Siebel Strategic Sourcing Analytics configuration points:

- **Configuring the Preferred Merchant Flag.** For more information on configuring the Preferred Merchant Flag, see Configuring the Preferred Merchant Flag on page 359.
- **Configuring the Customer Billable Indicator.** For more information on configuring the Customer Billable Indicator, see Configuring the Customer Billable Indicator on page 359.
- **Configuring the Receipts Indicator.** For more information on configuring the Receipts Indicator, see Configuring the Receipts Indicator on page 360.
- **Configuring the Default Expense Distribution Percentage.** For more information on configuring the Default Expense Distribution Percentage, see Configuring the Default Expense Distribution Percentage on page 361.
- **Configuring Lookup Dates for Currency Conversion.** The Siebel Strategic Sourcing Analytics application uses the actual expiry date (ACTUAL_EXP_DATE) for looking up the exchange rate. You can configure the module to use a different date if required. For more information on configuring the Default Expense Distribution Percentage, see Configuring Lookup Dates for Currency Conversion on page 361.
- **Configuring the Siebel Business Analytics Repository.** You can configure the General Ledger Account and the Cost Center tables for universal source. For more information on configuring the Siebel Business Analytics Repository for Siebel Strategic Sourcing Analytics, see Configuring the Siebel Business Analytics Repository for Siebel Strategic Sourcing Analytics on page 362.

Checklist for Configuring the Siebel Strategic Sourcing Analytics for post-load processing

This section contains the Siebel Strategic Sourcing Analytics configuration points that are specific to post-load processing.

The following section contains optional Siebel Strategic Sourcing Analytics configuration points:

- **Aggregating Siebel Strategic Sourcing Analytics tables.** You can aggregate the Purchase Receipts and Purchase Cycle Lines tables. For more information on Aggregating Siebel Strategic Sourcing Analytics tables, see Configuring Expenses for Post-Load Processing on page 363.
- **Configuring the extraction of Invoice Details.** If you identify values other than the default values for an expense record, you can use those values by adding a condition to the expression in the applicable post-load processing mapping. For more information on configuring the extraction of Invoice Details for Expense-Related Payments, see Configuring the Extraction of Invoice Details on page 364.
- **Running the S_M_PLP_EXPENSES_INVOICE_UPD_ALT session.** Setting the S_M_PLP_EXPENSES_INVOICE_UPD_ALT session to run when the Incremental Invoice Load Frequency differs for the Expense functional area. For more information on setting the session to run when the Incremental Invoice Load Frequency differs for the Expense functional area, see Configuring Expenses for Post-Load Processing on page 363.
- **Implementing temporary storage when aggregate Load Frequencies are modified for the Expense Functional Area.** For more information on implementing temporary storage when aggregate Load Frequencies are modified for the Expense functional area, see Configuring Expenses for Post-Load Processing on page 363.

Checklist for Configuring the Siebel Supply Chain Analytics

This section contains the mandatory and optional configuration points for the Siebel Supply Chain Analytics for Oracle 11i.

Checklist for Configuring the Siebel Supply Chain Analytics for Oracle 11i

This section contains the Siebel Supply Chain Analytics configuration points that are specific to Oracle 11i.

The following section contains mandatory Siebel Supply Chain Analytics configuration points:

- **Deploying stored procedures.** Compile the Compute_Bounds_Ora11i.sql stored procedure in the target data warehouse. For more information on deploying stored procedures, see Deploying Stored Procedures on page 49.
- **Configuring the Make-Buy Indicator.** Your organization may require different indicator codes. For more information on the Make-But Indicator, see Configuring the Make-Buy Indicator on page 387.

The following section contains optional Siebel Supply Chain Analytics configuration points:

- **Configuring the Bill of Materials (BOM) explosion option.** This configuration allows you to choose an explosion option to load a BOM structure into the IA_BOM_ITEMS table. For more information on configuring the BOM explosion option, see Configuring the Bill of Materials Explosion on page 378.
- **Configuring the left bound and right bound calculation.** This configuration allows you to turn on or off the calculation of the left bound and right bound in the IA_BOM_ITEMS table. For more information on configuring the left bound and right bound calculation, see Configuring the Left Bound and Right Bound Calculation Option on page 382.
- **Configuring the Quantity types for product transactions.** If your definition of goods received or delivery quantity is different from the prepackaged condition, then you can edit the condition to suit your business needs. For more information on configuring the Quantity type for product transactions, see Configuring Quantity Types for Product Transactions on page 384.
- **Configuring the Region Name.** This configuration allows you to load specific Region Names into the IA_CODES table. For more information on configuring the Region Name, see Configuring the Region Name on page 385.
- **Configuring the State Name.** This configuration allows you to load specific State Names into the IA_CODES table. For more information on configuring the State Name, see Configuring the State Name on page 386.
- **Configuring the Country Name.** This configuration allows you to load specific Country Names into the IA_CODES table. For more information on configuring the Country Name, see Configuring the Country Name on page 386.

Checklist for Configuring the Siebel Supply Chain Analytics for Post-Load Processing

This section contains the Siebel Supply Chain Analytics configuration points that are specific to post-load processing.

The following section contains optional Siebel Supply Chain Analytics configuration points:

- **Configure the Inventory Balance aggregate table.** This sections allows you configure the Inventory Balance (IA_INV_BALANCE_A1) aggregate table. For more information on configuring the Inventory Balance aggregate table, see Configuring the Inventory Balance Aggregate Table on page 389.
- **Configure the Product Transaction aggregate table.** This sections allows you configure the Product Transaction (IA_PROD_XACTS_A1) aggregate table. For more information on configuring the Product Transaction aggregate table, see Configuring the Product Transaction Aggregate Table on page 391.

13 Configuring Siebel Enterprise Contact Center Analytics

This chapter describes how to configure certain objects for the universal source to meet your business needs.

It contains the following topics:

Overview of Siebel Enterprise Contact Center Analytics

The Siebel Enterprise Contact Center Analytics application tracks incoming and outgoing contacts and their resulting activities, and the agents performing contact center tasks. The Siebel Enterprise Contact Center Analytics application is made up of these functional areas:

- Customer Service
- Contact Center and Agent Performance
- Service Delivery and Costs

About the Dimension Key Resolution Process for Universal Source

Dimension keys uniquely identify each record in a dimension table. The purpose of dimension keys is to relate one dimension table record to a fact table record. Thus, the dimension key must be stored in both the dimension table and the fact table and resolve using the dimension table's KEY_ID and SOURCE_ID column values.

For universal business adapters, users supply the dimension KEY_ID and SOURCE_ID column values through a flat file interface. The same values for KEY_ID and SOURCE_ID are expected in both the dimension and fact business adapters so that the correct dimension key is resolved and loaded into its fact table.

The dimension key resolution process entails two steps:

1. Run the dimension table workflows to extract and load dimension records.

   The dimension load mapping automatically creates a surrogate key for each record in the dimension table. This surrogate key value populates the dimension table's primary key column, which is referred to as the *dimension key*. Similar to the KEY_ID column, which uniquely identifies the record within the source system, the dimension key uniquely identifies the record in the data warehouse dimension table.

2. Run the fact table workflows to extract and load fact records.

   Records must contain the dimension ID column values for each fact record; these values must be the same values as the KEY_ID in the corresponding dimension tables.

The following sections describe these two steps in more detail by taking the example of one fact table (IA_REP_ACTVTS) and one dimension table (IA_EVENTS). However, this process applies to all fact and dimension tables joined by a dimension key.

Loading the Dimension Table

This task is explained by taking the example of the IA_EVENT_TYPES table.

Loading the IA_EVENT_TYPES table requires the following ETL processes:

1. The M_F_EVENT_TYPES_EXTRACT mapping extracts the data from file_event_types.csv and populates the TF_EVENT_TYPES staging table.
2. The M_F_EVENT_TYPES_LOAD mapping sources data from the staging table and passes it over to the Analytic Data Interchange (ADI). The ADI generates the surrogate key for each record in the staging table, then inserts it into IA_EVENT_TYPES target table.

Loading the Fact Table

This task is explained by taking the example of the IA_REP_ACTVTS fact table.

Loading the IA_REP_ACTVTS fact table requires the following ETL processes:

1. The M_F_REP_ACTVTS_EXTRACT mapping extracts the data from file_rep_actvts.csv and populates the TF_REP_ACTVTS staging table.
2. The M_F_REP_ACTVTS_LOAD mapping sources the data from the staging table, and the fact ADI mapplet resolves the dimension key by doing a lookup on IA_EVENT_TYPES using the values supplied in the ACTIVITY_TYPE_ID column and the SOURCE_ID column. Then, the ADI populates the IA_REP_ACTVTS fact table.

Since the dimension *_ID values are supplied through the Universal Interface flat file, it is critical that you supply the same value for the KEY_ID in the dimension table and the corresponding *_ID field in the joined fact table. In addition, you must verify that the SOURCE_ID column values match (for Universal Sources, the value of the SOURCE_ID column is GENERIC). If you supply different values for the two tables, the fact table load mapping is not able to resolve the dimension key. As a result, you cannot perform queries on the fact table using that dimension.

About Configuring the Event Type Columns

The IA_EVENT_TYPES is a dimension class table which tracks various types of events. This dimension table plays multiple roles in the Siebel Enterprise Contact Center Analytics application. The various roles are described in the following section.

The Activity Type Column

Contact center representatives engage in various activities, such as logging into the Automated Call Distributor (ACD) system to handle customer calls, taking a scheduled break, taking an unscheduled break, and so on. All such activities that you want to load into the warehouse should be provided in the file_event_types.csv source file to be stored in the IA_EVENT_TYPES table with the EVENT_CLASS column set to the CONTACT_CENTER_REP_ACTIVITY domain value.

The ACTIVITY_TYPE_KEY dimension key in IA_REP_ACTVTS fact table identifies the nature of the activity. This key is resolved using the IA_EVENT_TYPES table. To resolve the ACTIVITY_TYPE_KEY dimension key in IA_REP_ACTVTS table, the IA_REP_ACTVTS and IA_EVENT_TYPES tables are joined through the ACTIVITY_TYPE_ID column and the SOURCE_ID column. For the ACTIVITY_TYPE_KEY dimension key to resolve properly in the IA_REP_ACTVTS fact table, you must verify that the ACTIVITY_TYPE_ID column and the SOURCE_ID column values in file_rep_actvts.csv file match with the KEY_ID column and the SOURCE_ID column values in the file_event_types.csv file. If the two columns do not match for a particular record, the fact load mapping cannot resolve the dimension key for that fact record.

Call Type (or Contact Type) Column

In a contact center there are various reasons for which customers contact an organization and organizations contact a customer. The contact can happen through various channels, such as phone, online chat, email, fax, and so on.

Some of the reasons for a customer to contact your organization include:

- Placing an order for a product or service.

- Lodging a complaint.
- Inquiring about a product offering.

Some of the reasons for your organization to contact a customer include:

- Performing a customer satisfaction survey.
- Following up on an inquiry.

The call types that you want to load into the Siebel Customer-Centric Enterprise Warehouse are provided in the file_event_types.csv source file to be stored in the IA_EVENT_TYPES table with the EVENT_CLASS column set to the CONTACT_TYPE domain value.

The CALL_TYPE_KEY dimension key in IA_ACD_EVENTS fact table identifies the type of call. This key is resolved using the IA_EVENT_TYPES table. To resolve the CALL_TYPE_KEY dimension key in IA_ACD_EVENTS fact table, the IA_ACD_EVENTS and IA_EVENT_TYPES tables are joined through the CALL_TYPE_ID column and the SOURCE_ID column. For the CALL_TYPE_KEY dimension key to resolve properly in the IA_ACD_EVENTS fact table, you must verify that the CALL_TYPE_ID column and the SOURCE_ID column values in file_acd_events.csv file match with the KEY_ID column and the SOURCE_ID column values in the file_event_types.csv file. If the two columns do not match for a particular record, the fact load mapping cannot resolve the dimension key for that fact record.

The CONTACT_TYPE_KEY dimension key in IA_CNTCTREP_SNP fact table identifies the same information and it is resolved in a similar process. It requires the CNTCT_TYPE_ID column and the SOURCE_ID column values in the file_cntctrep_snp.csv file to match with the KEY_ID column and the SOURCE_ID column values in the file_event_types.csv file.

Call Event Type Column

In a call center, the Computer Telephony Integration (CTI) system tracks each activity associated with a call, including:

- Call placed in queue to be answered
- Call answered by a contact representative
- Call placed on hold by a contact representative
- Call transferred from one contact representative to another
- Call hung up by the customer

The call events that you want to load into the Siebel Customer-Centric Enterprise Warehouse are provided in the file_event_types.csv source file and stored in the IA_EVENT_TYPES table with the EVENT_CLASS column set to INTRA_CALL_ACTIVITY.

To resolve the CALL_EVENT_TYPE_KEY dimension key in IA_ACD_EVENTS fact table, the IA_ACD_EVENTS and IA_EVENT_TYPES tables are joined through the CALL_EVENT_TYPE_ID column and the SOURCE_ID column. For the CALL_EVENT_TYPE_KEY dimension key to resolve properly in the IA_ACD_EVENTS fact table, you must verify that the CALL_EVENT_TYPE_ID column and the SOURCE_ID column values in file_acd_events.csv file match with the KEY_ID column and the SOURCE_ID column values in the file_event_types.csv file. If the two columns do not match for a particular record, the fact load mapping cannot resolve the dimension key for that fact record.

After-Call Work Activity (ACW) Type Column

On many occasions, contact representative are required to perform certain tasks after the contact. These tasks might include creating a follow-up action item list, dispatching the case from the contact to a particular group, and so on. These activities are known as after-call work (ACW) activities. The call events that you want to load into the Siebel Customer-Centric Enterprise Warehouse are provided in the file_event_types.csv source file and stored in the IA_EVENT_TYPES table with the EVENT_CLASS column set to the ACW_ACTIVITY_TYPE domain value.

To resolve the ACW_ACT_TYPE_KEY dimension key in IA_CNTCTREP_SNP fact table, the IA_CNTCTREP_SNP table is joined with IA_EVENT_TYPES table through the ACW_ACT_TYPE_ID column and the SOURCE_ID column. For the ACW_ACT_TYPE_KEY dimension key to resolve properly in the IA_CNTCTREP_SNP fact table, you must verify that the ACW_TYPE_TYPE_ID column and the SOURCE_ID column values in the file_cntctrep_snp.csv file match with the KEY_ID column and the SOURCE_ID column values in the file_event_types.csv file. If the two columns do not match for a particular record, the fact load mapping cannot resolve the dimension key for that fact record.

About Configuring Contact Status Type Column

The IA_STATUS is a dimension class table that tracks various classes of status such as those associated with the purchase orders, requisitions, shipments, inventory, customer cases, contact statuses, and so on. These classes are differentiated by the STATUS_TYPE column.

For the Siebel Enterprise Contact Center Analytics application one of the important statuses is the Contact Status. All contacts made either by the customer to your organization, or by your organization to a customer, are assigned a status. Examples include:

- customer abandoned call before contact
- customer abandoned call during contact
- contact completed

The contact statuses that you want to load into the Siebel Customer-Centric Enterprise Warehouse are provided in the file_status.csv source file to be stored in the IA_STATUS table with the STATUS_TYPE column set to the CONTACT_STATUS domain value.

The IA_STAT_TYPE_CODE column in the IA_STATUS table also contains domain values. The four domain values ABANDONED, RELEASE, DISCONNECTED, and HANGUP, are used in the computation of contact center performance metrics. Therefore, it is critical that while you load all your Contact Statuses through the source file, the records are mapped into the appropriate IA_STAT_TYPE_CODE domain value.

To resolve the CNTCT_STATUS_KEY dimension key in IA_CNTCTREP_SNP fact table, the IA_CNTCTREP_SNP table is joined with IA_STATUS through the CNTCT_STATUS_ID column and the SOURCE_ID column. For the CNTCT_STATUS_KEY dimension key to resolve properly in the IA_CNTCTREP_SNP fact table, you must verify that the CNTCT_STATUS_ID column and the SOURCE_ID column values in file_cntctrep_snp.csv match with the KEY_ID column and the SOURCE_ID column values in the file_status.csv file. If the two columns do not match for a particular record, the fact load mapping cannot resolve the dimension key for that fact record.

Configuring Dates and Times

Local date and time in Siebel Enterprise Contact Center Analytics application calculations and analysis refers to the date and time at the contact's location. However, given different time zones and reporting requirements, you may need to use another date and time. To do so, you must supply these values through the flat file interface.

For example, when doing any period-based calculations or analysis on representative activities, prepackaged logic uses the ACTIVITY_START_LDT and ACTIVITY_END_LDT local date column values from the IA_REP_ACTVTS table. However, if you do not want to use the local dates, then pass the new dates in the ACTIVITY_START_DT and ACTIVITY_END_DT columns into the `file_rep_actvts.csv` flat file interface.

If you change the dates and times, you must do it consistently for all rows in the given table. For example, you cannot have some rows using local dates and times, while other rows are use a different time zone. Table 30 provides a list of the applicable local date columns for each relevant table.

Table 30. Date Columns and Flat Files

Flat file	Applicable Date Column	Table Using the Local Date
file_rep_actvts.csv	ACTIVITY_START_DT ACTIVITY_END_DT	IA_REP_ACTVTS
file_acd_events.csv	EVENT_START_DT EVENT_END_DT	IA_ACD_EVENTS
files_ cntctrep_snp.csv	CNTCT_START_DT CNTCT_END_DT	IA_CNTCTREP_SNP

To provide new dates in the flat file interface

1. Open the applicable flat file interface.
2. In the flat file interface, input the new dates in the *_DT fields.
3. Save your changes to the flat file interface.
4. Run a test load for 10 records to verify that your new dates are loaded into the applicable table.

Setting Up the Representative Activities Table

When setting up the Representative Activities table you must consider the following:

- All events in the Representative Activities table are time span events. The events are not point in time events.

- The calculation of the Actual, Scheduled, Login, and Break durations are based on the event durations in the source-system data. To avoid duplication in a representative's time, the representative activity records must not overlap in time. For example, if the Login and Break activities overlap in time in the source-system data, then the time durations are counted towards both categories.
- The hourly aggregate is the lowest level of aggregation provided. Representatives are counted as present for an hourly bucket if they are present for any part of that hour. For example, if a representative activity starts at 9.45 A.M. and ends at 10.45 A.M., the representative is counted as present for 9-10 A.M. and 10-11 A.M. time buckets. No weight factor is used to indicate the representative is available for part of the hour. However, the duration of activities are apportioned into the two hourly buckets.
- The number of breaks a representative takes is calculated by counting the number of break records. There is one break record for each representative for each break (scheduled or actual). If a break is split into multiple records in the source system, then it is counted as multiple breaks in the Siebel Customer-Centric Enterprise Warehouse.
- If a representative’s activity spans across the date boundary, then you must provide two different records for that representative for the same activity with different activity start dates and times. For example, if a representative logs on to the Automatic Call Distributor (ACD) system at 23:30 on January 4, 2004 and logs off from the ACD system at 00:30 on January 5, 2004, then create two records in the `file_rep_actvts.csv` flat file interface, as shown in the following table.

Representative	Activity Type	Activity Start Date	Activity Start Time	Activity End Date	Activity End Date
Rep 1	`LOGIN`	`01/04/2004`	`23:30:00`	`01/04/2004`	`23:59:59`
Rep 1	`LOGIN`	`01/05/2004`	`00:00:00`	`01/05/2004`	`00:30:00`

To set up the Representative Activities table

1. Open the `file_rep_actvts.csv` file using Microsoft WordPad or Notepad in the `$pmserver\SrcFiles` folder.
2. Type in your data to load the Representative Activities table.

3. Save and close the file.

Setting Up the Contact Representative Snapshot Table

The Contact Representative Snapshot table stores information at a contact and representative grain. For example, if a call is handled by two representatives, there are two records for that contact in the Contact Representative Snapshot table.

When setting up the Contact Representative Snapshot table you must consider the following:

- The Abandoned Contact Count, Answered Contact Count, Hangup Contact Count, and Released Contact Count metrics are counts of contacts based on the Contact Status. The Contact Representative Snapshot table is preconfigured to expect the Contact Status in the `file_cntctrep_snp.csv file` is to be at a Contact level. If you configure the Contact Status at the contact and representative level, you need to make sure that these aggregate metrics are defined at the contact and representative level in the appropriate workflows. You need to make any changes in the Select clause of the Source Qualifier SQL statement in the `M_PLP_CC_REP_A1_LOAD` mapping. You also need to configure the metadata in the repository file. You need to change the definitions of these metrics in the Logical Table Source that maps to the `IA_CNTCTREP_SNP` fact table.
- *Answered contacts* are defined as the contacts whose status is not marked as ABANDONED. The Answered Contact Count metric is calculated as follows:

  ```
  COUNT(DISTINCT (CASE WHEN IA_STATUS.IA_STAT_TYPE_CODE != 'ABANDONED'  THEN
  IA_CNTCTREP_SNP.CNTCT_NUM END)).
  ```

  You can choose not to count calls which are completed in the Interactive Voice Response (IVR) as an answered call. You can exclude these contacts from the Answered Contact Count by assigning them a different or new Contact Status.
- As contacts are associated with a representative, a Contact Representative Identifier needs to be provided for each fact record. The IVR can be modeled as a virtual representative and assigned a Contact Representative Identifier. Calls waiting in a general queue, and not assigned to a representative, also need to be given their own Contact Representative Identifier.
- The majority of the data for the Contact Representative Snapshot table is sourced from the data in the `file_acd_events.csv` file. You must make sure that the source data is consistent across the `file_acd_events.csv` and `file_cntctrep_snp.csv` files.

To set up the Contact Representative Snapshot table

1. Open the `file_cntctrep_snp.csv` file using Microsoft WordPad or Notepad in the `$pmserver\SrcFiles` folder.
2. Type in your data to load the Contact Representative Snapshot table.
3. Save and close the file.

Setting Up the Benchmarks and Targets Table

The benchmarks and targets subject area is used to establish targets for various metrics for your Contact Center and also capture industry benchmarks for the same metrics. The values of these metrics are calculated using the Contact Center Performance subject area and are compared to the Benchmarks and Targets table to evaluate the performance of your Contact Center.

When setting up the Benchmarks and Targets table you must consider the following:

- The `file_cc_bmk_tgt.csv` file must supply the effective date range for each benchmark record. The date range is used to identify the appropriate benchmark to compare with the actuals and the determination of other metrics such as the Service Level. Actuals refers to the actual value of the metric (during the period) as opposed to the planned or targeted value of the metric.
- You need to supply an appropriate date range for the benchmark records. For example, if the benchmark records do not vary over time, a large date range can be used as shown in the following table:

PERIOD_START_DT	01/01/1899
PERIOD_END_DT	01/01/3714

- The Benchmarks and Targets table is preconfigured at the contact level. You can define other benchmarks and targets, for example, an Hourly-Total-Hold-Duration benchmark, and these can be added using the extension columns in the data warehouse. For more information on the methodology for storing additional data in the data warehouse, see Chapter 10, “Storing, Extracting, and Loading Additional Data.”
- For each dimension in the Benchmark and Targets fact table, you can decide if a benchmark or target varies by that dimension or not. If you choose to keep a benchmark or target constant over a dimension, you need to supply a question mark (?) as the value for the dimension ID. In addition, the metric needs to be leveled in the repository at the grand-total level of that dimension. This dimension ID also needs to be removed from the join in the SQL statement in the `M_CC_ORGLOC_A1_EXTRACT_SERVICE_LEVEL` mapping. If you choose to vary a benchmark or target by a dimension, you need to provide benchmark or target for each value of the dimension.
- The `FORECAST_CNTCT_CNT` table in the source file is preconfigured to forecast the number of calls for a day for a combination of dimensions.

The Benchmarks and Targets table is preconfigured with the smallest effective date range of a day. To changing the grain to be hourly, perform the following procedure.

To configure the grain of the Benchmarks and Targets table

1. Add the Hour Key to the `IA_CC_BMK_TGT` fact table.
2. Modify the `M_F_CC_BMK_TGT_EXTRACT` and `M_F_CC_BMK_TGT_LOAD` mappings to populate the Hour Key based on the Period Start Date and Period End Date.

   These dates need to fall on the hour boundaries and not in the middle of an hourly interval.

3. Modify the `M_CC_ORGLOC_A1_EXTRACT_SERVICE_LEVEL` mapping SQL statement to now join also on Hour Key to the Benchmarks and Targets table.
4. Modify the metadata in the repository to include the new physical and logical joins to the `IA_HOUR_OF_DAY` dimension.
5. Set the content pane settings on the fact table to the newly added Hour (Time) dimension.

To set up the Benchmarks and Targets table

1. Open the `file_cc_bmk_tgt.csv` file using Microsoft WordPad or Notepad in the `$pmserver\SrcFiles` folder.
2. Type in your data to load the Benchmarks and Targets.
3. Save and close the file.

Configuring Flags for Siebel Enterprise Contact Center Analytics

Many of the fact and dimension tables within the Siebel Enterprise Contact Center Analytics application use flag fields to provide value-added information pertaining to a contact or contact representative. These flag fields are configurable and include the following:

- CONSULT_ FLAG
- CONFERENCE_FLAG
- PERTINENT_ INFO_FLG
- CNTCT_MTCH_FLAG
- IVR_FLAG

The possible values for these flag fields in the data warehouse tables are Y or N. However, there is a conversion process to set these values. If you input any value other than N in the flat file interface, the flag defaults to Y. On the other hand, if you supply N, then the flag retains N as the flag’s value.

If you want to change this default logic, you can do so by changing the expression clause in the Expression transformation within the extract mapping. For example, if you want to change the default value of flag fields to N, so that any input value for the flag fields in the flat file interface other than Y is set to N, then you have to change the expression clause from:

```
IIF(<name_of_flag_port> != ‘N’, ‘Y’, ‘N’)
```

to:

```
IIF(<name_of_flag_port> != ‘Y’, ‘N’, ‘Y’)
```

See Table 31 for a list of all flags that you can reconfigure, the corresponding Expression transformations and mappings that contains the flag’s default definition, and a description of each flag’s value.

Table 31. Configurable Flag Values and Descriptions

Flag	Applicable Mappings and Expression Transformations	Flag Values	Descriptions
CONSULT_ FLAG	M_F_CNTCTREP_SNP_EXTRACT EXP_CNTCTREP_SNP_EXTRACT	Y	Indicates that the contact representative consulted with other contact representatives during the course of the call or contact.
		N	Indicates that the contact representative did not consult with other contact representatives during the course of the call or contact.
CONFERENCE_FLAG	M_F_CNTCTREP_SNP_EXTRACT EXP_CNTCTREP_SNP_EXTRACT	Y	Indicates that the contact representative conferenced with other contact representatives during the course of the call or contact.
		N	Indicates that the representative did not conference with other representatives during the course of the call or contact.
PERTINENT_ INFO_FLG	M_F_CNTCTREP_SNP_EXTRACT EXP_CNTCTREP_SNP_EXTRACT	Y	Indicates that the pertinent information was available for the contact.
		N	Indicates that the pertinent information was not available for the contact.
CNTCT_MTCH_FLAG	M_F_CNTCTREP_SNP_EXTRACT EXP_CNTCTREP_SNP_EXTRACT	Y	Indicates that the contact was matched with the existing customer data using Customer Entered Digits (CED), such as PIN numbers, account numbers, or social security numbers.
		N	Indicates that the contact could not be matched with the existing customer data using Customer Entered Digits (CED), such as PIN numbers, account numbers, or social security numbers.

Table 31. Configurable Flag Values and Descriptions

Flag	Applicable Mappings and Expression Transformations	Flag Values	Descriptions
IVR_FLAG	M_F_ACD_EVENTS_EXTRACT EXP_ACD_EVENTS_EXTRACT	Y	Indicates that the event associated with the call was recorded in the (IVR) system.
		N	Indicates that the event associated with the call was not recorded in the IVR system.

To configure a flag for Siebel Enterprise Contact Center Analytics

1. In PowerCenter Designer, open the Configuration for Universal Source folder.
2. Open the applicable mapping.

   For example, if you want to change the default logic for the CONSULT_ FLAG, open the M_F_REP_ACTVTS_EXTRACT mapping.
3. Double-click the applicable Expression transformation to display the applicable port.

   For example, if you want to change the logic for the CONSULT_ FLAG port, double-click the EXP_CNTCTREP_SNP_EXTRACT Expression transformation. Locate the CONSULT_ FLAG port.
4. In the Ports tab, modify the default SQL statement for the flag port.

   For example, if you wanted to change the default CONSULT_ FLAG statement:

   ```
   IIF(CONSULT_ FLAG != 'N','Y','N')
   ```

   You can change it as follows:

   ```
   IIF(CONSULT_ FLAG != 'Y','N','Y')
   ```
5. Validate and save your changes to the repository.

Excluding Representative Data from the Contact Representative Aggregate Tables for Post-Load Processing

You may want to exclude certain contact representatives from the aggregation process for various reasons. For example, you may want to exclude representatives who are on vacation, or who are working on special projects.

The default configuration calculates contact-related information for all contact representatives in the enterprise. There are five aggregate tables supplied with the Siebel Enterprise Contact Center Analytics application for improving the performance of the dashboards and reports:

- IA_CC_REP_A1
- IA_CC_REP_A2
- IA_CC_REP_A3
- IA_CC_ORGLOC_A1
- IA_CC_ORGLOC_A2

To exclude data about specific representatives from the aggregation calculation

1. In PowerCenter Designer, open the Configuration for Post Load Processing folder.
2. Open the `M_PLP_CC_REP_ORGLOC_EXTRACT` mapping.
3. Double-click the `EXP_IA_CNTCTREP_SNP` transformation.
4. Locate the `EXCLUSION_IND_FLG` port.
5. This port is preconfigured with a value of N indicating that all rows are included in the aggregates. Change this logic to include your logic to determine which groups of records you want to exclude.
6. Validate and save the mapping to the repository

   NOTE: If you exclude data from an aggregate table, you also need to apply the same filter to the Logical Table Source corresponding to the `IA_CNTCTREP_SNP` base fact table in the repository metadata (Fact—Service—Contact Center Performance logical table). The metrics computed from the base fact tables are now consistent with those computed from the aggregate tables.

14 Configuring Siebel Enterprise Sales Analytics

This chapter describes how to configure certain objects for particular sources to meet your business needs.

It contains the following topics:

Overview of Siebel Enterprise Sales Analytics

The Siebel Enterprise Sales Analytics application allows you to analyze the movement of sales opportunities through different stages of the sales cycle. This analysis includes insight into which items are booked, backlogged, and invoiced. It also provides you with information that allows you to evaluate the sales performance of individual sales representatives or departments. The Siebel Enterprise Sales Analytics application contains the functional area, Orders and Revenue.

The Orders and Revenue functional area consists of orders, invoices, and backlog. 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.

In the Siebel Enterprise Sales Analytics application, two main types of backlog exist:

- Operational
- Financial

The scheduled, unscheduled, delinquent, and blocked backlogs belong to the Operational backlog. Three different sources can populate Bookings and Revenue:

- Oracle 11i
- SAP R/3
- Universal source

Orders and Revenue also requires post-load processing mappings to populate its tables.

Process of Configuring Siebel Enterprise Sales Analytics for SAP R/3

This section contains Siebel Enterprise Sales Analytics configuration points that are specific to SAP R/3. Of the three functional areas in the Siebel Enterprise Sales Analytics application, Orders and Revenue is the only functional area that has prepackaged SAP R/3 business adapters to populate the warehouse tables; therefore, you can not find configuration points for the other two functional areas in this section.

To configure Siebel Enterprise Sales Analytics for SAP R/3, perform the following tasks:

- Configuring the Booking Flag Calculation in the Sales Order Lines Table on page 248
- Tracking Multiple Attribute Changes in Bookings on page 251
- Configuring the Booking Flag Calculation in the Sales Schedule Lines Level on page 252
- Configuring Early and Late Tolerances for Shipments on page 254
- Configuring the Sales Invoice Extract on page 255
- Configuring Order Types for Backlog Calculations on page 255
- Accounting for Negative Values in Orders, Invoices, and Picks on page 256
- Configuring the Date Parameters for the SAP R/3 Parameter File on page 257

Related Topic

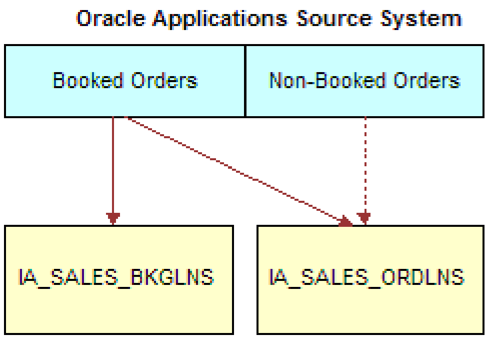
- About Tracking Attribute Changes in Bookings on page 250

Configuring the Booking Flag Calculation in the Sales Order Lines Table

This task is a step in the Process of Configuring Siebel Enterprise Sales Analytics for SAP R/3 on page 248.

Sales order lines are the itemized lines that make up a sales order. This information is stored in the `IA_SALES_ORDLNS` table.

By default, only booked orders are extracted from the SAP R/3 source system as shown in Figure 52. Therefore, all orders loaded into the Sales Order Lines and Bookings tables are flagged as booked (EXT_BOOKING_FLAG = ‘Y’).



By default, only booked orders are loaded into the Sales Order Lines (IA_SALES_ORDLNS) and Booking Lines (IA_SALES_BKGLNS) tables. However, you can also load nonbooked orders into IA_SALES_ORDLNS.

Figure 52. SAP R/3: Default Configuration for Loading Booked Orders

In SAP R/3, the booking flag (EXT_BOOKING_FLAG) is calculated by EXT_SD_DOC_CATEGORY expression. The source for the EXT_SD_DOC_CATEGORY expression is VBAK-VBTYP. The following code is the preconfigured expression for EXT_BOOKING_FLAG:

```
IIF(EXT_SD_DOC_CATEGORY = 'C' OR

EXT_SD_DOC_CATEGORY = 'H' OR

EXT_SD_DOC_CATEGORY = 'L' OR

EXT_SD_DOC_CATEGORY = 'K','Y','N')
```

Using this code, only sales orders in the C, H, L, and K categories are booked and extracted. If you need to extract other document categories, you can change the expression in the MPLT_SAS_SALES_ORDLNS mapplet.

To configure the booking flag calculation in the Sales Order Lines table

1. Open Designer, and open the Configuration for SAP R/3 folder.
2. Expand the MPLT_SAS_SALES_ORDLNS mapplet.
3. Double-click the EXP_SALES_ORDLNS expression transformation

   The Edit Transformation dialog box is displayed
4. Click the Ports tab
5. Edit the EXT_BOOKING_FLAG_OUT expression.

6 Validate and save your changes to the repository.

About Tracking Attribute Changes in Bookings

Changes in booked orders are tracked in the Booking Lines table (IA_SALES_BKGLNS), not in the Sales Order Lines table (IA_SALES_ORDLNS). By default, the only changes tracked in the IA_SALES_BKGLNS table are changes in the ordered amount, ordered quantity, or Booking ID. By default, the Booking ID is defined as:

```
EXT_SALES_ORDER_NUM ||'~'|| TO_CHAR(EXT_SALES_ORDER_ITEM)||'~'||
EXT_MATERIAL_NU||'~'||EXT_CLIENT_CODE
```

Any changes in these fields results in another row in the IA_SALES_BKGLNS table. However, changes in any other fields does not result in a new row; instead, the existing information are overwritten with the changed information. No history is kept for changes to these other field values. If you want to track other changes you can do so. For example, you may want to track changes to the sales representative who is handling the order. The ETL processes are prepackaged to overwrite sales representative changes; however, if you want to retain them, you must add the attribute to the Booking ID definition in the Booking ID expression in the Source Adapter mapplet (MPLT_SAI_SALES_ORDLNS). The following section describes what happens if you modify the Booking ID to include the sales representative.

About Viewing the Data Warehouse Changes by Salesperson ID

Assume you want to track changes to the sales representative for bookings and debookings. You decide to do this to better evaluate each representative's sales performance. To track changes by Salesperson ID, you have to modify the VAR_BOOKING_ID to use the value:

```
EXT_SALES_REP_NUM_VAR
```

The following paragraphs and tables describe what happens in the source system and the IA_SALES_BKGLNS table when you change sales representatives under this scenario.

Day 1: One order is placed with Salesperson 1001. The source system displays the information as shown in Table 32.

Table 32. SAP R/3: Source System Table Row After Day One Activity

Sales Order Number	Sales Order Line Number	Salesperson ID	Quantity	Selling Price	Date
1	1	1001	100	25	1-June-2000

The row in Table 36 is entered into the IA Bookings table (IA_SALES_BKGLNS) as shown in Table 33.

Table 33. SAP R/3: IA_SALES_BKGLNS Table Row After Day One Activity

SALES_ORDER _NUM	SALES_ ORDER_ ITEM	SALESREP _ID	SALES _QTY	NET_DOC _AMT	BOOKING_ ID	BOOKED_ON _DT
1	1	1001	100	2500	1001	1-June-2000

Day 2: Salesperson 1002 takes over this order, replacing Salesperson 1001. Thus, the salesperson associated with the order is changed from 1001 to 1002 in the source system. The row in the source system looks like the row shown in Table 34.

Table 34. SAP R/3: Source System Table Row After Day Two Activity

Sales Order Number	Sales Order Line Number	Salesperson ID	Quantity	Selling Price	Date
1	1	*1002*	100	25	2-June-2000

The Sales Order Lines ADI, which also writes to the booking table, now does a debooking for the old line and inserts a new row into the IA_SALES_BKGLNS booking table. On day two, the row in the IA_SALES_BKGLNS table looks like the row shown in the Table 35.

Table 35. SAP R/3: IA_SALES_BKGLNS Table Row After Day Two Activity

SALES_ORDER_ NUM	SALES_ ORDER_ ITEM	SALESREP_ ID	SALES _QTY	NET_D OC_A MT	BOOKING_I D	BOOKED_ON _DT
1	1	1001	100	2500	1001	1-June-2000
1	1	1001	-100	-2500	1001	2-June-2000
1	1	1002	100	2500	1002	2-June-2000

Tracking Multiple Attribute Changes in Bookings

This task is a step in the Process of Configuring Siebel Enterprise Sales Analytics for SAP R/3 on page 248.

When you modify the default VAR_BOOKING_ID column, the SQL statement is configured as follows for SAP R/3:

```
EXT_SALES_ORDER_NUM ||'~'|| TO_CHAR(EXT_SALES_ORDER_ITEM)||'~'||
EXT_MATERIAL_NUM||'~'||EXT_CLIENT_CODE
```

However, if you want to track changes based on more than one attribute, in the SQL statement you must concatenate the attribute column IDs in the VAR_BOOKING_ID column. For example, if you want to track changes in Salespersons and Sold-to-Customer, then concatenate the technical name IDs in the VAR_BOOKING_ID column as follows:

```
EXT_SALES_ORDER_NUM ||'~'|| TO_CHAR(EXT_SALES_ORDER_ITEM)||'~'||
EXT_MATERIAL_NUM||'~'||EXT_SALES_REP_NUM_VAR||'~'||EXT_CLIENT_CODE
```

To track dimensional attribute changes in bookings

1. In PowerCenter Designer, open the Configuration for SAP R/3 folder.
2. Open the MPLT_SAI_SALES_ORDLNS mapplet.
3. Double-click the EXP_SALES_ORDLNS Expression transformation to open the Edit Transformation box.
4. In the Ports tab, edit the expression for the VAR_BOOKING_ID port, and enter the ID of the attribute for which you want to track changes.

   If you want to track changes in multiple attributes, concatenate the IDs of all attributes and put the concatenated value in the VAR_BOOKING_ID column.
5. Validate and save your changes to the repository.

Configuring the Booking Flag Calculation in the Sales Schedule Lines Level

This task is a step in the Process of Configuring Siebel Enterprise Sales Analytics for SAP R/3 on page 248.

Sales schedule lines detail when each order’s items are slated for shipment. Each sales order is broken into sales order lines, and each sales order line can have multiple schedule lines.

For example, you might not have enough stock to fulfill a particular sales order line, therefore you create two schedules to fulfill it. One schedule ships what you currently have in stock, and the other schedule includes enough time for you to manufacture and ship the remaining items of the sales order line. This information is stored in the IA_SALES_SCHLNS table. This topic describes how to modify the type of information stored in this table.

About Loading Bookings at the Schedule Line Level

As initially configured for SAP R/3, bookings are recorded at the Sales Order Line level and the Sales Schedule Line level. For both book levels, the EXT_BOOKING_FLAG is set based on EXT_SD_DOC_CATEGORY. Sales document categories are booked at the Sales Order Line level or the Sales Schedule Line level. Figure 53 shows bookings recorded at the Sales Order Line level.

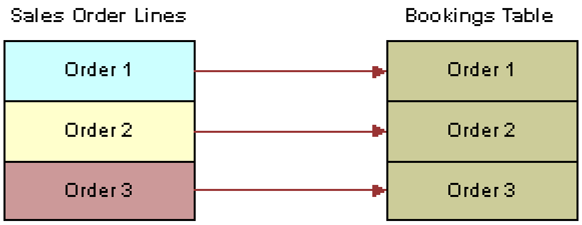


Figure 53. SAP R/3: Bookings at the Sales Order Line Level

Bookings recorded at the Sales Schedule Line level provide a more granular view, as the orders are segmented by schedule line. Bookings recorded at the Schedule Line level provide one row in the Bookings table for each schedule line, as shown in Figure 54.

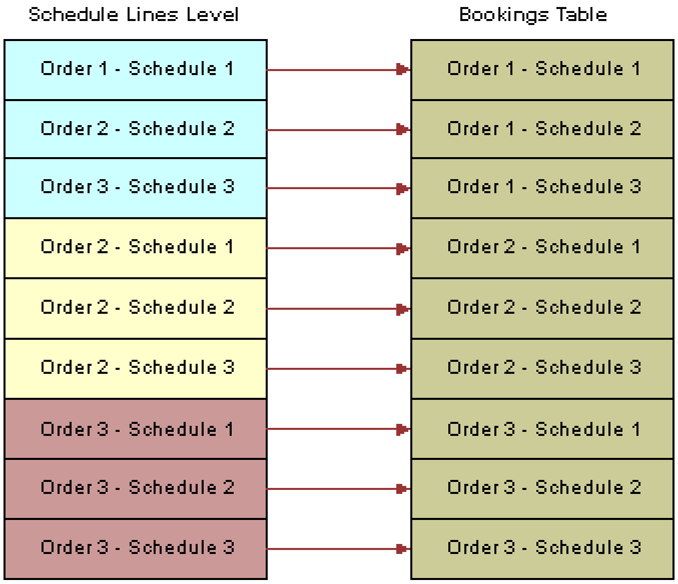


Figure 54. SAP R/3i: Bookings at the Schedule Line Level

There are booking flags in the Sales Order Lines Source Adapter and Sales Schedule Lines Source Adapter mapplets. The EXT_BOOKING_FLAG expression in the Sales Schedule Lines level is preconfigured as follows:

```
IIF(EXP_SD_DOC_CATEGORY = 'E' OR

EXP_SD_DOC_CATEGORY = 'F', 'Y','N')
```

To configure bookings at the sales schedule lines level

1. Open Designer, and open the Configuration for SAP R/3 folder.
2. Expand the `MPLT_SAS_SALES_SCHLNS` mapplet.
3. Double-click the `EXP_SALES_SCHLNS` expression transformation

   The Edit Transformation dialog box is displayed
4. Click the Ports tab
5. Edit the `EXT_BOOKING_FLAG` expression.
6. Validate and save your changes to the repository.

Configuring Early and Late Tolerances for Shipments

This task is a step in the Process of Configuring Siebel Enterprise Sales Analytics for SAP R/3 on page 248.

You can configure the definition of early and late shipments by editing the `EXP_SCHLNS_PICKQTY_UPDATE` expression in the `M_S_SALES_SCHLNS_PICKQTY_LOAD` mapping in SAP R/3. The `M_S_SALES_SCHLNS_PICKQTY_LOAD` mapping compares each of the completed pick lines against their corresponding schedule lines, and then updates the Schedule Lines table with the totals for the pick lines. This comparison allows easy querying against the Schedule Lines table to determine which schedule lines have been shipped on time, early, or late. The logic is prepackaged to flag orders scheduled to ship on a different day than their pick date as either early or late.

However, if you want to redefine the number of days before a pick is considered early or late, you can configure the `EXP_SCHLNS_PICKQTY_UPDATE` Expression transformation.

To configure early and late tolerances for shipments

1. Open Designer, and open the Configuration for SAP R/3 folder.
2. Open the `M_S_SALES_SCHLNS_PICKQTY_LOAD` mapping.
3. Double-click the `EXP_SCHLNS_PICKQTY_UPDATE` Expression transformation to open the Edit Transformation box.
4. In the Ports tab, select the expression for the port to modify and display the SQL statement.

   For example, if you want to allow two days after the scheduled pick date before you flag the pick as late, edit the `VAR_LATE_TIME_TOT` by entering 2.

   - To set the number of days before a pick is flagged as early, edit the SQL statement for the `VAR_EARLY_TIME_TOT` port.
   - To set the number of days before a pick is flagged as late, edit the SQL statement for the `VAR_LATE_TIME_TOT` port.
5. Validate and save your changes to the repository.

Configuring the Sales Invoice Extract

This task is a step in the Process of Configuring Siebel Enterprise Sales Analytics for SAP R/3 on page 248.

Sales invoice lines are payments for items ordered by a customer. This information is stored in the IA_SALES_IVCLNS table. This topic describes how to modify the type of information stored in this table.

By default, the Siebel Enterprise Sales Analytics application is configured to extract completed sales invoices when performing the Sales Invoice data extract. In SAP R/3, the VBRK-RFBSK = 'C' filter completes an invoice.

To extract incomplete sales invoices, and complete invoices, remove the extract filter statement.

To remove the extract filter for sales invoices

1. Open Designer, and open the Configuration for SAP R/3 folder.
2. Open a Sales Invoice Lines Business Component mapplet. Modify both the regular extract mapplet (MPLT_BCS_SALES_IVCLNS) and the primary extract mapplet (MPLT_BCS_SALES_IVCLNS_PRIMARY), and the partner extract mapplet (MPLT_BCS_STAGE_SALES_IVCLNS_PARTNER).
3. Open the Source Qualifier and click the Property tab
4. Remove the VBRK-RFBSK = 'C' filter.
5. Validate and save your changes to the repository.

Configuring Order Types for Backlog Calculations

This task is a step in the Process of Configuring Siebel Enterprise Sales Analytics for SAP R/3 on page 248.

The BACKLOG_FLAG in the IA_SALES_ORDLNS table is also used to identify which sales orders are eligible for backlog calculations. The BACKLOG_FLAG value is derived from the Sales Document Category.

You can add or remove Sales Orders types. Valid values for the Backlog Flag are Y and N. The following code is the preconfigured expression for the backlog calculation:

```
IIF(ISNULL(EXT_REJECTION_CODE),

IIF(EXT_SD_DOC_CATEGORY = 'C' OR

EXT_SD_DOC_CATEGORY = 'E' OR

EXT_SD_DOC_CATEGORY = 'F' OR

EXT_SD_DOC_CATEGORY = 'G' OR

EXT_SD_DOC_CATEGORY = 'H' OR
EXT_SD_DOC_CATEGORY = 'I',

IIF(DOC_DELIVERY_FLAG = 'Y' AND ITEM_DELIVERY_FLAG = 'Y', 'Y', 'N'), 'N'), 'N')
```

To edit the Backlog flag

1 Open Designer, and open the Configuration for SAP R/3 folder.

2 Open the MPLT_SAS_SALES_ORDLNS mapplet.

3 Open the Expression transformation EXP_SALES_ORDLNS in the MPLT_SALES_ORDLNS mapplet to edit the Backlog Flag port.

4 Modify the statement in the BACKLOG_FLAG port.

5 Repeat Step 2 through to Step 4, for the EXP_SALES_SCHLNS expression in the MPLT_SALES_SCHLNS mapplet.

6 Validate the expression, and save your changes to the repository.

Accounting for Negative Values in Orders, Invoices, and Picks

This task is a step in the Process of Configuring Siebel Enterprise Sales Analytics for SAP R/3 on page 248.

The Siebel Customer-Centric Enterprise Warehouse is preconfigured not to use negative values in the quantity or amount columns for the IA_SALES_IVCLNS or IA_SALES_ORDLNS tables. These two columns are preconfigured with a value of 1.0. However, you can change these values to negative values by using the VAR_NEGATIVE_SIGN and VAR_NEGATIVE_SIGN_QTY columns.

For example, to account for a negative value for the BV and ZUN document types, you can use the following conditional statement to define the VAR_NEGATIVE_SIGN and VAR_NEGATIVE_SIGN_QTY columns:

```
IIF(EXT_DOCUMENT_TYPE = 'BV' OR

EXT_DOCUMENT_TYPE = 'ZUN' , -1.0, 1.0)
```

To configure mapplets to account for negative values

1 Open Designer, and open the Configuration for SAP R/3 folder.

2 Open the applicable Source Adapter mapplet.

3 Double-click the Expression transformation to open the Edit Transformation box.

4 In the Ports tab, edit the VAR_NEGATIVE_SIGN and the VAR_NEGATIVE_SSIGN_QTY ports.

5 Validate and save your changes to the repository.

Configuring the Date Parameters for the SAP R/3 Parameter File

This task is a step in the Process of Configuring Siebel Enterprise Sales Analytics for SAP R/3 on page 248.

You need to set the PARM_NVALUE_1 value in the `file_parameters_sapr3.csv` file to the number of days that you expect your orders to be open. This configuration is necessary for ETL as SAP R/3 does not update the last changed date for a table when a user updates that table.

The PARM_NVALUE_1 value is used by the following session parameter codes:

- S_M_S_STAGE_SALES_ORDHDR_BUSN_DATA_EXTRACT:INCRDATE
- S_M_S_STAGE_SALES_ORDHDR_PARTNER_EXTRACT:INCRDATE
- S_M_S_STAGE_SALES_ORDLNS_BUSN_DATA_EXTRACT:INCRDATE
- S_M_S_STAGE_SALES_ORDLNS_PARTNER_EXTRACT:INCRDATE
- S_M_S_STAGE_SALES_PICK_PARTNER_EXTRACT:INCRDATE
- S_M_S_SALES_PCKLNS_EXTRACT:INCRDATE
- S_M_S_SALES_SCHLNS_EXTRACT:INCRDATE
- S_M_S_SALES_ORDLNS_EXTRACT:INCRDATE
- S_M_S_SALES_SHPMTS_EXTRACT:INCRDATE

NOTE: There are always orders that are open for a long period of time. To make sure that ETL captures changes to these orders, it is recommended that you occasionally set the PARM_NVALUE_1 value to a value equivalent to that period of time.

To modify the CSV parameter file

1. Open the `file_parameters_sapr3.csv` file using Microsoft WordPad or Notepad in the `$pmserver\SrcFiles` folder.
2. Search for the column labeled PARM_NVALUE_1.
3. Change the default date value to the date value you require.
4. Save your file.

Process of Configuring Siebel Enterprise Sales Analytics for Oracle 11i

This section contains Siebel Enterprise Sales Analytics configuration points that are specific to Oracle 11i. Of the three functional areas in the Siebel Enterprise Sales Analytics application, Orders and Revenue is the only functional area that has prepackaged Oracle 11i business adapters to populate the warehouse tables; therefore, you can not find configuration points for the other two functional areas in this section.

To configure Siebel Enterprise Sales Analytics for Oracle 11i, perform the following tasks:

- Configuring Sales Order Lines Data Storage on page 258
- Tracking Multiple Attribute Changes in Bookings on page 261
- Configuring Sales Schedule Lines Data Storage on page 262
- Configuring Early and Late Tolerances for Shipments on page 264
- Configuring Sales Invoice Lines Data Storage on page 265
- Configuring Different Types of Backlog Calculations on page 265
- Accounting for Negative Values in Orders, Invoices, and Picks on page 267

Related Topic

- About Tracking Attribute Changes in Bookings on page 259

Configuring Sales Order Lines Data Storage

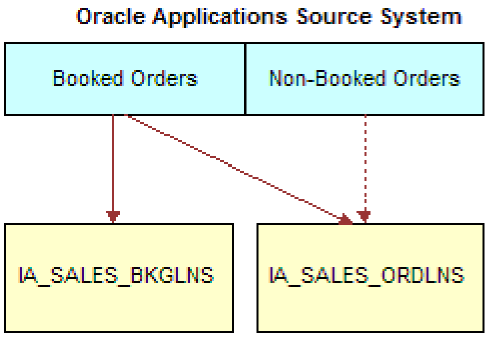
This task is a step in the Process of Configuring Siebel Enterprise Sales Analytics for Oracle 11i on page 257.

Sales order lines are the itemized lines that make up a sales order. This information is stored in the IA_SALES_ORDLNS table. This topic describes how to modify the type of information stored in this table.

About the Handling of Booked and Nonbooked Orders in the Order Lines and Bookings Table

By default, only booked orders are extracted from the Oracle 11i source system as shown in Figure 55. Therefore, all orders loaded into the Sales Order Lines and Bookings tables are flagged as booked (EXT_BOOKING_FLAG = ‘Y’).

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



By default, only booked orders are loaded into the Sales Order Lines (IA_SALES_ORDLNS) and Booking Lines (IA_SALES_BKGLNS) tables. However, you can also load nonbooked orders into IA_SALES_ORDLNS.

Figure 55. Oracle 11i: Default Configuration for Loading Booked Orders

To include nonbooked orders in the Sales Order Lines table

1. In PowerCenter Workflow Manager, open the Configuration for Oracle Applications v11i folder.
2. Expand the Oracle 11i Enterprise Sales Application workflow.
3. Expand the O_FACTS_EXTRACT worklet and the O_FACT_EXTRACT_SALES worklet.
4. Double-click the S_M_I_SALES_ORDLNS_SCHLNS_EXTRACT session to open the Edit Tasks box.
5. In the Transformations tab, edit the MPLT_BCI_SALES_ORDLNS SQL statement.

   Remove the OE_ORDER_LINES_ALL.BOOKED_FLAG='Y' filter in the SQL statement to include nonbooked orders in the Sales Order lines table.
6. Validate and save your changes to the repository.

About Tracking Attribute Changes in Bookings

Changes in booked orders are tracked in the Booking Lines table (IA_SALES_BKGLNS), not in the Sales Order Lines table (IA_SALES_ORDLNS). By default, the only changes tracked in the IA_SALES_BKGLNS table are changes in the ordered amount, ordered quantity, or Booking ID. By default, the Booking ID is defined as:

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

Any changes in these fields results in another row in the IA_SALES_BKGLNS table. However, changes in any other fields does not result in a new row; instead, the existing information are overwritten with the changed information. No history is kept for changes to these other field values. If you want to track other changes you can do so. For example, you may want to track changes to the sales representative who is handling the order. The ETL processes are prepackaged to overwrite sales representative changes; however, if you want to retain them, you must add the attribute to the Booking ID definition in the Booking ID expression in the Source Adapter mapplet (MPLT_SAI_SALES_ORDLNS). The following section describes what happens if you modify the Booking ID to include the sales representative.

About Viewing the Data Warehouse Changes by Salesperson ID

Assume you want to track changes to the sales representative for bookings and debookings. You decide to do this to better evaluate each representative's sales performance. To track changes by Salesperson ID, you have to modify the VAR_BOOKING_ID to use the value:

```
TO_CHAR(INP_SALESREP_ID)
```

The following paragraphs and tables describe what happens in the source system and the IA_SALES_BKGLNS table when you change sales representatives under this scenario.

Day 1: One order is placed with Salesperson 1001. The source system displays the information as shown in Table 36.

Table 36. Oracle 11i: Source System Table Row After Day One Activity

Sales Order Number	Sales Order Line Number	Salesperson ID	Quantity	Selling Price	Date
1	1	1001	100	25	1-June-2000

The row in Table 36 is entered into the IA Bookings table (IA_SALES_BKGLNS) as shown in Table 37.

Table 37. Oracle 11i: IA_SALES_BKGLNS Table Row After Day One Activity

SALES_ORDER_NUM	SALES_ ORDER_ ITEM	SALESREP_ID	SALES_QTY	NET_DOC_AMT	BOOKING_ ID	BOOKED_ON_DT
1	1	1001	100	2500	1001	1-June-2000

Day 2: Salesperson 1002 takes over this order, replacing Salesperson 1001. Thus, the salesperson associated with the order is changed from 1001 to 1002 in the source system. The row in the source system looks like the row shown in Table 38.

Table 38. Oracle 11i: Source System Table Row After Day Two Activity

Sales Order Number	Sales Order Line Number	Salesperson ID	Quantity	Selling Price	Date
1	1	*1002*	100	25	2-June-2000

The Sales Order Lines ADI, which also writes to the booking table, now does a debooking for the old line and inserts a new row into the IA_SALES_BKGLNS booking table. On day two, the row in the IA_SALES_BKGLNS table looks like the row shown in the Table 39.

Table 39. Oracle 11i: IA_SALES_BKGLNS Table Row After Day Two Activity

SALES_ORDER_NUM	SALES_ORDER_ITEM	SALESREP_ID	SALES_QTY	NET_DOC_AMT	BOOKING_ID	BOOKED_ON_DT
1	1	1001	100	2500	1001	1-June-2000
1	1	1001	-100	-2500	1001	2-June-2000
1	1	1002	100	2500	1002	2-June-2000

Tracking Multiple Attribute Changes in Bookings

This task is a step in the Process of Configuring Siebel Enterprise Sales Analytics for Oracle 11i on page 257.

When you modify the default VAR_BOOKING_ID column, the SQL statement is configured as follows for Oracle 11i:

```
TO_CHAR(INP_LINE_ID)||'~'||TO_CHAR(INP_INV_ITEM_ID)||'~'||to_char(INP_WAREHOUSE_ID
)
```

However, if you want to track changes based on more than one attribute, in the SQL statement you must concatenate the attribute column IDs in the VAR_BOOKING_ID column. For example, if you want to track changes in Salespersons and Sold-to-Customer, then concatenate the technical name IDs in the VAR_BOOKING_ID column as follows:

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

To track dimensional attribute changes in bookings

1 In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.

2 Open the MPLT_SAI_SALES_ORDLNS mapplet.

3 Double-click the EXP_SALES_ORDLNS Expression transformation to open the Edit Transformation box.

4 In the Ports tab, edit the expression for the VAR_BOOKING_ID port, and enter the ID of the attribute for which you want to track changes.

   If you want to track changes in multiple attributes, concatenate the IDs of all attributes and put the concatenated value in the VAR_BOOKING_ID column.

5 Validate and save your changes to the repository.

Configuring Sales Schedule Lines Data Storage

This task is a step in the Process of Configuring Siebel Enterprise Sales Analytics for Oracle 11i on page 257.

Sales schedule lines detail when each order's items are slated for shipment. Each sales order is broken into sales order lines, and each sales order line can have multiple schedule lines.

For example, you might not have enough stock to fulfill a particular sales order line, therefore you create two schedules to fulfill it. One schedule ships what you currently have in stock, and the other schedule includes enough time for you to manufacture and ship the remaining items of the sales order line. This information is stored in the IA_SALES_SCHLNS table. This topic describes how to modify the type of information stored in this table.

About Loading Bookings at the Schedule Line Level

As initially configured for Oracle 11i, bookings are recorded at the Sales Order Line level. For each booked order, there is at least one row in the Bookings table, as shown in Figure 56. By default, the EXT_BOOKING_FLAG is set to N in the Schedule Lines MPLT_SAI_SALES_SCHLNS Source Adapter mapplet.

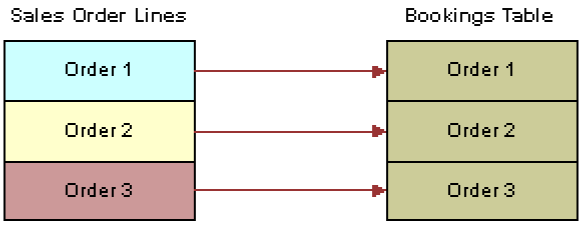


Figure 56. Oracle 11i: Bookings at the Order Line Level

Bookings may be recorded at the Sales Schedule Line level instead of the Sales Order Line level. At the Sales Schedule Line level, bookings provide a more granular view, as the orders are segmented by schedule line. Bookings recorded at the Schedule Line level provide one row in the Bookings table for each schedule line, as shown in Figure 57. The Booking flag (EXT_BOOKING_FLAG) is set to Y in the Schedule Lines Source Adapter mapplet.

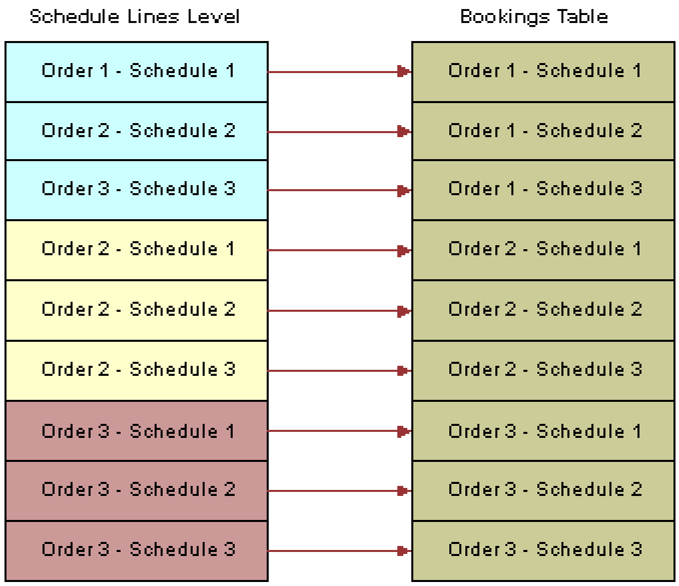


Figure 57. Oracle 11i: Bookings at the Schedule Line Level

About Booking Flags

There are booking flags in both the Sales Order Lines Source Adapter mapplet and in the Sales Schedule Lines Source Adapter mapplet. Table 40 lists how the analytic system reacts based on the booking flag settings. Usually, one of the flags is set to Y and the other to N, resulting in the booking logic preferred. However, if both mapplets' flags are set to Y, the system records bookings at the default Order Lines level. If both mapplets are set to N, no booking logic is enforced.

Table 40. Sales Order Lines Booking Flag Settings

Sales Order Lines Source Adapter—Booking Flag Value	Sales Schedule Lines Source Adapter—Booking Flag Value	Result
'Y'	'N'	Bookings are done at the order line level.
'Y'	'Y'	Bookings are done at the order line level.
'N'	'Y'	Bookings are done at the schedule line level.
'N'	'N'	No booking logic.

To configure bookings at the sales schedule lines level

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. Open the MPLT_SAI_SALES_SCHLNS mapplet.
3. Double-click the EXP_SALES_SCHLNS Expression transformation to open the Edit Transformation box.
4. In the Ports tab, edit the expression for EXT_BOOKING_FLAG to row bookings at the Schedule Line level.

   Modify the EXT_BOOKING_FLAG definition by inserting Y.
5. In addition, change the EXT_BOOKING_FLAG in the Sales Order Lines Source Adapter to N or the system records bookings at the Order Lines level by default.
6. Validate and save your changes to the repository.

Configuring Early and Late Tolerances for Shipments

This task is a step in the Process of Configuring Siebel Enterprise Sales Analytics for Oracle 11i on page 257.

You can configure the definition of early and late shipments by editing the EXP_SALES_SCHLNS_PICKQTY expression in the M_I_SALES_SCHLNS_PICKQTY_LOAD mapping in Oracle 11i. The M_I_SALES_SCHLNS_PICKQTY_LOAD mapping compares each of the completed pick lines against their corresponding schedule lines, and then updates the Schedule Lines table with the totals for the pick lines. This comparison allows easy querying against the Schedule Lines table to determine which schedule lines have been shipped on time, early, or late. The logic is prepackaged to flag orders scheduled to ship on a different day than their pick date as either early or late.

However, if you want to redefine the number of days before a pick is considered early or late, you can configure the EXP_SALES_SCHLNS_PICKQTY Expression transformation.

To configure early and late tolerances for shipments

In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.

1. Open the M_I_SALES_SCHLNS_PICKQTY_LOAD mapping.
2. Double-click the EXP_SALES_SCHLNS_PICKQTY Expression transformation to open the Edit Transformation box.
3. In the Ports tab, select the expression for the port to modify and display the SQL statement.

   For example, if you want to allow two days after the scheduled pick date before you flag the pick as late, edit the VAR_LATE_TIME_TOT by entering 2.

   - To set the number of days before a pick is flagged as early, edit the SQL statement for the VAR_EARLY_TIME_TOT port.
   - To set the number of days before a pick is flagged as late, edit the SQL statement for the VAR_LATE_TIME_TOT port.

4 Validate and save your changes to the repository.

Configuring Sales Invoice Lines Data Storage

This task is a step in the Process of Configuring Siebel Enterprise Sales Analytics for Oracle 11i on page 257.

Sales invoice lines are payments for items ordered by a customer. This information is stored in the `IA_SALES_IVCLNS` table. This topic describes how to modify the type of information stored in this table.

Configuring the Sales Invoice Extract

By default, the Siebel Enterprise Sales Analytics application is configured to extract completed sales invoices when performing the Sales Invoice data extract. Oracle 11i uses a flag to indicate whether a sales invoice is complete. In particular, completed sales invoices are those where the `RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG = Y` in Oracle 11i.

To extract incomplete sales invoices, as well as complete invoices, remove the extract filter statement.

To remove the extract filter for sales invoices

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. Open a Sales Invoice Lines Business Component mapplet. Modify both the regular extract mapplet (`MPLT_BCI_SALES_IVCLNS`) and the primary extract mapplet (`MPLT_BCI_SALES_IVCLNS_PRIMARY`).
3. Open the Source Qualifier to edit the SQL statement in the SQL Query and User Defined Join fields.
4. In the User Defined Join field and in the SQL Query field, remove the statement:

   ```
   AND RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG(+) = Y
   ```

5. Validate and save your changes to the repository.

Configuring Different Types of Backlog Calculations

This task is a step in the Process of Configuring Siebel Enterprise Sales Analytics for Oracle 11i on page 257.

Backlog information is stored in the `IA_SALES_IA_BLGLNS` and `IA_SALES_BLGHIS` tables. This topic describes how to modify the type of information stored in these tables. Many types of backlog exist in the Siebel Enterprise Sales Analytics application—financial backlog, operational backlog, delinquent backlog, scheduled backlog, unscheduled backlog, and blocked backlog. Each type of backlog is defined by two particular dates in the sales process; therefore, calculations of backlog hits multiple fact tables.

For example, financial backlog records which items have been ordered but payment has not been received. Thus, to calculate the number of financial backlog items, you use the Sales Order Lines table (to determine which items have been ordered) and the Sales Invoice Lines table (to see which orders have been paid for). Using these two tables, you can determine the number of items and the value of those items that are on financial backlog.

Adding Closed Orders to Backlog Calculations

By default, the Siebel Enterprise Sales Analytics application only extracts open sales orders from the Sales Order Lines table (IA_SALES_ORDLNS) for backlog calculations to populate the Backlog tables (IA_SALES_BLGLNS and IA_SALES_BLGHIS). *Open sales orders* are defined as orders that are not canceled or not complete. The purpose in extracting only open orders is that in most organizations those orders that are closed are no longer a part of backlog. However, if you want to extract sales orders that are marked as closed, you may remove the default filter condition from the extract mapping.

In Oracle 11i, open sales orders are flagged in the source system with one of the following two statuses—S6 and S9. If you want to remove the filter condition, you must remove the condition containing these two source system values.

For example, assume your customer orders ten items. Six items are invoiced and shipped, but four items are placed on operational and financial backlog. This backlog status continues until one of two things happens:

- The items are eventually shipped and invoiced.
- The remainder of the order is canceled.

If you choose to extract sales orders that are flagged as closed, you must remove the condition in the Backlog flag. To do so, use the following procedure.

To remove open order extract filters

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. Open the MPLT_BCI_SALES_BLGLNS mapplet.
3. Open the Source Qualifier transformation to edit the SQL statement in the SQL Query field and the User Defined Join field.
4. Edit the statement by modifying the filter.
5. Validate and save your changes to the repository.

Configuring Order Types for Backlog Calculations

This task is a step in the Process of Configuring Siebel Enterprise Sales Analytics for Oracle 11i on page 257.

The BACKLOG_FLAG in the IA_SALES_ORDLNS table is also used to identify which sales orders are eligible for backlog calculations. By default, all sales order types have their Backlog flag set to Y. As a result, all sales orders are included in backlog calculations.

However, if you wish to process only certain types of sales orders, you must insert a conditional statement for the Backlog flag. Valid values for the Backlog Flag are Y and N.

To edit the Backlog flag

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. Open the MPLT_SAI_SALES_ORDLNS mapplet.
3. Open the Expression transformation EXP_SALES_ORDLNS to edit the Backlog Flag port.
4. Modify the statement in the BACKLOG_FLAG port to include or exclude sales orders from backlog calculations.
5. Validate the expression, and save your changes to the repository.

Accounting for Negative Values in Orders, Invoices, and Picks

This task is a step in the Process of Configuring Siebel Enterprise Sales Analytics for Oracle 11i on page 257.

By default, the Siebel Customer-Centric Enterprise Warehouse does not use negative values in the quantity or amount columns for the IA_SALES_IVCLNS table or the IA_SALES_ORDLNS table. However, you can make these values negative using a column called VAR_NEGATIVE_SIGN. By default, this column has the value 1.0. To make the values negative, modify the column value to be -1.

For example, to account for a negative return value for a Return Material Authorization (RMA) or for a negative value in a credit memo, you can use a conditional statement to define the VAR_NEGATIVE_SIGN column.

Assume that the S14 column in SO_LINES_ALL table has been configured in Oracle 11i to have the value 30 if the order line is a return. You can use this identifier for returned orders as a condition for setting the VAR_NEGATIVE_SIGN column to be -1. To do this, you can modify the VAR_NEGATIVE_SIGN column's definition in the MPLT_SAO_SALES_ORDLNS as follows:

```
DECODE(INP_LINES_S14, 30, -1, 1)
```

For Oracle 11i, the VAR_NEGATIVE_SIGN column is available in the following Source Adapters—MPLT_SAI_SALES_ORDLNS, MPLT_SAI_SALES_IVCLNS and MPLT_SAI_SALES_PCKLNS.

In Oracle 11i the VAR_NEGATIVE_SIGN column's value is set based on the type of order line.

To configure mapplets to account for negative values

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. Open the applicable Source Adapter mapplet.
3. Double-click the Expression transformation to open the Edit Transformation box.

4 In the Ports tab, edit the `VAR_NEGATIVE_SIGN` port.

   For example, if the S14 column in `SO_LINES_ALL` table has been configured in Oracle 11i to have the value 30 if the order line is a return, then you can use this identifier for returned orders as a condition for setting the `VAR_NEGATIVE_SIGN` column to be -1. To do so, you would set the `VAR_NEGATIVE_SIGN` column's definition as follows:

   ```
   DECODE(INP_LINES_S14, 30, -1, 1)
   ```

5 Validate and save your changes to the repository.

Process of Aggregating Siebel Enterprise Sales Analytics Tables

This section contains Siebel Enterprise Sales Analytics configuration points for aggregating the Sales Invoice Lines and Sales Order Lines tables.

To aggregate the Sales Invoice Lines and Sales Order Lines tables, perform the following tasks:

- Configuring the Sales Invoice Lines Aggregate Table on page 270
- Configuring the Sales Order Lines Aggregate Table on page 274

Related Topics

- About Configuring the Sales Invoice Lines Aggregate Table on page 268
- About Configuring the Sales Order Lines Aggregate Table on page 273

About Configuring the Sales Invoice Lines Aggregate Table

The Sales Invoice Lines aggregate table (IA_SALES_IVCLNS_A1) is used to capture information about the invoices issued for your sales orders. You need to configure the Sales Invoice Lines aggregate table in order to run initial ETL and incremental ETL.

For your initial ETL run, you need to configure the GRAIN parameter for the time aggregation level in the Sales Invoice Lines aggregate fact table.

For the incremental ETL run, you need to configure the time aggregation level 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 Sales Invoice Lines table for your incremental run:

- GRAIN
- SOURCE_ID

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

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

Table 41 lists the values for the SOURCE_ID parameter. The value of this parameter is preconfigured to reflect the ETL mapping's folder.

Table 41. Source System Parameter Values

Source System	Value
Oracle 11i	OAP11I
SAP R/3	SAPR3
Universal	GENERIC

NOTE: You can change the default value for the Source_ID parameter if you use multiple instances of the same source system. For example, you can run multiple instances of SAP R/3 and use separate Source IDs for each instance. You can name the first instance SAPR3_1, the second instance SAPR3_2, and so on.

The Sales Invoice Lines aggregate table is fully loaded from the base table in the initial ETL run. The table can grow to millions of records. Thus, the Sales Invoice aggregate table is not fully reloaded from the base table after each incremental ETL run. Siebel Customer-Centric Enterprise 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. Siebel Customer-Centric Enterprise Warehouse finds the records to be deleted in the base table since the last ETL run, and loads them into the NU_SALES_IVCLNS table. The measures in these records are multiplied by (-1). The mapping responsible for this task is suffixed with PRE_D, and it is run before the records are deleted from the base table. The mapping is run in the source-specific workflow.
2. Siebel Customer-Centric Enterprise Warehouse finds the records to be updated in the base table since the last ETL run, and loads them into the NU_SALES_IVCLNS table. The measures in these records are multiplied by (-1). The mapping responsible for this task is suffixed with PRE_U, and it is run before the records are updated in the base table. It is run in the source-specific workflow.
3. Siebel Customer-Centric Enterprise Warehouse finds the inserted or updated records in the base table since the last ETL run, and loads them into the NU_SAELS_IVCLNS table, without changing their sign. The mapping responsible for this task is suffixed with POST, 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. Siebel Customer-Centric Enterprise Warehouse aggregates the NU_SALES_IVCLNS table, and joins it with the IA_SLS_IVCLNS_A1 aggregate table to update existing or insert new buckets to the aggregate table. This step is part of the post load-processing workflow, and the mapping is suffixed with INCR.

Configuring the Sales Invoice Lines Aggregate Table

This task is a step in the Process of Aggregating Siebel Enterprise Sales Analytics Tables on page 268.

To load the Sales Invoice Lines aggregate table (IA_SLS_IVCLNS_A1), you need to configure the post-load-processing parameter file and the source system parameter files, and run the initial and then the incremental workflows.

To configure the file_parameters_plp.csv parameters file

1. Open the file_parameters_plp.csv file using Microsoft WordPad or Notepad in the $pmserver\srcfiles folder.
2. Replace the default PARM_SVALUE2 values with your new values.

   For a list of values for each parameter see the About Configuring the Sales Invoice Lines Aggregate Table on page 268.

   The default values are shown in the following table:

PARM_TYPE	PARM_CODE	PARM_SVALUE_1	PARM_SVALUE_2	PARM_SVALUE_3
PLP	S_M_PLP_SALES_IVCLNS_POST:GRAIN	S	MONTH	MPLT_ADI_SALES_IVCLNS_PRE_POST
PLP	S_M_PLP_SLS_IVCLNS_A1_AGG_INIT:GRAIN	S	'MONTH'	—

   NOTE: You need to use single quotes for the S_M_PLP_SLS_IVCLNS_A1_AGG_INIT:GRAIN session value.

3. Save and close the file.

To configure the Sales Invoice Lines aggregate table for Oracle 11i

1. Open the file_parameters_ora11i.csv file using Microsoft WordPad or Notepad in the $pmserver\srcfiles folder.

2 Replace the default PARM_SVALUE2 values with your new values.

For a list of values for each parameter see the About Configuring the Sales Invoice Lines Aggregate Table on page 268.

The default values are shown in the following table:

PARM_TYPE	PARM_CODE	PARM_SVALUE_1	PARM_SVALUE_2	PARM_SVALUE_3
ORA11I	S_M_I_SALES_IVCLNS_PRE_D:GRAIN	S	MONTH	MPLT_ADI_SALES_IVCLNS_PRE_POST
ORA11I	S_M_I_SALES_IVCLNS_PRE_D:SOURCE_ID	S	'OAP11I'	—
ORA11I	S_M_I_SALES_IVCLNS_PRE_U:GRAIN	S	MONTH	MPLT_ADI_SALES_IVCLNS_PRE_POST
ORA11I	S_M_I_SALES_IVCLNS_PRE_U:SOURCE_ID	S	'OAP11I'	—

NOTE: You need to use single quotes for the S_M_I_PURCH_RCPTS_PRE_D:SOURCE_ID and S_M_I_PURCH_RCPTS_PRE_U:SOURCE_ID session values.

3 Save and close the file.

To configure the Sales Invoice Lines aggregate table for SAP R/3

1 Open the `file_parameters_sapr3.csv` file using Microsoft WordPad or Notepad in the `$pmserver\srcfiles` folder.

2 Replace the default PARM_SVALUE2 values with your new values.

For a list of values for each parameter see the About Configuring the Sales Invoice Lines Aggregate Table on page 268.

The default values are shown in the following table:

PARM_TYPE	PARM_CODE	PARM_SVALUE_1	PARM_SVALUE_2	PARM_SVALUE_3
SAPR3	S_M_S_SALES_IVCLNS_PRE_D:GRAIN	S	MONTH	MPLT_ADI_SALES_IVCLNS_PRE_POST
SAPR3	S_M_S_SALES_IVCLNS_PRE_D:SOURCE_ID	S	‘SAPR3'	—
SAPR3	S_M_S_SALES_IVCLNS_PRE_U:GRAIN	S	MONTH	MPLT_ADI_SALES_IVCLNS_PRE_POST
SAPR3	S_M_S_SALES_IVCLNS_PRE_U:SOURCE_ID	S	‘SAPR3'	—

NOTE: You need to use single quotes for the S_M_S_SALES_IVCLNS_PRE_D:SOURCE_ID and S_M_S_SALES_IVCLNS_PRE_U:SOURCE_ID session values.

3 Save and close the file.

To configure the Sales Invoice Lines aggregate table for Universal Source

1 Open the `file_parameters_univ.csv` file using Microsoft WordPad or Notepad in the `$pmserver\srcfiles` folder.

2 Replace the default PARM_SVALUE2 values with your new values.

For a list of values for each parameter see the About Configuring the Sales Invoice Lines Aggregate Table on page 268.

The default values are shown in the following table:

PARM_TYPE	PARM_CODE	PARM_SVALUE_1	PARM_SVALUE_2	PARM_SVALUE_3
UNIV	S_M_F_SALES_IVCLNS_PRE_U:GRAIN	S	MONTH	MPLT_ADI_SALES_IVCLNS_PRE_POST
UNIV	S_M_F_SALES_IVCLNS_PRE_U:SOURCE_ID	S	'GENERIC'	—

NOTE: You need to use single quotes for the S_M_F_SALES_IVCLNS_PRE_U:SOURCE_ID session value.

3 Save and close the file.

About Configuring the Sales Order Lines Aggregate Table

The Sales Order Lines aggregate table (IA_SLS_ORDLNS_A1) is used to capture information about the order lines issued for your sales orders. You need to configure the Sales Order Lines aggregate table in order to run initial ETL and incremental ETL.

For your initial ETL run, you need to configure the GRAIN parameter for the time aggregation level in the Sales Order Lines aggregate fact table.

For the incremental ETL run, you need to configure the time aggregation level 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 Sales Order Lines table for your incremental run:

- GRAIN
- SOURCE_ID

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

- DAY
- WEEK
- MONTH
- QUARTER

- YEAR

Table 42 lists the values for the SOURCE_ID parameter. The value of this parameter is preconfigured to reflect the ETL mapping's folder.

Table 42. Source System Parameter Values

Source System	Value
Oracle 11i	OAP11I
SAP R/3	SAPR3
Universal	GENERIC

NOTE: You can change the default value for the Source ID parameter if you use multiple instances of the same source system. For example, you can run multiple instances of SAP R/3 and use separate Source IDs for each instance. You can name the first instance SAPR3_1, the second instance SAPR3_2, and so on.

The Sales Order Lines aggregate table is fully loaded from the base table in the initial ETL run. The table can grow to millions of records. Thus, the Sales Order aggregate table is not fully reloaded from the base table after each incremental ETL run. Siebel Customer-Centric Enterprise 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. Siebel Customer-Centric Enterprise Warehouse finds the records to be deleted in the base table since the last ETL run, and loads them into the `NU_SALES_ORDLNS` table. The measures in these records are multiplied by (-1). The mapping responsible for this task is suffixed with PRE_D, and it is run before the records are deleted from the base table. The mapping is run in the source-specific workflow.
2. Siebel Customer-Centric Enterprise Warehouse finds the records to be updated in the base table since the last ETL run, and loads them into the `NU_SALES_ORDLNS` table. The measures in these records are multiplied by (-1). The mapping responsible for this task is suffixed with PRE_U, and it is run before the records are updated in the base table. It is run in the source-specific workflow.
3. Siebel Customer-Centric Enterprise Warehouse finds the inserted or updated records in the base table since the last ETL run, and loads them into the `NU_SAELS_ORDLNS` table, without changing their sign. The mapping responsible for this task is suffixed with POST, 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. Siebel Customer-Centric Enterprise Warehouse aggregates the `NU_SALES_ORDLNS` table, and joins it with the `IA_SLS_ORDLNS_A1` aggregate table to update existing or insert new buckets to the aggregate table. This step is part of the post load-processing workflow, and the mapping is suffixed with INCR.

Configuring the Sales Order Lines Aggregate Table

This task is a step in the Process of Aggregating Siebel Enterprise Sales Analytics Tables on page 268.

To load the Sales Order Lines aggregate table (`IA_SLS_ORDLNS_A1`), you need to configure the post-load-processing parameter file and the source system parameter files, and run the initial and then the incremental workflows.

To configure the file_parameters_plp.csv parameters file

1. Open the `file_parameters_plp.csv` file using Microsoft WordPad or Notepad in the `$pmserver\srcfiles` folder.
2. Replace the default PARM_SVALUE2 values with your new values.

   For a list of values for each parameter see the About Configuring the Sales Order Lines Aggregate Table on page 273.

   The default values are shown in the following table:

PARM_TYPE	PARM_CODE	PARM_SVALUE_1	PARM_SVALUE_2	PARM_SVALUE_3
PLP	S_M_PLP_SALES_ORDLNS_POST:GRAIN	S	MONTH	MPLT_ADI_SALES_ORDLNS_PRE_POST
PLP	S_M_PLP_SLS_IVCLNS_A1_AGG_INIT:GRAIN	S	'MONTH'	—

   NOTE: You need to use single quotes for the S_M_PLP_SLS_IVCLNS_A1_AGG_INIT:GRAIN session value.

3. Save and close the file.

To configure the Sales Order Lines aggregate table for Oracle 11i

1. Open the `file_parameters_ora11i.csv` file using Microsoft WordPad or Notepad in the `$pmserver\srcfiles` folder.

2 Replace the default PARM_SVALUE2 values with your new values.

For a list of values for each parameter see the About Configuring the Sales Order Lines Aggregate Table on page 273.

The default values are shown in the following table:

PARM_TYPE	PARM_CODE	PARM_SVALUE_1	PARM_SVALUE_2	PARM_SVALUE_3
ORA11I	S_M_I_SALES_ORDLNS_PRE_D:GRAIN	S	MONTH	MPLT_ADI_SALES_ORDLNS_PRE_POST
ORA11I	S_M_I_SALES_ORDLNS_PRE_D:SOURCE_ID	S	'OAP11I'	—
ORA11I	S_M_I_SALES_ORDLNS_PRE_U:GRAIN	S	MONTH	MPLT_ADI_SALES_ORDLNS_PRE_POST
ORA11I	S_M_I_SALES_ORDLNS_PRE_U:SOURCE_ID	S	'OAP11I'	—

NOTE: You need to use single quotes for the S_M_I_SALES_ORDLNS_PRE_D:SOURCE_ID and S_M_I_SALES_ORDLNS_PRE_U:SOURCE_ID session values.

3 Save and close the file.

To configure the Sales Order Lines aggregate table for SAP R/3

1 Open the `file_parameters_sapr3.csv` file using Microsoft WordPad or Notepad in the `$pmserver\srcfiles` folder.

2 Replace the default PARM_SVALUE2 values with your new values.

For a list of values for each parameter see the About Configuring the Sales Order Lines Aggregate Table on page 273.

The default values are shown in the following table:

PARM_TYPE	PARM_CODE	PARM_SVALUE_1	PARM_SVALUE_2	PARM_SVALUE_3
SAPR3	S_M_S_SALES_ORDLNS_PRE_D:GRAIN	S	MONTH	MPLT_ADI_SALES_ORDLNS_PRE_POST
SAPR3	S_M_S_SALES_ORDLNS_PRE_D:SOURCE_ID	S	'SAPR3'	—
SAPR3	S_M_S_SALES_ORDLNS_PRE_U:GRAIN	S	MONTH	MPLT_ADI_SALES_ORDLNS_PRE_POST
SAPR3	S_M_S_SALES_ORDLNS_PRE_U:SOURCE_ID	S	‘SAPR3'	—

NOTE: You need to use single quotes for the S_M_S_SALES_ORDLNS_PRE_D:SOURCE_ID and S_M_S_SALES_ORDLNS_PRE_U:SOURCE_ID session values.

3. Save and close the file.

To configure the Sales Order Lines aggregate table for Universal Source

1. Open the `file_parameters_univ.csv` file using Microsoft WordPad or Notepad in the `$pmserver\srcfiles` folder.

2. Replace the default PARM_SVALUE2 values with your new values.

For a list of values for each parameter see the About Configuring the Sales Order Lines Aggregate Table on page 273.

The default values are shown in the following table:

PARM_TYPE	PARM_CODE	PARM_SVALUE_1	PARM_SVALUE_2	PARM_SVALUE_3
UNIV	S_M_F_SALES_ORDLNS_PRE_U:GRAIN	S	MONTH	MPLT_ADI_SALES_ORDLNS_PRE_POST
UNIV	S_M_F_SALES_ORDLNS_PRE_U:SOURCE_ID	S	'GENERIC'	—

NOTE: You need to use single quotes for the S_M_F_SALES_ORDLNS_PRE_U:SOURCE_ID session value.

3 Save and close the file.

About Tracking Multiple Products for Siebel Enterprise Sales Analytics

The Sales Order Lines table contains two columns, `ORDHD_KEY_ID` and `ORDLN_KEY_ID`, that track individual products when they are grouped and sold as a single package. These two columns allow you to analyze the relationship of all products sold as a single unit. The `ORDHD_KEY_ID` column stores the Order ID of the entire sales order. The `ORDLN_KEY_ID` column stores the Line Item ID of the parent product.

For example, assume a customer purchases a package that includes a computer, scanner, and printer. In addition, the customer purchases a monitor separately. In this case, there are two parent items: the package and the monitor. The computer, scanner, and printer are all child orders of the parent order *package*, while the parent order *monitor* is a single-item purchase.

Your data warehouse may store this sales information in the Sales Order Lines table as seen in Table 43. The ORDLN_KEY_ID field contains the Line Item ID of the parent product in order to maintain the relationship between the parent and child products in a package. In this example, the ORDLN_KEY_ID field is Line_1 for each of the three child products (A1, A2, A3) that were sold as a part of the parent package, Parent A.

Table 43. Sales Order Table Columns With Parent/Child Relationships

Key_ID	SALES_ORDER _NUM	PRODUCT_ ID	ORDHD_ KEY_ID	ORDLN_ KEY _ID	Relationship (Not a column in the table.)
Line_1	1000	Package	1000	Line_1	Parent A
Line_2	1000	Computer	1000	Line_1	Child A1
Line_3	1000	Scanner	1000	Line_1	Child A2
Line_4	1000	Printer	1000	Line_1	Child A3
Line_5	1000	Monitor	1000	Line_5	Parent B (no children)

In contrast, if each of the four items described in Table 43 were bought individually, the ORDLN_KEY_ID would have a different Line Item ID for every row. In this case, the Sales Order Lines table would look like Table 44.

Table 44. Sales Order Table Columns Without Parent/Child Relationships

Key_ID	SALES_ORDER_ NUM	PRODUCT_ ID	ORDHD_ KEY_ID	ORDLN_ KEY _ID	Relationship (Not a column in the table.)
Line_1	1000	Computer	1000	Line_1	None
Line_2	1000	Scanner	1000	Line_2	None
Line_3	1000	Printer	1000	Line_3	None
Line_4	1000	Monitor	1000	Line_4	None

Adding Dates to the Order Cycle Time Table for Post-Load Processing

It is recommended that you do not change the structure of any of the data warehouse tables, with the exception of the Order Cycle Time table—IA_SALES_CYCLNS. In particular, the Order Cycle Time table may not store every sales-related date. As a result, you want to create a new date field in this table to accommodate your additional dates.

To add more dates, you need to understand how the Order Cycle Times table is populated. Thus, if you want to change the dates loaded into the Order Cycle Time table (IA_SALES_CYCLNS), then you have to modify the M_PLP_SALES_CYCHDR_LOAD mappings that take the dates from the IA_* tables and load them into the Cycle Time table.

NOTE: Be sure that the date is already being extracted and stored in IA and OD data warehouse tables.

To add dates to the Cycle Time table load

1. In the PowerCenter Designer, open the Configuration for Post Load Processing folder.
2. In Warehouse Designer, modify the table definition for the target table to verify that it has a field to store this date.

   For example, if you are loading the Validated on Date in the IA_SALES_CYCLNS table, then you need to create a new column, VALIDATED_ON_DT, and modify the target definition of the IA_SALES_CYCLNS table.
3. In Source Analyzer, modify the table definition of the source table to include this new column.

   Continuing with the example, you would include the VALIDATED_ON_DT column in the IA_SALES_CYCLNS source table.
4. Create the table in the database with the new table structure.

   TIP: If you have already loaded data in the IA_SALES_CYCLNS table, then make sure that you backup the data before you recreate this table.
5. Modify the M_PLP_SALES_CYCLNS_INCR_LOAD mapping to select the new column from any of the following source tables, and load it to the IA_SALES_CYCLNS target table:

   - IA_SALES_ORDLNS
   - IA_SALES_IVCLNS
   - IA_SALES_PCKLNS
   - IA_SALES_SCHLNS
6. Modify The Source Qualifier SQL Override for the mapping, and map the column in the Transformation to map it to the target table.

About Configuring the Backlog Period Date for Siebel Enterprise Sales Analytics

The Backlog table (IA_BLGLNS) stores backlog data for the current month. In contrast, the Backlog History table (IA_BLGHIS) stores snapshots of all previous months' historical backlog data. The periods for which the Backlog History table tracks backlog data is defined by the Backlog Period Date. By default, the date is set as the last calendar day of the month; however you may configure this date. You may want to view backlog history at a more detailed level, such as by day or by week, instead of by month. The following example describes how historical backlog data is stored and what the implications are for changing the backlog time period.

Example of How Backlog Data Is Stored in the Backlog History Table

Assume you represent a manufacturing company where financial backlog is defined as any item that is ordered, but not invoiced. On February 1, 2001, you received an order (Sales Order #1) for 30 products. 20 were shipped and invoiced and 10 were shipped, but not invoiced. At the end of the day, there is an entry in the Backlog table and in the Backlog History table. The entry in the Backlog History table looks like that shown in Table 45.

Table 45. Oracle 11i: Backlog History Table Entry as of February 1, 2001

SALES_ORDER_NUM (Sales Order Number)	BACKLOG _DK (Backlog Date)	BACKLOG_PERIOD_DK (Backlog Period Date)	OPEN_QTY (Backlog Quantity)
1	02/01/2001	02/28/2001	10

On February 2, 5 of the 10 financial backlog items are invoiced and, thus, removed from the backlog. Thus, there is an update to the existing row in the Backlog History table, as shown in Table 46.

Table 46. Oracle 11i: Backlog History Table Entry as of February 2, 2001

SALES_ORDER_NUM (Sales Order Number)	BACKLOG _DK (Backlog Date)	BACKLOG_PERIOD_DK (Backlog Period Date)	OPEN_QTY (Backlog Quantity)
1	~~02/01/2001~~ 02/01/2001	02/28/2001	~~10~~ 5

No further activity happens until February 28. On February 28, the remaining 5 items on financial backlog are invoiced and removed from financial backlog. In addition, a new sales order (Sales Order #2) comes in for 50 new items. All of the items are put on financial backlog.

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

For the 50 new financial backlog items, there is a new entry into the Backlog History table. So, as of February 28, 2001, the Backlog History table looks like the Table 47.

Table 47. Oracle 11i: Backlog History Table Entry as of February 28, 2001

SALES_ORDER_NUM (Sales Order Number)	BACKLOG _DK (Backlog Date)	BACKLOG_PERIOD_DK (Backlog Period Date)	OPEN_QTY (Backlog Quantity)
1	~~02/01/2001~~ 02/02/2001	02/28/2001	~~10~~ 5
2	02/28/2001	02/28/2001	50

On March 1, 30 more items are ordered (Sales Order #3), all of which are on financial backlog. The resulting Backlog History table looks like Table 48.

Table 48. Oracle 11i: Backlog History Table Entry as of March 1, 2001

SALES_ORDER_NUM (Sales Order Number)	BACKLOG _DK (Backlog Date)	BACKLOG_PERIOD_DK (Backlog Period Date)	OPEN_QTY (Backlog Quantity)
1	~~02/01/2001~~ 02/02/2001	02/28/2001	5
2	02/28/2001	02/28/2001	50
2	03/01/2001	03/31/2001	50
3	03/01/2001	03/31/2001	30

Because backlog history is maintained at the monthly level, you have a partial history of your backlogs. Based on the latest state of the Backlog History table shown in Table 48, you can see that sales order number 1 and 2 ended up with 5 and 50 financial backlogged items respectively. You do not have visibility into what the initial financial backlogged item quantities were for both of these sales orders; you only have their ending quantities.

If you decide that you want to track more details on how the items moved out of backlog, then you'll have to maintain the history at a more granular level. For instance, if you want to know the number of items that were on backlog when the it was first opened, you would have to track the backlog history by day, instead of by month.

For example, if you maintained backlog history at the daily level you would be able to capture that sales order 1 had an initial backlog of 10 as of February 1 and the backlog quantity shrank to 5 as of February 2. So, by capturing history at the daily level, you could then compute cycle times on how long it took to move items out of backlog. However, if you decide to capture backlog history at a more detailed level, you may compromise performance because tracking backlog history at the daily level can increase the size of the Backlog History table exponentially.

If you choose to change the time period for which historical backlog data is kept, you must verify that all types of backlog are being stored at the same grain; which requires modification to multiple mappings. Table 49 provides a list of all applicable mappings and their corresponding Expression transformations that you must modify.

Table 49. Oracle 11i: Backlog History Applicable Mappings and Expression Transformations

Mapping	Expression Transformation
M_I_SALES_BLGLNS_LOAD	EXP_SALES_BLGLNS

The backlog history period is monthly by default. The default SQL statement in the Expression transformation of the listed mappings is as follows:

```
trunc(DATE_DIFF(LAST_DAY(CAL_DAY_DT),to_date('01-JAN-1900','DD-MON-
YYYY'),'DD')) + 2415021
```

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

Configuring the Backlog Period Date for Siebel Enterprise Sales Analytics

By default, the date is set as the last calendar day of the month; however you may configure this date. You may want to view backlog history at a more detailed level, such as by day or by week, instead of by month.

To configure the backlog period date

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. Open the applicable Sales Backlog Lines mapping.
3. Double-click the Expression transformation to display the BACKLOG_PERIOD_DK port.

   The SQL statement in this port's expression contains the backlog period date.
4. In the Ports tab, modify the default SQL statement for the BACKLOG_PERIOD_DK.

   For example, if you want to store backlog history at the weekly level, replace the existing statement:

```
trunc(DATE_DIFF(LAST_DAY(CAL_DAY_DT),to_date('01-JAN-1900','DD-MON-YYYY'),'DD')) +
2415021
```

with this new statement:

```
trunc(DATE_DIFF(CAL_WEEK_END_DT),to_date('01-JAN-1900','DD-MON-YYYY'),'DD')) +
2415021
```

```
TO_CHAR(CAL_WEEK_END_DT,’J’)
```

NOTE: The CAL_WEEK_END_DT is not prepackaged to be extracted from IA_DATES; to use this calculation you have to extract this data from IA_DATES and pass it to the Expression transformation.

5 Validate and save your changes to the repository.

About the Grain at Which Currency Amounts and Quantities Are Stored

Quantities and currency amounts may be stored at different grains in the Order Cycle Time (IA_SALES_CYCLNS) table, because of their unique ability to be rolled up.

Consider as an example a situation where a customer orders one package, which includes a computer, scanner, printer, and two speakers. In addition to the package, the customer also orders one monitor, which is not included in the package deal. In this case, the sales quantities are listed for the parent line item as well as for each child line items. However, the currency amounts are only listed for the parent line items; they are not listed for the individual child line items. Table 50 illustrates this example.

Table 50. Storing Currency Amounts at the Parent Line Level in Order Cycle Time table

Key_ID	SALES_ ORDER_ NUM	PRODUCT_ ID	ORDHD_ KEY _ID	ORDLN_ KEY _ID	SALES _QTY	Currency Amount	Relation
Line_1	1000	Package	1000	Line_1	1	$1000	Parent A
Line_2	1000	Computer	1000	Line_1	1	-	Child A1
Line_3	1000	Scanner	1000	Line_1	1	-	Child A2
Line_4	1000	Printer	1000	Line_1	1	-	Child A3
Line_5	1000	Speaker	1000	Line_1	2	-	Child A4
Line_6	1000	Monitor	1000	Line_5	1	$400	Parent B (no children)

Consider another example. In this example, a customer orders the same package, which includes a computer, scanner, printer, and two speakers. In addition to the package, the customer also orders one monitor, which is not included in the package deal. In this case, the quantities are provided for the parent and child line item levels. In addition, the currency amounts are also listed for both the parent and child line item levels. Table 51 illustrates this example.

Table 51. Storing Currency Amounts at the Child Line Level in Order Cycle Time table

Key ID	Order #	Product	ORDHD_ KEY_ID	ORDLN_ KEY_ID	Relationship	Qty	Currency Amount
Line_1	1000	Package	1000	Line_1	Parent A	1	-
Line_2	1000	Computer	1000	Line_1	Child A1	1	725
Line_3	1000	Scanner	1000	Line_1	Child A2	1	75
Line_4	1000	Printer	1000	Line_1	Child A3	1	150
Line_5	1000	Speaker	1000	Line_1	Child A4	2	50
Line_6	1000	Monitor	1000	Line_5	Parent B (no children)	1	$400

For more information on parent and child relationships, see About Tracking Multiple Products for Siebel Enterprise Sales Analytics on page 278.

About the Sales Order Dates

The Order Cycle Time table (IA_SALES_CYCLNS) store many sales-related dates, which are sourced from the IA_SALES_ORDLNS, IA_SALES_SCHLNS, IA_SALES_PCKLNS, OD_SALES_ORDHLD, and IA_SALES_IVCLNS tables. Table 52 lists all the Sales Order Date columns.

Table 52. Sales Order Date Columns

Date Columns
CREATED_ON_DK
ORDERED_ON_DK
BOOKED_ON_DK
ACT_PICK_ON_DK
ACT_SHIP_ON_DK
INVOICED_ON_DK
CLOSED_ON_DK
PURCH_ORDER_DT
ORDERED_ON_DT

Table 52. Sales Order Date Columns

Date Columns
REQUIRED_BY_DT
CUST_REQ_SHIP_DT
CREATED_ON_DT
PROMISED_ON_DT
BOOKED_ON_DT
ADDNL_BOOKED_ON_DT
CANCELLED_ON_DT
CSD_FIRST_PICK_DT
CSD_LAST_PICK_DT
CSD_FIRST_PACK_DT
CSD_LAST_PACK_DT
CSD_FIRST_LOAD_DT
CSD_LAST_LOAD_DT
CSD_FIRST_SHIP_DT
CSD_LAST_SHIP_DT
CSD_FIRST_DELV_DT
CSD_LAST_DELV_DT
ACT_FIRST_PICK_DT
ACT_LAST_PICK_DT
ACT_FIRST_PACK_DT
ACT_LAST_PACK_DT
ACT_FIRST_LOAD_DT
ACT_LAST_LOAD_DT
ACT_LAST_SHIP_DT
ACT_FIRST_DELV_DT
ACT_LAST_DELV_DT
FIRST_INVOICE_DT
LAST_INVOICE_DT

Domain Values and CSV Worksheet Files for Siebel Enterprise Sales Analytics

Table 53 lists the CSV worksheet files and the domain values for Siebel Enterprise Sales Analytics in the `$pmserver\LkpFiles` folder.

For more information on configuring domain values with CSV worksheet files, see About Domain Values on page 154 and Configuring the Domain Value Set with CSV Worksheet Files on page 159.

Table 53. Domain Values and CSV Worksheet Files for Siebel Enterprise Sales Analytics

Worksheet File Name	Description	Session
domainValues_InvoiceTypes_ora11i.csv	Lists the Invoice Document Type column and the corresponding domain values for the Oracle 11i application.	S_M_I_XACT_TYPES_SALES_IVCLNS_LOAD
domainValues_InvoiceTypes_sapr3.csv	Lists the Invoice Document Type Code and the Invoice Document Type Desc columns, and the corresponding domain values for the SAP R/3 application.	S_M_S_XACT_TYPES_SALES_IVCLNS_LOAD
domainValues_PickTypes_ora11i.csv	Lists the Picking Document Type column and the corresponding domain values for the Oracle 11i application.	S_M_I_XACT_TYPES_SALES_PCKLNS_LOAD
domainValues_PickTypes_sapr3.csv	Lists the Picking Document Type Code and the Picking Document Type Desc columns, and the corresponding domain values for the SAP R/3 application.	S_M_S_XACT_TYPES_SALES_PCKLNS_LOAD
domainValues_OrderTypes_ora11i.csv	Lists the Order Document Type column and the corresponding domain values for the Oracle 11i application.	S_M_I_XACT_TYPES_SALES_ORDLNS_LOAD
domainValues_OrderTypes_sapr3.csv	Lists the Order Document Type Code and the Order Document Type Desc columns, and the corresponding domain values for the SAP R/3 application.	S_M_S_XACT_TYPES_SALES_ORDLNS_LOAD
domainValues_PickStatus_ora11i.csv	Lists the Picking Status Code and the Status Desc columns, and the corresponding domain values for the Oracle 11i application.	S_M_I_STATUS_SALES_PCKLNS_LOAD
domainValues_PickStatus_sapr3.csv	Lists the Picking Status Code and the Status Desc columns, and the corresponding domain values for the SAP R/3 application.	S_M_S_STATUS_SALES_PCKLNS_LOAD, S_M_S_STATUS_SALES_GOODS_MVMT_LOAD

Table 53. Domain Values and CSV Worksheet Files for Siebel Enterprise Sales Analytics

Worksheet File Name	Description	Session
domainValues_InvoiceStatus_ora11i.csv	Lists the Invoice Status Code and the Status Desc columns, and the corresponding domain values for the Oracle 11i application.	S_M_I_STATUS_SALES_IVCLNS_LOAD
domainValues_InvoiceStatus_sapr3.csv	Lists the Invoice Status Code and the Status Desc columns, and the corresponding domain values for the SAP R/3 application.	S_M_S_STATUS_SALES_IVCLNS_LOAD
domainValues_OrderOverallStatus_ora11i.csv	Lists the Order Status Code and the Status Desc columns, and the corresponding domain values for the Oracle 11i application.	S_M_I_STATUS_SALES_ORDLNS_CYCLES_LOAD
domainValues_OrderOverallStatus_sapr3.csv	Lists the Order Status Code and the Status Desc columns, and the corresponding Domain Value for the SAP R/3 application.	S_M_S_STATUS_SALES_OVERALL_LOAD
domainValues_RejectionStatus_sapr3.csv	Lists the Order Rejection Status Code and the Status Desc columns, and the corresponding domain values for the SAP R/3 application.	S_M_S_STATUS_SALES_REJECTIONS_LOAD
domainValues_PayMethodCode_ora11i.csv	Lists the Order Payment Method Code and the Method Desc columns, and the corresponding domain values for the Oracle 11i application.	S_M_I_PAY_TYPES_LOAD
domainValues_PayMethodCode_sapr3.csv	Lists the method code column and the corresponding domain value for the SAP R/3 application.	S_M_S_METHODS_LOAD

Configuring Siebel Supply Chain Analytics for Siebel Enterprise Sales Analytics

The Siebel Enterprise Sales Analytics application uses tables that are also used in the Siebel Supply Chain Analytics application.

For Oracle 11i you need to use the following configuration steps for Siebel Supply Chain Analytics to configure Siebel Enterprise Sales Analytics:

- Configuring Quantity Types for Product Transactions on page 384
- Configuring the Region Name on page 385
- Configuring the State Name on page 386

- Configuring the Country Name on page 386
- Configuring the Make-Buy Indicator on page 387

For post-load processing for Oracle 11i and SAP R/3, you need to use the following configuration steps for Siebel Supply Chain Analytics to configure Siebel Enterprise Sales Analytics:

- Configuring the Inventory Balance Aggregate Table on page 389
- Configuring the Product Transaction Aggregate Table on page 391

For information on configuring Siebel Supply Chain Analytics for SAP R/3 to configure Siebel Enterprise Sales Analytics, see About the SAP R/3 Inventory Transfer Process for Siebel Supply Chain Analytics on page 393.

Configuring Siebel Financial Analytics for Siebel Enterprise Sales Analytics

The Siebel Enterprise Sales Analytics application uses tables that are also used in the Siebel Financial Analytics application.

For Oracle 11i, you need to use the following configuration steps for Siebel Financial Analytics to configure Siebel Enterprise Sales Analytics:

- Extracting Data Posted at the Detail-Level for Oracle 11i on page 311
- Mapping Siebel General Ledger Analytics Account Numbers to Group Account Numbers on page 312
- Filtering Extracts Based on Set of Books ID for Siebel General Ledger Analytics on page 313
- Configuring Siebel General Ledger Analytics Transaction Extracts on page 314
- Configuring the General Ledger Balance ID on page 319
- Configuring AR Balance ID for Siebel Receivables Analytics and Siebel Profitability Analytics on page 320
- Configuring the AR Adjustments Extract for Siebel Receivables Analytics on page 321
- Configuring the AR Schedules Extract on page 322
- Configuring the AR Cash Receipt Application Extract for Siebel Receivables Analytics on page 322
- Configuring the AR Credit-Memo Application Extract for Siebel Receivables Analytics on page 323

For SAP R/3, you need to use the following configuration steps for Siebel Financial Analytics to configure Siebel Enterprise Sales Analytics:

- Extracting Data Posted at the Header Level for SAP R/3 on page 329
- Configuring the Group Account Number Categorization for Siebel General Ledger Analytics on page 330
- Configuring the Transaction Types for Siebel Financial Analytics on page 331
- Configuring the Siebel General Ledger Analytics Balance Extract on page 335
- Configuring the Siebel Receivables Analytics Balance Extract on page 338

For post-load processing, you need to use the following configuration steps for Siebel Financial Analytics to configure Siebel Enterprise Sales Analytics:

- Configuring Aging Buckets for Siebel Receivables Analytics on page 345
- Configuring the History Period for the Invoice Level for Siebel Receivables Analytics on page 346

15 Configuring Siebel Enterprise Workforce Analytics

This chapter describes how to configure certain objects for particular sources to meet your business needs. It contains the following topics:

Overview of Siebel Enterprise Workforce Analytics

Siebel Enterprise Workforce Analytics contains information for workforce operations, employee events, and payroll.

The Siebel Enterprise Workforce Analytics application has the following functional areas:

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

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

- **Human Resource Performance.** The information stored in the Human Resource Performance area allows you to measure several areas of performance, including contribution and productivity, workforce effectiveness, and trends analytics.

- **Retention.** Under the Retention functional area you can find the events that are the hallmarks of employees' professional life cycle. These events include their hiring information, their promotional opportunities realized and not realized, the quality of the employees' job performance as measured by performance ranking, their length of service, and the reasons for termination, both voluntary and involuntary. Monitoring retention rates within departments is useful in determining potential problem areas that may want to be addressed by senior management.
- **U.S. Statutory Compliance.** The U.S. Statutory Compliance functional area stores information that help Human Resources departments prepare government-required reports.
- **Workforce Profile.** The Workforce Profile functional area provides you with the tools to separate sensitive from nonsensitive information, and to restrict access to sensitive data. Sensitive information includes such data as ethnicity, age, native language, marital status, and performance ratings. Nonsensitive information includes information such as job title, work location, and position status.

About Configuring Workforce Operations for Oracle 11i

There are two dimensions in the U.S. Statutory Compliance configuration for the Oracle 11i folder for which there are mandatory changes to the configuration information—the Employees dimension and the Jobs dimension. Failure to configure the following mandatory configuration information can result in incorrect data being fed to reports and other ETL processes.

NOTE: If you want to retain values from the source system or previously existing values that are not included in the domain values, enter an Else statement in the expression for the code.

This section also provides the necessary information on configuring the Workforce Profile functional area for Oracle 11i—configuring address types, configuring phone types, modifying the derive flag, and modifying the snapshot extract date.

NOTE: Currently, no configuration changes are required for Oracle 11i for the Retention functional area.

Process of Configuring Workforce Operations for Oracle 11i

This section contains Workforce Operations configuration information that is specific to Oracle 11i.

To configure Workforce Operations for Oracle 11i, perform the following tasks:

- Configuring the Employees Dimension for U.S. Statutory Compliance on page 293
- Configuring the Jobs Dimension for U.S. Statutory Compliance on page 296
- Enhancing Existing Mappings for U.S. Statutory Compliance on page 298
- Configuring Address Types for Workforce Profile on page 299
- Configuring Phone Types for Workforce Profile on page 299

- Modifying the Derive Flag for Workforce Profile on page 300
- Modifying the Snapshot Extract Date for Workforce Profile on page 301

Configuring the Employees Dimension for U.S. Statutory Compliance

This task is a step in the Process of Configuring Workforce Operations for Oracle 11i on page 292.

Within the Employees dimension there are mandatory changes to the configuration information in the Expression transformation EXP_EMPLOYEES for the mapping M_I_EMPLOYEES_EXTRACT.

The configuration information includes the domain values for the following:

- Ethnic Group Code (ETHN_GRP_CODE_OUT)
- Ethnic Group Description (ETHN_GRP_DESC_OUT)
- Veteran Status Code (VETERAN_STAT_CODE_OUT)
- Veteran Status Description (VETERAN_STAT_DESC_OUT)

Configuring Ethnic Group Codes

The domain values for Ethnic Group Code and Ethnic Group Description in Siebel Customer-Centric Enterprise Warehouse are shown in Table 54.

Table 54. Domain Values for Ethnic Group Code and Ethnic Group Description

Ethnic Group Code	Ethnic Group Description
1	White
2	Black
3	Asian
4	American Indian/Alaskan Native
5	Native Hawaiian or Other Pacific Islander
6	Hispanic or Latino (White)
7	Hispanic or Latino (All Other Races)
8	Race Unknown
9	Others

For each of these ports' expressions, it is necessary that the source-supplied values are mapped to the expected domain values so that correct ethnic group and veteran status information is supplied. For example, if the source-supplied values were as shown in Table 55, there would be a discrepancy between the domain values in Siebel Customer-Centric Enterprise Warehouse and source-supplied value for Ethnic Group Code 1, 2, and 4.

Table 55. Sample Source Values for Ethnic Group Code and Ethnic Group Description

Ethnic Group Code	Ethnic Group Description
1	Caucasian
2	African American
4	Asian

To modify the Ethnic Group code

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In the M_I_EMPLOYEES_EXTRACT mapping, double-click the EXP_EMPLOYEES Expression transformation.
3. Modify the ETHN_GRP_CODE_OUT port's expression by mapping the source-supplied values to the domain values.
4. Validate and save your changes to the repository.

Configuring Ethnic Group Descriptions

For the Ethnic Group Description, make sure that the description used with the source-specified code matches the description defined in the domain values. For example, if the source system uses the description, African American, instead of Black for code 2, the description must be modified to reflect the descriptive value of Black.

To modify the Ethnic Group description

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In the M_I_EMPLOYEES_EXTRACT mapping, double-click the EXP_EMPLOYEES Expression transformation.
3. Modify the ETHN_GRP_DESC_OUT port's expression by mapping the source-supplied values to the domain values.
4. Validate and save your changes to the repository.

Configuring Veteran Status Codes

The steps to resolve the Veteran Status Code and the Veteran Status Description requirements are very similar to those for Ethnic Group. The domain values for Veteran Status Code and Veteran Status Description are shown in Table 56.

Table 56. Domain Value for Veteran Status Code and Veteran Status Description

Veteran Status Code	Veteran Status Description
1	Special Disabled Veterans
2	Vietnam Era Veterans
3	Other Protected Veterans
4	Newly Separated Veterans

For each of these ports' expressions, it is necessary that the source-supplied values are mapped to the expected domain values so that the correct veteran status information is supplied. For example, if the source-supplied values for Veteran Status Code and Veteran Status Description were as shown in Table 57, there would be a discrepancy between the domain values and source-supplied values for all codes.

Table 57. Sample Source-Supplied Veteran Status Code and Veteran Status Description

Veteran Status Code	Veteran Status Description
VET	Other Veterans
VIETVET	Vietnam Veterans
VIETVETDIS	Disabled Vietnam Veteran

To modify Veteran Status code

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In the M_I_EMPLOYEES_EXTRACT mapping, double-click the EXP_EMPLOYEES Expression transformation.
3. Modify the VETERAN_STAT_CODE_OUT port's expression by mapping the source-supplied values to the domain values.
4. Validate and save your changes to the repository.

Configuring Veteran Status Descriptions

As with the Ethnic Group Description, make sure that the Veteran Status Description used with the source-specified code matches the description defined in the domain values. For example, in the sample source data in Table 57, the source system uses the description Other Veterans instead of Other Protected Veterans for source-supplied code VET.

To modify Veteran Status description

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In the M_I_EMPLOYEES_EXTRACT mapping, double-click the EXP_EMPLOYEES Expression transformation.
3. Modify the VETERAN_STAT_DESC_OUT port's expression by mapping the source-supplied values to the domain values.
4. Validate and save your changes to the repository.

Configuring the Jobs Dimension for U.S. Statutory Compliance

This task is a step in the Process of Configuring Workforce Operations for Oracle 11i on page 292.

Within the Jobs dimension there are two mandatory changes to the configuration information in the Expression transformation EXP_JOBS for the mapping M_I_JOBS_EXTRACT. The configuration information includes the domain values for the following:

- EEO Job Category Code (EEO_JOB_CAT_CODE_OUT)
- EEO Job Category Description (EEO_JOB_CAT_DESC_OUT)

Configuring EEO Job Category Codes

The domain values for EEO Job Category Code and EEO Job Category Description in Siebel Customer-Centric Enterprise Warehouse are shown in Table 58.

Table 58. Domain Values for EEO Job Category Code and EEO Job Category Description

EEO Job Category Code	EEO Job Category Description
1	Officials and Managers
2	Professionals
3	Technicians
4	Sales Workers
5	Office and Clerical
6	Craft Workers
7	Operatives
8	Laborers
9	Service Workers

For each of these ports' expressions, it is necessary that the source-supplied values are mapped to the expected domain values so that the correct EEO job category information is supplied. For example, if the source-supplied values were as shown in Table 59 there would be a discrepancy between the domain values and source-supplied values for EEO Job Categories 1, 2, 5, and 8.

Table 59. Sample Source-Supplied EEO Job Category Code and EEO Job Category Description

EEO Job Category Code	EEO Job Category Description
1:EEO1CODE	Laborers (Unskilled)
2:EEO1CODE	Office-Clerical
3:EEO1CODE	Technicians
4:EEO1CODE	Sales Workers
5:EEO1CODE	Professionals
6:EEO1CODE	Craft Workers
7:EEO1CODE	Operatives (Semi-Skilled)
8:EEO1CODE	Officials and Managers
9:EEO1CODE	Service Workers

To modify EEO Job Category code

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In the M_I_JOBS_EXTRACT mapping, double-click the EXP_JOBS Expression transformation.
3. Modify the EEO_JOB_CAT_CODE_OUT port's expression by mapping the source-supplied values to the domain values.
4. Validate and save your changes to the repository.

Configuring EEO Job Category Descriptions

As with the Ethnic Group Description and Veteran Status Description, make sure that the EEO Job Category Description used with the source-specified code matches the description defined in the domain values. For example, in the sample source data in Table 59, the source system uses the description Office-Clerical instead of Office and Clerical for source-supplied Code 2.

To modify the EEO Job Category description

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In the M_I_JOBS_EXTRACT mapping, double-click the EXP_JOBS Expression transformation.
3. Modify the EEO_JOB_CAT_DESC_OUT port's expression by mapping the source-supplied values to the domain values.
4. Validate and save your changes to the repository.

Enhancing Existing Mappings for U.S. Statutory Compliance

This task is a step in the Process of Configuring Workforce Operations for Oracle 11i on page 292.

The final task before running the upgrade mappings involves:

1. Adding the EEO_JOB_CAT_DESC port to the corresponding staging table, TI_JOBS. (TI_EMPLOYEES already has the ETHN_GRP_DESC port, therefore no enhancement is required.)
2. Enhancing the Source Qualifier for the existing load mappings, M_I_EMPLOYEES_LOAD and M_I_JOBS_LOAD, to accommodate the added ETHN_GRP_DESC and EEO_JOB_CAT_DESC columns.

First, modify the staging table, TI_JOBS, to add port EEO_JOB_CAT_DESC. You must modify the staging tables in both the Target and Source folder, as well as in the back-end database.

To modify the TI_JOBS staging table

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In the Source folder, double-click the TI_JOBS staging table located in the IA_ORA_STAGE sub-folder to open the Edit Tables window.
3. Select Replace after dragging and dropping the table into the Warehouse Designer.
4. In the Columns tab, add a new port directly below EEO_JOB_CAT_CODE port.
5. In the new field, enter the port name of EEO_JOB_CAT_DESC, and select a data type of String (254).
6. Click OK and save changes to the repository.

   NOTE: Open the Target folder, and repeat the preceding procedure. Please note that you must modify the corresponding staging table in the back-end database to accommodate the new port.

The next task is to modify the Source Qualifier for the load mappings M_I_EMPLOYEES_LOAD and M_I_JOBS_LOAD.

To modify the Source Qualifier in the load mapping for M_I_EMPLOYEES_LOAD

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In the M_I_EMPLOYEES_LOAD mapping, double-click the SQ_EMPLOYEES Source Qualifier.
3. In the Ports tab, add a new port directly below the ETHN_GRP_CODE port.
4. In the new field, enter the port name of ETHN_GRP_DESC, and select a data type of String (254).
5. Click OK.
6. In the Properties tab, open SQL Query field. There are two options:
   - If you have not modified the SQL statement, you can select Generate SQL and click OK.
   - If you are not using the preconfigured SQL statement, you must modify the join condition manually.

7. Validate and save changes to the repository.

To modify the Source Qualifier in the load mapping for M_I_JOBS_LOAD

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In the M_I_JOBS_LOAD mapping, double-click the SQ_JOBS Source Qualifier.
3. In the Ports tab, add a new port directly below the EEO_JOB_CAT_CODE port.
4. In the new field, enter the port name of EEO_JOB_CAT_DESC, select a data type of String (254), and click OK.
5. In the Properties tab, open SQL Query field. There are two options:
   - If you have not modified the SQL statement, you can select Generate SQL and click OK.
   - If you are not using the preconfigured SQL statement, you must modify the join condition manually.
6. Validate and save changes to the repository.

After you have modified the domain values in the upgrade mappings, and enhanced the column structure in the existing mappings, run the upgrade mappings prior to proceeding with configuring the extract mappings. For a discussion on configuring the Employees Dimension and configuring the Jobs Dimension, see About Configuring Workforce Operations for Oracle 11i on page 292. Make sure to run the mapping only once to properly reflect data.

Configuring Address Types for Workforce Profile

This task is a step in the Process of Configuring Workforce Operations for Oracle 11i on page 292.

By default the address type for employee information is type M for mailing type. To modify the address type you must configure the MPLT_BCO_EMPLOYEES Business Component.

To configure Address Type

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In Mapplet Designer, open the MPLT_BCI_EMPLOYEES mapplet.
3. In the Properties tab, modify the SQL query to accommodate the new address type.

   For example, the address type has been modified to H for home address instead of the default M for mailing.

4. Validate and save changes to the repository.

Configuring Phone Types for Workforce Profile

This task is a step in the Process of Configuring Workforce Operations for Oracle 11i on page 292.

By default the phone type for employee information is P. To modify the phone type you must configure the MPLT_BCI_EMPLOYEES Business Component.

To configure Phone Type

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In Mapplet Designer, open the MPLT_BCI_EMPLOYEES mapplet.
3. In the Properties tab, modify the SQL statement to accommodate the new phone type.

   For example, the phone type has been modified to F for Home Fax phone type instead of the default P.
4. Validate and save changes to the repository.

Modifying the Derive Flag for Workforce Profile

This task is a step in the Process of Configuring Workforce Operations for Oracle 11i on page 292.

The $$DERIVE_FLAG setting determines where the specific history table is populated. By default, the $$DERIVE_FLAG is set to YES. However, if you want to change this flag, you can do so by modifying the `file_parameters_ora11i.csv` file in the installation directory. This file contains a default value for the sessions shown in Table 60.

Table 60. Derive Flag—Default Value for the Sessions

Session	Default Value
S_M_I_EMP_HISTORY_A1_DERIVE	YES
S_M_I_EMP_HISTORY_A2_DERIVE	YES

After you have modified the file, the next time you run the session, the new Derive Flag value is populated.

To modify the Default Derive Flag

1. Open the `file_parameters_ora11i.csv` file in the installation directory, in `$pmserver\srcfiles` folder.
2. Replace the default Derive Flag with the new Derive Flag.

   Perform this action for all applicable sessions, as shown in the following figure.

S_M_I_EMP_HISTORY_A1_DERIVE:DERIVE_FLAG	0	0	0	0	S	NO
S_M_I_EMP_HISTORY_A2_DERIVE:DERIVE_FLAG	0	0	0	0	S	YES

3. Save and close the file.

Modifying the Snapshot Extract Date for Workforce Profile

This task is a step in the Process of Configuring Workforce Operations for Oracle 11i on page 292.

By default, the $$EXTRACT_DATE is set to 01/01/1970 for Oracle 11i. However, if you want to modify this value, you can do so by modifying the `file_parameters.ora11i.csv` file in the installation directory. This file contains this default value for the sessions in Table 61.

Table 61. Sessions with 01/01/1970 as the Default Snapshot Extract Date

Session
S_M_I_EMP_SNAPSHOT_1_EXTRACT_P1
S_M_I_EMP_SNAPSHOT_2_EXTRACT_P2
S_M_I_EMP_SNAPSHOT_3_EXTRACT_P3
S_M_I_EMP_SNAPSHOT_4_EXTRACT_P4

After you have modified the file, the next time you run the session, the new snapshot extract date is populated.

To redefine the Snapshot Extract date

1. Open the `file_parameters_ora11i.csv` file in the installation directory, in `$pmserver\srcfiles`.
2. In the PARAM_DVALUE_1 column, enter the date for which you want the data extracted.

   Perform this action for all applicable sessions. By default the date format is YYYYMMDDHH24MISS. For example, the default date of 01/01/1970 would appear as 19700101000000.
3. Save and close the file.

Process of Configuring Workforce Payroll for Oracle 11i

This section contains configuration instructions for Payroll specific to Oracle 11i.

To configure Workforce Payroll for Oracle 11i, perform the following tasks:

- Modifying the Workforce Payroll Filters on page 302
- Improving ETL Performance for Workforce Payroll on page 302

Modifying the Workforce Payroll Filters

This task is a step in the Process of Configuring Workforce Payroll for Oracle 11i on page 301.

Siebel Customer-Centric Enterprise Warehouse is preconfigured to extract the input value name of Pay Value. Siebel Customer-Centric Enterprise Warehouse does not extract classification elements such as Information, Balance, or Employer. To modify the Payroll filter perform the following procedure.

NOTE: If you change the Payroll filter, you need to also change the Siebel Business Analytics metadata, so your reports are run correctly.

To modify the Payroll filters

1. Open the `FILE_PARAMETERS_ORA11i.CSV` file using Notepad in the installation directory in `$pmserver\srcfiles`.
2. Search for the strings in the following table, and change or delete these to match your requirements.

PAY_INPUT_VALUES_F.NAME='Pay Value'
CLASSIFICATION_NAME NOT LIKE '%Information%'
CLASSIFICATION_NAME NOT LIKE '%Employer%'
CLASSIFICATION_NAME NOT LIKE '%Balance%

   NOTE: There are two entries for each string—Payroll Extract and Pay Type Extract. You need to make the same changes for both entries.

3. Save and close the file.

Improving ETL Performance for Workforce Payroll

This task is a step in the Process of Configuring Workforce Payroll for Oracle 11i on page 301.

Siebel Customer-Centric Enterprise Warehouse is preconfigured to use the Rule Base Optimization to improve the performance of the Payroll extract mapping. Yon can change this by using the following procedure.

To improve ETL performance for Payroll

1. Open the `FILE_PARAMETERS_ORA11i.CSV` file using Notepad in the installation directory in `$pmserver\srcfiles`.
2. Search and change the following string:

```
/*+ RULE*/
```

3. Save the file and exit.

About Aggregating the Payroll Table for Siebel Enterprise Workforce Analytics

You can aggregate the Payroll table to a different time levels, and aggregate levels of Employees, Jobs, and Payment Types dimensions. There are two time grain parameters to configure for this aggregate table and these parameters need to have the same value.

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

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

The Payroll aggregate table is fully loaded from the base table in the initial ETL run. The table can grow to millions of records. The Payroll aggregate table is not fully reloaded from the base table after an ETL run. Siebel Customer-Centric Enterprise Warehouse minimizes the incremental aggregation effort, by modifying the aggregate table incrementally as the base table is updated. Siebel Customer-Centric Enterprise Warehouse looks for new records in the base table during the incremental ETL. This process is done in two steps:

1. There are new records in the IA_PAYROLL table, which are inserted after the last ETL run. These new records are inserted into the NU_PAYROLL table. This step is part of the post load-processing workflow, and the mapping is suffixed with POST.
2. Siebel Customer-Centric Enterprise Warehouse aggregates the NU_PAYROLL table, and joins it with the IA_PAYROLL_A4 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 INCR.

Aggregating the Payroll Table for Siebel Enterprise Workforce Analytics

To load the Payroll aggregate table (IA_PAYROLL_A4), you need to configure the post-load processing parameter file, `file_parameters_plp.csv`, and run the initial workflow and then the incremental workflow.

To configure the file_parameters_plp.csv parameters file

1. Open the `file_parameters_plp.csv` file using Microsoft WordPad or Notepad in the `$pmserver\srcfiles` folder.

2 Replace the default PARM_SVALUE2 values with your new values.

For a list of values for each parameter see the About Aggregating the Payroll Table for Siebel Enterprise Workforce Analytics on page 303.

The default values are shown in the following table:

PARM_TYPE	PARM_CODE	PARM_SVALUE_1	PARM_SVALUE_2	PARM_SVALUE_3
PLP	S_M_PLP_PAYROLL_A4_INIT:GRAIN	S	'MONTH'	—
PLP	S_M_PLP_PAYROLL_A4_POST:GRAIN	S	MONTH	MPLT_LKP_PERIOD_START_END_DK

NOTE: You need to use single quotes for the S_M_PLP_PAYROLL_A4_INIT:GRAIN session value.

3 Save and close the file.

Domain Values and CSV Worksheet Files for Siebel Enterprise Workforce Analytics

Table 62 lists the CSV worksheet files and the domain values for Siebel Enterprise Workforce Analytics in the $pmserver\LkpFiles folder.

For more information on configuring domain values with CSV worksheet files, see About Domain Values on page 154 and Configuring the Domain Value Set with CSV Worksheet Files on page 159.

Table 62. Domain Values and CSV Worksheet Files for Siebel Enterprise Workforce Analytics

Worksheet File Name	Description	Session
domainValues_Employment_ora11i.csv	Lists the User Person Type column and the corresponding domain values of employment category for the Oracle 11i application.	S_M_I_EMPLOYMENT_EXTRACT

Table 62. Domain Values and CSV Worksheet Files for Siebel Enterprise Workforce Analytics

Worksheet File Name	Description	Session
domainValues_Pay_Types_ora11i.csv	Lists the Classification Name and Element Name, and the corresponding domain values of pay type group for the Oracle 11i application.	S_M_I_PAY_TYPES_LOAD
domainValues_EventTypes_ora11i.csv	Lists the Event Type, Event Code, and Meaning, and the corresponding domain values of event group, event sub-group and event reason for the Oracle 11i application.	S_M_I_EVENT_TYPES_LOAD

Configuring Domain Values and CSV Worksheet Files for Siebel Enterprise Workforce Analytics

You need to configure the CSV files in Siebel Enterprise Workforce Analytics by mapping domain values to columns. The CSV worksheet files for Siebel Enterprise Workforce Analytics are in the `$pmserver\LkpFiles` folder.

For more information on configuring domain values with CSV worksheet files, see About Domain Values on page 154 and Configuring the Domain Value Set with CSV Worksheet Files on page 159.

To configure the domainValues_Employment_ora11i.csv

1. Identify the Person Types in your Orace11i source system by using the following SQL:

   ```
   SELECT DISTINCT SYSTEM_PERSON_TYPE, USER_PERSON_TYPE FROM

   PER_PERSON_TYPES

   WHERE SYSTEM_PERSON_TYPE IN

   ('EMP','OTHER','EMP_APL','EX_EMP','EX_EMP_APL','RETIREE','PRTN')

   ORDER BY 1, 2
   ```

   NOTE: If you have modified the Payroll filter, you need to also modify the SQL. For more information on modifying Payroll filters, see Modifying the Workforce Payroll Filters on page 302.

2. Open the `domainValues_Employment_ora11i.csv` file using Microsoft WordPad or Notepad in the `$pmserver\lkpfiles` folder.

3. Copy the Person Types to the SYSTEM_PERSON_TYPE and USER_PERSON_TYPE columns.

   Use commas to separate the entries.

4 Map each User Person Type to one domain value.

System Person Types are also extracted with User Person Type to help you map the domain values.

For more information Employment domain values, see *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.

NOTE: Incorrect mappings result in inaccurate calculations of Siebel Business Analytics metrics.

5 Save and close the file.

To configure the domainValues_Pay_Types_ora11i.csv

1 Identify the Pay Elements in your Orace11i source system by using the following SQL:

```
SELECT DISTINCT CLASSIFICATION_NAME, ELEMENT_NAME

FROM

PAY_ELEMENT_TYPES_F,

PAY_ELEMENT_CLASSIFICATIONS

WHERE

PAY_ELEMENT_CLASSIFICATIONS.CLASSIFICATION_ID =
PAY_ELEMENT_TYPES_F.CLASSIFICATION_ID AND

CLASSIFICATION_NAME NOT LIKE '%Information%' AND

CLASSIFICATION_NAME NOT LIKE '%Employer%' AND

CLASSIFICATION_NAME NOT LIKE '%Balance%'

ORDER BY 1, 2
```

2 Open the `domainValues_Pay_Types_ora11i.csv` file using Microsoft WordPad or Notepad in the `$pmserver\lkpfiles` folder.

3 Copy the Pay Elements to the CLASSIFICATION_NAME and ELEMENT_NAME columns.

Use commas to separate the entries.

4 Map each Element Name to one domain value.

Classification Names are also extracted with Element Names to help you map the domain values. If the element is not related to Payroll Pay Check, you can map the element to Other.

For more information on Pay Type domain values, see *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.

NOTE: Incorrect mappings result in inaccurate calculations of Siebel Business Analytics metrics.

5 Save and close the file.

To configure the domainValues_EventTypes_ora11i.csv

1. Identify the Event Types in your Orace11i source system by using the following SQL:

   ```
   SELECT DISTINCT LOOKUP_TYPE, LOOKUP_CODE, MEANING

   FROM FND_LOOKUP_VALUES

   WHERE LOOKUP_TYPE IN

   ('EMP_ASSIGN_REASON',

   'LEAV_REAS',

   'PROPOSAL_REASON')

   ORDER BY 1, 2, 3
   ```

2. Open the `domainValues_EventTypes_ora11i.csv` file using Microsoft WordPad or Notepad in the `$pmserver\lkpfiles` folder.

3. Except the first five rows in the file, delete all the other rows.

4. Copy the Event Types to the LOOKUP_TYPE, LOOKUP_CODE, and MEANING columns from row 6.

   Use commas to separate the entries.

5. Map each Event Type (LOOKUP_CODE) to one domain value for each of the 3 domain columns—IA_EVENT_GRP_CODE, IA_EVENT_SUBG_CODE, and IA_EVENT_REASON_CODE.

   Event Category (LOOKUP_TYPE) and Event Description (MEANING) are also extracted with Event Type to help you map the domain values.

   For more information on Event Type domain values, see *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.

   NOTE: Incorrect mappings result in inaccurate calculations of Siebel Business Analytics metrics.

6. Save and close the file.

16 Configuring Siebel Financial Analytics

This chapter describes how to configure certain objects for particular sources to meet your business needs. Siebel Financial Analytics consists of Siebel General Ledger Analytics, Siebel Payables Analytics, Siebel Receivables Analytics, and Siebel Profitability Analytics.

This chapter contains the following topics:

Overview of Siebel Financial Analytics

Siebel Financial Analytics consists of Siebel General Ledger Analytics, Siebel Payables Analytics, Siebel Receivables Analytics, and Siebel Profitability Analytics.

- **Siebel General Ledger Analytics.** The Siebel General Ledger Analytics application provides information to support your enterprise's balance sheet and provides a clearer understanding of the chart of accounts.

  The default configuration for the Siebel 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.

- **Siebel Payables Analytics.** The Siebel 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 Siebel 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 Siebel 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.

- **Siebel Receivables Analytics.** The Siebel 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 Siebel 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 Siebel 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.

- **Siebel Profitability Analytics.** The Siebel Profitability Analytics application provides cost analysis, revenue trends, and sales performance to provide an accurate picture of profit and loss. The information found in the Siebel Profitability Analytics application pertains to data found in the revenue and expense account groupings of your financial statements and chart of accounts. The Siebel 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 Siebel 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 Siebel Profitability Analytics application provides cost analysis, revenue trends, and profitability analysis at the products and customer level, and the income statement at the company and business divisions level.

Process of Configuring Siebel Financial Analytics for Oracle 11i

To configure the Siebel Financial Analytics application for Oracle 11i, perform the following tasks:

- Extracting Data Posted at the Detail-Level for Oracle 11i on page 311
- Mapping Siebel General Ledger Analytics Account Numbers to Group Account Numbers on page 312
- Filtering Extracts Based on Set of Books ID for Siebel General Ledger Analytics on page 313
- Configuring Siebel General Ledger Analytics Transaction Extracts on page 314
- Configuring General Ledger Cost Of Goods Extract on page 315
- Configuring the General Ledger Account Hierarchies on page 315
- Loading Hierarchies for Siebel General Ledger Analytics on page 316
- Configuring the General Ledger Balance ID on page 319
- Configuring AP Balance ID for Siebel Payables Analytics on page 320
- Configuring AR Balance ID for Siebel Receivables Analytics and Siebel Profitability Analytics on page 320

- Configuring the AR Adjustments Extract for Siebel Receivables Analytics on page 321
- Configuring the AR Schedules Extract on page 322
- Configuring the AR Cash Receipt Application Extract for Siebel Receivables Analytics on page 322
- Configuring the AR Credit-Memo Application Extract for Siebel Receivables Analytics on page 323
- Configuring the Customer Costs Lines and Product Costs Lines Tables for Siebel Profitability Analytics on page 324

Related Topic

- About the Customer Costs Lines and Product Costs Lines Tables for Siebel Profitability Analytics on page 324

Extracting Data Posted at the Detail-Level for Oracle 11i

This task is a step in the Process of Configuring Siebel Financial Analytics for Oracle 11i on page 310.

By default, the Siebel Customer-Centric Enterprise Warehouse assumes that the posting from your journal to your Oracle General Ledger is done at the summary level, and that references are maintained in Oracle General Ledger for AP and AR subledgers. If import references are not maintained in Oracle General Ledger and the posting from AP and AR is at the detail-level, then modify the filter condition in M_I_GL_XACTS_JOURNALS_EXTRACT and disable the S_M_I_XACTS_IMP_GLRF_EXTRACT session so that only the session S_M_I_GL_XACTS_JOURNALS_EXTRACT loads into the common table TI_STAGE_GLRF_DERV.

To modify the filter condition for posting at the detail level

1. In the PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. Open the M_I_GL_XACTS_JOURNALS_EXTRACT mapping.
3. Select the FIL_GL_XACTS_JOURNAL filter, and click the Properties tab to edit the filter condition.

   To load postings at the detail level, replace the 1=2 condition with 1=1.
4. Validate and save your changes to the repository.

To disable the session for Siebel General Ledger Analytics extract

1. In PowerCenter Workflow Manager, open the Configuration for Oracle Applications v11i folder.
2. Expand the Oracle 11i Strategic Sourcing Application workflow.
3. Expand the O_FACTS_EXTRACT worklet and the O_Facts_Extract_Finance_Common_AP_AR_GL worklet
4. Double-click the S_M_I_XACTS_IMP_GLRF_EXTRACT session to open the Edit Tasks window.
5. Click Disable this task, and click OK.

6 Repeat Step 2 to Step 5 for the Oracle 11i Siebel Enterprise Sales Analytics application and the Oracle 11i Finance application workflows.

NOTE: As a best practice, you must move unused sessions to another folder to avoid error messages and preserve it for future use.

Mapping Siebel General Ledger Analytics Account Numbers to Group Account Numbers

This task is a step in the Process of Configuring Siebel Financial Analytics for Oracle 11i on page 310.

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 *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.

You can categorize your Oracle General Ledger accounts into specific group account numbers. You may use this information during data extraction as well as front-end reporting. The GROUP_ACCT_NUM field denotes the nature of the Siebel General Ledger Analytics accounts. For example, Cash account, Payroll account, and so on. Refer to the master_code column in the `file_group_acct_names_ora11i.csv` file for values you can use. For a list of the Group Account Number domain values, see *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.

There are two columns in the fact table that categories expenses—Xact Type Key (Cost Types) and Xact Type Alloc Key (Cost Allocation Type). The Xact Type Key categorizes the expenses into Marketing, Sales, Service, and so on. The Xact Type Alloc Key further categorizes these into Direct and Allocation expenses.

NOTE: It is critical that you map the Xact Type Key and Xact Type Alloc Key columns for reports to work.

The mappings to General Ledger Accounts Numbers are important for both Profitability Analysis (Income Statement) and General Ledger accounts.

The logic for assigning the accounts is located in the `file_group_acct_codes_ora11i.csv` file. Table 63 shows the layout of the `file_group_acct_codes_ora11i.csv` file.

Table 63. Layout of file_group_acct_codes_ora11i.csv File

SOB ID	FROM ACCT	TO ACCT	GROUP_ACCT_NUM
1	101010	101099	CA
1	131010	131939	FG INV
1	152121	152401	RM INV
1	171101	171901	WIP INV
1	173001	173001	PPE
1	240100	240120	ACC DEPCN
1	261000	261100	INT EXP

Table 63. Layout of file_group_acct_codes_ora11i.csv File

SOB ID	FROM ACCT	TO ACCT	GROUP_ACCT_NUM
1	181011	181918	CASH
1	251100	251120	ST BORR

In Table 63, in the first row, all accounts within the account number range from 101010 to 101099 containing a Set of Books (SOB) ID equal to 1 are assigned to Current Asset. Each row maps all accounts within the specified account number range and with the given Set of Books ID.

If you need to create a new group of account numbers, you can create new rows in the `file_group_acct_names_ora11i.csv` file. You can then assign GL accounts to the new group of account numbers in the `file_group_acct_codes_ora11i.csv` file.

NOTE: When you specify the Group Account Number, you must capitalize the letters and use the values in the `master_code` column of the `file_group_acct_names_ora11i.csv` file.

To map Siebel General Ledger Analytics Accounts to Group Account Numbers

1. Open the `file_group_acct_codes_ora11i.csv` file with Microsoft WordPad or Notepad in the `$pmserver\srcfiles` folder.
2. Edit the fields in the following table:

Field Name	Description
SOB ID	The set of books ID for the Siebel General Ledger Analytics accounts.
FROM ACCT and TO ACCT	The From Account and To Account specify the range of Siebel General Ledger Analytics accounts for the mapping. The value you specify comes from the value of the natural account segment of the Siebel General Ledger Analytics account.
GROUP_ACCT_NUM	This field denotes the nature of the Siebel General Ledger Analytics accounts. For example, Cash account, Payroll account, and so on. Refer to the file_group_acct_names_ora11i.csv file for values you can use.

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

3. Save and close the CSV file.

Filtering Extracts Based on Set of Books ID for Siebel General Ledger Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for Oracle 11i on page 310.

If you have multiple sets of books and want to use only some of them as sources for the extract, then you have to modify the Source Qualifier. For example, assume that you have four sets of books for your enterprise—a set of books for your U.S. organization (SOB_ID = 1), a set of books for your Japan organization (SOB_ID = 2), a set of books for your German organization (SOB_ID = 3), and a set of books for your enterprise as a whole (SOB_ID = 4). If you want to extract only the enterprise level information, you extract only transactions where the SOB_ID = 4. Therefore, in the Source Qualifier's SQL Query and User Defined Join fields, you must add the following filter statement:

```
AND SET_OF_BOOKS_ID IN (4)
```

To filter extracts based on Set of Books ID

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In Mapplet Designer, open the MPLT_BCI_GL_ACCOUNTS.
3. Double-click the Source Qualifier to open the Edit Transformations window, and click the Properties tab.
4. Insert the filter condition in the SQL Query field and in the User Defined Join field.

   For example, if you want to use only the Set of Books whose ID is 4, then you insert the following filter:

   ```
   AND GL_SETS_OF_BOOKS.SET_OF_BOOKS_ID IN (4)
   ```

   in the WHERE clause in the SQL Query field, and at the end of the statement in the User Defined Join field.
5. Validate and save changes to the repository.

Configuring Siebel General Ledger Analytics Transaction Extracts

This task is a step in the Process of Configuring Siebel Financial Analytics for Oracle 11i on page 310.

There are two separate transaction extracts for Siebel General Ledger Analytics—General Ledger Revenue and General Ledger COGS. By default, the Siebel General Ledger Analytics application extracts only Completed revenue and COGS that have been posted to the general ledger. Completed revenue transactions are those where the RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG = Y. If you want to extract incomplete revenue transactions, you can remove the filter in the Business Component.

You must modify both the regular mapplet (MPLT_BCI_GL_REVENUE) as well as the primary extract mapplet (MPLT_BCI_GL_REVENUE_PRIMARY).

To modify the extract filter for Siebel General Ledger Analytics Revenue

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In Mapplet Designer, open the MPLT_BCI_GL_REVENUE mapplet.

3. Double-click the Source Qualifier to open the Edit Transformations window, and click the Properties tab.

   In the User Defined Join field and in the SQL Query field, remove the statement:

   ```
   AND RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG = Y
   ```

4. Validate and save your changes to the repository.
5. Repeat Step 2 to Step 4 for the MPLT_BCI_GL_REVENUE_PRIMARY mapplet.

Configuring General Ledger Cost Of Goods Extract

This task is a step in the Process of Configuring Siebel Financial Analytics for Oracle 11i on page 310.

By default, the Siebel General Ledger Analytics application extracts only COGS transactions that have been posted to the general ledger. All COGS transactions that have been transferred satisfy the following condition—MTL_TRANSACTION_ACCOUNTS.GL_BATCH_ID <> -1. If you want to extract all transactions, you can remove the filter in the Business Component mapplet.

Because Oracle General Ledger never deletes posted transactions, there are no prebuilt primary extract and delete mappings for COGS data. Therefore, if you decide to remove this filter and begin extracting unposted transactions, you must also create primary extract and delete mappings similar to those used for Siebel General Ledger Analytics Revenue. You can use the primary extract mapping (M_I_GL_REVENUE_PRIMARY_EXTRACT), and the delete mapping (M_I_GL_REVENUE_DELETE) as models for the extract and delete mappings you are creating in Oracle 11i.

To modify the extract filter for General Ledger COGS

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In Mapplet Designer, open MPLT_BCI_GL_COGS.
3. Double-click the Source Qualifier to open the Edit Transformations window, and click the Properties tab.

   In the User Defined Join field and in the SQL Query field, remove the statement:

   ```
   AND MTL_TRANSACTION_ACCOUNTS.GL_BATCH_ID <> -1
   ```

4. Validate and save your changes to the repository.

Configuring the General Ledger Account Hierarchies

This task is a step in the Process of Configuring Siebel Financial Analytics for Oracle 11i on page 310.

The Siebel Customer-Centric Enterprise Warehouse prepackages a hierarchy table (IA_HIERARCHIES) to store hierarchy information. The Siebel General Ledger Analytics application uses this table to store Siebel General Ledger Analytics account hierarchy information.

General Ledger accounts are loaded as part of the Finance workflow.

To configure the General Ledger hierarchical information

1. Load the General Ledger Accounts data into the IA_GL_ACCOUNTS table.
2. Load all hierarchies that are configured in your source system into the IA_HIERARCHIES table.
3. Configure mappings and sessions to update the following hierarchy columns in the IA_GL_ACCOUNTS table—HIER1_KEY, HIER2_KEY, HIER3_KEY, HIER4_KEY, HIER5_KEY, and HIER6_KEY.

Loading Hierarchies for Siebel General Ledger Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for Oracle 11i on page 310.

The Siebel General Ledger Analytics application prepackages M_I_GL_HIER_DERIVE mapping to extract hierarchy definitions from Oracle 11i. To better assist you in loading your hierarchy information into the IA_HIERARCHIES table, Table 64 gives you descriptions of the columns presented in the table.

Table 64. Columns in the IA_HIERARCHIES Table

Column Name	Description
HIER_KEY	This surrogate key is generated for each hierarchy. This key must be linked to the HIERARCHY_KEY column in the IA_GL_ACCOUNTS table.
HIERARCHY_ID	This column uniquely identifies the hierarchy within a given category. For general ledger account hierarchies, the ID format is as follows: ‘GL_HIER’ || <top node of hierarchy> || HIER1_CODE... HIER10_CODE
HIER_CODE	This code represents a hierarchy. The hierarchy code defines the name of the hierarchy (for example, Balance Sheet, Profit and Loss, and so on).
HIER_NAME	This column provides the name of the entire hierarchy.
HIER_CAT_CODE	This code represents the category of the hierarchy. For general ledger account hierarchies, the category code is GL_HIER.
HIER_CAT_DESC	This is a description of the category to which the hierarchy belongs. For general ledger account hierarchies, the category description is General Ledger Hierarchy.
HIER_SUBCAT_CODE	This is the code of the subcategory of the hierarchy.
HIER_SUBCAT_DESC	This is a description of the subcategory of the hierarchy.
LEVEL_MIN_NUM	This is the starting number, which is set to 0. For general ledger accounts, this is the beginning account number that is associated with this hierarchy. All accounts including and between the LEVEL_MIN_NUM and LEVEL_MAX_NUM are included in this category.

Table 64. Columns in the IA_HIERARCHIES Table

Column Name	Description
LEVEL_MAX_NUM	This is the last number, which is set to 99999999. For general ledger accounts, this is the ending account number that is associated with this hierarchy. All accounts including and between the LEVEL_MIN_NUM and LEVEL_MAX_NUM are included in this category.
HIER[X]_CODE	Each of these code columns represents a level in the hierarchy, where [X] denotes the level. HIER1 is the highest level of the hierarchy, and HIER20 is the lowest. Each code name column (HIER[X]_NAME) corresponds to a code column (HIER[X]_CODE). The code of the highest level of the hierarchy is catenated with GL_ACCT to form the hierarchy of the IA_HIERARCHIES table.
HIER[X]_NAME	Each of these code name columns represents a level in the hierarchy, where X denotes the level. Hier1 is the highest level of the hierarchy, and HIER20 is the lowest. Each code name column (HIERX_NAME) corresponds to a code column (HIERX_CODE).
HIER_ATTR[X]_CODE and HIER_ATTR[X]_NAME	There are five sets of extension columns for code name pairs. The X represents the level, where the same level is shared by each code name pair, such as HIER_ATTR1_CODE and HIER_ATTR1_NAME
HIER_ATTR[X]_Text	These are extension columns to store additional text. There are three available for your use.

The Hierarchy ID is set for every unique hierarchy structure. The format is:

```
‘GL_HIER’ ~ HIER_CODE ~ HIER1_CODE... HIER20_CODE
```

where:

- GL_HIER specifies that the hierarchy applies to general ledger accounts
- HIER_CODE specifies the ID for a hierarchy structure
- HIER1_CODE ~ HIER2_CODE... HIER20_CODE specifies each of the unique hierarchy levels in the given hierarchy structure

For example, if one of the Siebel General Ledger Analytics account hierarchies has the following structure:

```
Account (A)=> Current Asset (CA) => Fixed Asset (FA) => Balance Sheet (BS)
```

where the Balance Sheet hierarchy level is the highest level of the hierarchy, and BS denotes the hierarchy code, then set the Hierarchy ID to:

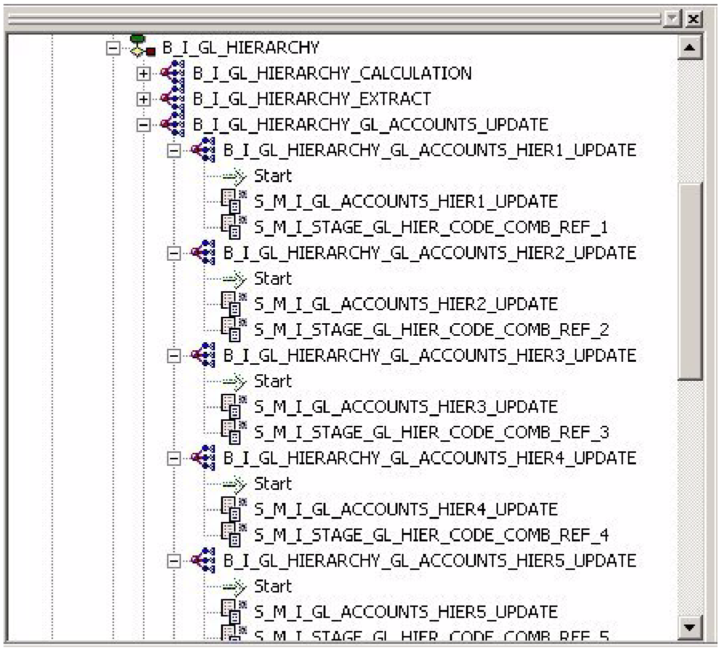
```
GL_HIER~BS~FA~CA~A
```

After you load the IA_HIERARCHIES table, you then must update the IA_GL_ACCOUNT table with the hierarchy information. Updating the IA_GL_ACCOUNT table requires a two-step process.

To load hierarchies in the IA_GL_ACCOUNTS table

1 You must first determine which hierarchies apply to each General Ledger Account.

To make this determination, you must modify the S_M_I_STAGE_GL_HIER_CODE_COMB_REF_[X] sessions, ([X] denotes the hierarchy number in the IA_GL_ACCOUNTS table). Because there are a maximum of six possible hierarchy structures in the IA_GL_ACCOUNT table, there are six sessions, where each session locates information for exactly one hierarchy structure, as shown in the following figure.



```
B_I_GL_HIERARCHY
  B_I_GL_HIERARCHY_CALCULATION
  B_I_GL_HIERARCHY_EXTRACT
  B_I_GL_HIERARCHY_GL_ACCOUNTS_UPDATE
    B_I_GL_HIERARCHY_GL_ACCOUNTS_HIER1_UPDATE
      Start
      S_M_I_GL_ACCOUNTS_HIER1_UPDATE
      S_M_I_STAGE_GL_HIER_CODE_COMB_REF_1
    B_I_GL_HIERARCHY_GL_ACCOUNTS_HIER2_UPDATE
      Start
      S_M_I_GL_ACCOUNTS_HIER2_UPDATE
      S_M_I_STAGE_GL_HIER_CODE_COMB_REF_2
    B_I_GL_HIERARCHY_GL_ACCOUNTS_HIER3_UPDATE
      Start
      S_M_I_GL_ACCOUNTS_HIER3_UPDATE
      S_M_I_STAGE_GL_HIER_CODE_COMB_REF_3
    B_I_GL_HIERARCHY_GL_ACCOUNTS_HIER4_UPDATE
      Start
      S_M_I_GL_ACCOUNTS_HIER4_UPDATE
      S_M_I_STAGE_GL_HIER_CODE_COMB_REF_4
    B_I_GL_HIERARCHY_GL_ACCOUNTS_HIER5_UPDATE
      Start
      S_M_I_GL_ACCOUNTS_HIER5_UPDATE
      S_M_I_STAGE_GL_HIER_CODE_COMB_REF_5
```

Each of these sessions derives the General Ledger Account references for a particular hierarchy, which is later used to update the HIER[X]_KEY column in the IA_GL_ACCOUNTS table. The 'HIER[X]_KEY' refers to the HIER_KEY surrogate key in the IA_HIERARCHIES table.

2 Load the appropriate hierarchy structure in the HIER[X]_KEY columns in the IA_GL_ACCOUNT table.

To load this hierarchy structure, you must modify the MPLT_SAI_GL_ACCOUNTS_HIER_UPDATE Source Adapter mapplet in the M_I_GL_ACCOUNTS_HIER_UPDATE mapping in Oracle 11i.

To configure hierarchy code in the session

1 In PowerCenter Workflow Manager, open the Configuration for Oracle Applications v11i folder.

2 Expand Oracle11i_Finance_Application_GL_ACCOUNT_Hierarchy/ W_O_GL_HIERARCHY_GL_ACCOUNTS_UPDATE/W_O_GL_HIERARCHY_GL_ACCOUNTS_HIER[X]_UPDATE.

3 In Worklet Designer, right-click the W_O_GL_HIERARCHY_GL_ACCOUNTS_HIER[X]_UPDATE, and click Open.

4 Double-click the appropriate session (S_M_I_STAGE_GL_HIER_CODE_COMB_REF_[X]) to open the Edit Tasks window.

5 In the Transformations tab, edit the SQL Query field by replacing the default hierarchy code.

The last statement in the SQL statement contains the hierarchy code. The hierarchy code determines the hierarchy for which the mapping session calculates the Siebel General Ledger Analytics references.

For example, in the S_M_I_STAGE_GL_HIER_CODE_COMB_REF_1 session, the statement:

```
TI_STAGE_GL_HIERRG.HIER_CODE='1042'
```

calculates all the Siebel General Ledger Analytics account references for the hierarchy code '1042'. This code is the AXIS_SET_ID defined in the Oracle Applications source table, RG_REPORT_AXIS_SETS. Depending on which hierarchy you want to store in the HIER1_KEY column in the IA_GL_ACCOUNTS table, the corresponding HIER_CODE must be set in the SQL statement.

To configure the hierarchy code in the update mapping

1 In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.

2 In Mapplet Designer, open the MPLT_SAI_GL_ACCOUNTS_HIER_UPDATE Source Adapter mapplet.

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

4 Edit the EXT_HIER[X]_ID port.

The value of each field must be the same as the set in the corresponding mapping—M_I_STAGE_GL_HIER_CODE_COMB_REF_X. For example, if you want to change the EXT_HIER1_ID port, then change the expression:

```
IIF(VAR_HIER_CODE='1042',INP_HIERARCHY_ID,SUBSTR(OD_HIER1_ID,1,INSTR(OD_HIER1_ID,I
NP_GL_ACCOUNT_NUM)-2))
```

In this case, you replace the 1042 with the applicable code used in the HIER_CODE column of the IA_HIERARCHIES table.

5 Validate and save your changes to the repository.

Configuring the General Ledger Balance ID

This task is a step in the Process of Configuring Siebel Financial Analytics for Oracle 11i on page 310.

By default, the General Ledger Balance ID is maintained at the following granularity for Oracle 11i:

```
TO_CHAR(SET_OF_BOOKS_ID)||’~’||TO_CHAR(INP_CODE_COMB_ID)
```

However, if you want to maintain your General Ledger Balance ID at a different grain, you can redefine the GL Balance ID for any applicable mapplets.

NOTE: You have to modify both the regular mapplet and update mapplets for AR and AP. For example, for AR, you would perform this process for MPLT_SAI_AR_XACTS, as well as the update mapplet MPLT_SAI_AR_XACTS_UPDATE for Oracle 11i.

To modify the General Ledger Balance ID

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In Mapplet Designer, open the MPLT_SAI_AR_XACTS Source Adapter mapplet.
3. Double-click the Expression transformation to open the Edit Transformations window and select the Ports tab.
4. Edit the Balance ID definition in the EXT_GL_BALANCE_ID column.
5. Validate and save your changes to the repository.
6. Repeat Step 2 to Step 5 for the MPLT_SAI_AR_XACTS_UPDATE mapplet.
7. For AP, repeat Step 2 to Step 5 for the MPLT_SAI_AP_XACTS_INSERT and MPLT_SAI_AP_XACTS_UPDATE mapplets.

Configuring AP Balance ID for Siebel Payables Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for Oracle 11i on page 310.

This section contains Siebel Payables Analytics configuration information that is specific to Oracle 11i.

By default, the Accounts Payable (AP) Balance ID is maintained at the following granularity:

```
GL_ACCOUNT_ID||'~'||VENDOR_SITE_ID||'~'||ORGANIZATION_ID
```

However, if you want to maintain your AP balance at a different grain, you can redefine the Balance ID value in the applicable mapplets. You have to modify both the insert mapplet (MPLT_SAI_AP_XACTS_INSERT) as well as the update mapplet (MPLT_SAI_AP_XACTS_UPDATE) for Oracle 11i.

To modify the Accounts Payable Balance ID

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In Mapplet Designer, open the MPLT_SAI_AP_XACTS_INSERT mapplet.
3. Double-click the Expression transformation to open the Edit Transformations window, and click the Ports tab to edit the Balance ID definition in the EXT_NU_BALANCE_ID column.
4. Validate and save your changes to the repository.
5. Repeat Step 2 to Step 4 for the MPLT_SAI_AP_XACTS_UPDATE mapplet.

Configuring AR Balance ID for Siebel Receivables Analytics and Siebel Profitability Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for Oracle 11i on page 310.

By default, the AR Balance ID is maintained at the following granularity:

```
SET_OF_BOOKS_ID||'~'||CODE_COMBINATION_ID||'~'||CUSTOMER_ID||'~'||CUSTOMER_SITE_US
E_ID
```

However, if you want to maintain your AR balance at a different grain, you can redefine the Balance ID value in the applicable mapplets. You have to modify both the regular mapplet (MPLT_SAI_AR_XACTS) as well as the update mapplet (MPLT_SAI_AR_XACTS_UPDATE) for Oracle 11i.

To modify the AR Balance ID

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In Mapplet Designer, open the MPLT_SAI_AR_XACTS mapplet.
3. Double-click the Expression transformation to open the Edit Transformations window, and click the Ports tab to edit the Balance ID definition in the EXT_NU_AR_BALANCE_ID column.
4. Validate and save your changes to the repository.
5. Repeat Step 2 to Step 4 for the MPLT_SAI_AR_XACTS_UPDATE mapplet.

Configuring the AR Adjustments Extract for Siebel Receivables Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for Oracle 11i on page 310.

By default, Siebel Receivables Analytics extracts only approved adjustment entries against accounts receivable transactions. *Approved adjustments* are entries where the AR_ADJUSTMENTS_ALL.STATUS = A. If you want to extract additional types of AR adjustment entries, you can remove the filter in the Business Component mapplet. By modifying or removing the filter, you can extract other entries, such as those that require more research, those that are rejected, and those that are not accrued charges.

You must modify both the regular mapplet (MPLT_BCI_AR_XACTS_ADJ) as well as the primary extract mapplet (MPLT_BCI_AR_XACTS_ADJ_PRIMARY). Repeat the following procedure for each mapplet.

To modify the extract filter for Accounts Receivable adjustments

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In Mapplet Designer, open the MPLT_BCI_AR_XACTS_ADJ mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations window, and click the Properties tab.

   In the SQL Query field and in the User Defined Join field, modify the statement:

   ```
   AND AR_ADJUSTMENTS_ALL.STATUS = A
   ```

4. Validate and save your changes to the repository.
5. Repeat Step 2 to Step 4 for the MPLT_BCI_AR_XACTS_ADJ_PRIMARY mapplet.

Configuring the AR Schedules Extract

This task is a step in the Process of Configuring Siebel Financial Analytics for Oracle 11i on page 310.

By default, Siebel Receivables Analytics extracts only completed schedules; that is, transactions where the RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG(+) = Y. If you want to extract additional types of AR schedule entries, you must remove the filter in the Business Component mapplet. By modifying or removing the filter, you can extract other entries, such as those that were marked as incomplete.

You must modify both the regular mapplet (MPLT_BCI_AR_XACTS_SCH) as well as the primary extract mapplet (MPLT_BCI_AR_XACTS_SCH_PRIMARY). Repeat the following procedure for each mapplet.

To modify the extract filter for Accounts Receivable schedules

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In Mapplet Designer, open the MPLT_BCI_AR_XACTS_SCH mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations window.

   In the User Defined Join field and in the SQL Query field, modify the statement:

   ```
   AND RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG(+) = Y
   ```

4. Validate and save your changes to the repository.
5. Repeat Step 2 to Step 4 for the MPLT_BCI_AR_XACTS_SCH_PRIMARY mapplet.

Configuring the AR Cash Receipt Application Extract for Siebel Receivables Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for Oracle 11i on page 310.

By default, Siebel Receivables Analytics extracts only confirmed, cash-receipt application entries against accounts receivable transactions. *Confirmed receipts* are entries where the AR_RECEIVABLE_APPLICATIONS_ALL.CONFIRMED_FLAG = Y OR NULL. If you want to extract additional types of cash-receipt application entries, you can remove the filter in the Business Component mapplet. By modifying or removing the filter, you can extract other entries, such as nonconfirmed applications.

You must modify both the regular mapplet (MPLT_BCI_AR_XACTS_APPREC) as well as the primary extract mapplet (MPLT_BCI_AR_XACTS_APPREC_PRIMARY).

To modify the extract filter for AR cash receipt application

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In Mapplet Designer, open the MPLT_BCI_AR_XACTS_APPREC mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations window, and click the Properties tab.

   In the User Defined Join field and in the SQL Query field, modify the statement:

```
AND NVL(AR_RECEIVABLE_APPLICATIONS_ALL.CONFIRMED_FLAG,'Y') = Y
```

4. Validate and save your changes to the repository.
5. Repeat Step 2 to Step 4 for the MPLT_BCI_AR_XACTS_APPREC_PRIMARY mapplet.

Configuring the AR Credit-Memo Application Extract for Siebel Receivables Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for Oracle 11i on page 310.

By default, Siebel Receivables Analytics extracts only confirmed, credit-memo application entries against accounts receivable transactions. *Confirmed credit memos* are entries where the `AR_RECEIVABLE_APPLICATIONS_ALL.CONFIRMED_FLAG = Y OR NULL`. If you want to extract additional types of AR credit-memo application entries, you can remove the filter. By modifying or removing the filter, you can extract other entries such as nonconfirmed, credit memos.

You must modify both the regular mapplet (`MPLT_BCI_AR_XACTS_APPCM`), as well as the primary extract mapplet (`MPLT_BCI_AR_XACTS_APPCM_PRIMARY`). Repeat the following procedure for each mapplet.

To modify the extract filter for Accounts Receivable Credit-Memo Application receipts

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. In Mapplet Designer, open the `MPLT_BCI_AR_XACTS_APPCM` mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations window, and click the Properties tab.

   In the User Defined Join field and in the SQL Query field, modify the statement:

   ```
   AND NVL(AR_RECEIVABLE_APPLICATIONS_ALL.CONFIRMED_FLAG,'Y') = Y
   ```

4. Validate and save your changes to the repository.
5. Repeat Step 2 to Step 4 for the `MPLT_BCI_AR_XACTS_APPCM_PRIMARY` mapplet.

About the Customer Costs Lines and Product Costs Lines Tables for Siebel Profitability Analytics

The Customer Costs Lines (IA_CUST_COSTLNS) table stores cost details by customers. The total cost by customers include both direct cost that is captured in the financial system by products, and the allocated costs that are captured in the costing system. The table also stores the source of allocations. For example, the Sales and Marketing costs are not be captured as direct costs by customers. However, at a later point in time, the costs could be allocated from a combination of General Ledger accounts or Cost Centers into various customers. This table stores the source cost centers and General Ledger accounts as well. The Customer Costs can be categorized by Sales, Marketing, Service, and Operating costs. These could be further broken down to Salaries, Events, and Promotions. Siebel Customer-Centric Enterprise Warehouse provides a set of categories, and these can be changed depending on the user needs and cost components by which customers are tracked. The actual cost lines are also tracked in this table. Apart from Customers, there are number of other dimensions by which the costs can be tracked such as Sales Region, Sales Geography, Company, Business Area and the associated hierarchies. The different cost lines such as Sales, Marketing, and Operating costs, have different details and many dimensions are included in this table. Some of the dimension keys are not be applicable to certain cost components. It is important that an organization identifies the dimensional keys that are used for Product Cost Analysis for various components.

The Product Costs Lines (IA_PROD_COSTLNS) table stores cost details by products. The total cost by products include both direct cost that is captured in the financial system by products, and the allocated costs that are captured in the costing system. The table also stores the source of allocations. For example, the Sales and Marketing costs are not captured as direct costs by products. However, at a later point in time, the costs are allocated from a combination of General Ledger accounts and Cost Centers into various products. This table also stores the source cost centers and General Ledger accounts. The product costs can be categorized by Sales, Marketing, Service, and Operating costs. These could be further broken down into Salaries, Events, and Promotions. Siebel Customer-Centric Enterprise Warehouse provides a set of common categories, and these can be changed depending on the user needs and cost components by which products are tracked. The actual cost lines are also tracked in this table. Apart from products, there are number of other dimensions by which the costs are tracked such as Sales Region, Sales Geography, Company, Business Area, and the associated hierarchies. The different cost lines such as Sales, Marketing, and Operating costs, have different details and many dimensions are included in this table. Some of the dimension keys are not applicable to certain cost components. It is important that an organization identifies the dimensional keys that is used for Product Cost Analysis for various components.

Configuring the Customer Costs Lines and Product Costs Lines Tables for Siebel Profitability Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for Oracle 11i on page 310.

In Siebel Profitability Analytics, the Customer Costs Lines and Product Costs Lines fact tables store the costing and expenses for the Profitability functional area. You need to use these tables with General Ledger Revenue and General Ledger COGS fact tables.

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

To load the Customer Costs Lines and Product Costs Lines tables

1. Open the `file_cust_costlns.csv` file in the `$pmserver\srcfiles` folder.
2. Insert a record into the file for each customer costing transaction you want to load into the Customer Cost fact table.

   For more information on how to populate each field in the CSV file, please see the `Cust_Costlns_Interface_description.xls` file.

   For the SOURCE_ID column, you need to provide the same source identifier value as in the `file_parameters_ora11i.csv` file.

3. Save and close the file.
4. Repeat Step 1 to Step 3 for the `file_prod_costlns.csv` file.
5. Open Workflow Manager, open the Configuration for Universal Source folder, and run the Universal_Finance_Profitability workflow.

   NOTE: By default, the Universal_Finance_Profitability workflow loads the Transaction Type dimension and the Customer Costs Lines and Product Costs Lines fact tables. If you need to load other dimensions, open up the Extract_Dimensions and Load_Dimensions worklets and enable the corresponding dimension extract and load sessions.

Extract, Transform, and Load for SAP R/3

NOTE: This section is only relevant if you are implementing Siebel Financial Analytics and Siebel Enterprise Sales Analytics.

The extract, transform, and load (ETL) method used for SAP R/3 data depends on the granularity in which SAP R/3 presents the data, as well as whether or not you want to integrate the Siebel Enterprise Sales Analytics application. Usually, SAP R/3 updates sale and purchase subledger data in Siebel General Ledger Analytics at the header/date level.

There are different sets of workflows to extract, apportion, and load the data into the various General Ledger tables in the data warehouse, depending on whether your data is posted at the header level or the detail level.

Fact Table ETL Process for Header-Level Sales Data for SAP R/3

The Siebel General Ledger Analytics application extracts invoice and sales order header-level data from the SAP R/3 tables BKPF (Accounting Document Header table) and BSEG (Accounting Document Segment table) and loads them into six staging tables. The entire ETL data flow is comprised of four areas:

- Extracting Sales Order and Invoice Data
- Loading Invoice and Sales Order Header Tables
- Apportioning Amounts and Quantities
- Loading General Ledger Staging Table

Each of these areas is discussed in the following sections.

About Extracting Sales Order and Invoice Data

Before loading data at the header level for sales orders and invoice data, Siebel General Ledger Analytics extracts transactional data from the SAP R/3 tables BKPF and BSEG and loads it into three staging tables:

- Siebel General Ledger Analytics staging table (TS_STAGE_GL)
- Sales Order Header staging table (TS_STAGE_SO_HDR)
- Invoice Header staging table (TS_STAGE_IV_HDR)

By default, only those transactions from BKPF and BSEG that have a status of NULL or S are extracted. Null status implies that the transactions are already posted. The status S marks the record as a noted item.

About Loading Invoice and Sales Order Header Tables

In the load process, the Invoice Header and Sales Order Header tables are populated by different sources. The TS_STAGE_IV_HDR, IA_SALES_IVCLNS, and OD_SALES_IVCLNS tables load the Invoice Header table (TS_STAGE_GL_IHD). The TS_STAGE_IV_HDR supplies header records for invoice data. Using these headers, you can extract sales invoice line item data from IA_SALES_IVCLNS table and aggregate it to provide total amount (NET_DOC_AMT), total quantity (INVOICED_QTY), and total number of items (TOTAL_ITEMS) for each invoice. These aggregated amounts are stored in the TS_STAGE_GL_IHD header table.

The method in which Siebel General Ledger Analytics loads the Invoice Header table is similar to the loading of the Sales Order Header table. The TS_STAGE_SO_HDR, IA_SALES_ORDLNS, and IA_SALES_HIST tables load the Sales Order Header table (TS_STAGE_GL_SHD). The TS_STAGE_SO_HDR table supplies header records for sales order data. Using these headers, you can extract sales order line item data from IA_SALES_ORDLNS table and aggregate it to provide total amount (NET_DOC_AMT), total quantity (SALES_ORDER), and total number of items (TOTAL_ITEMS) for each sales order. These aggregated amounts are stored in the TS_STAGE_GL_SHD header table.

About Apportioning Amounts and Quantities

The Invoice and Sales Order Header staging tables record total amounts and quantities for each document number (invoice number or sales order number). However, you must break down these amounts into different segments, so that you can load the appropriate amounts into the applicable staging table. There are six finance-related staging tables involved with apportioning amounts and quantities:

- Accounts Payable Transaction staging table (TS_AP_XACTS)
- Accounts Receivable Transaction staging table (TS_AR_XACTS)
- Tax Transaction staging table (TS_TAX_XACTS)
- General Ledger Cost of Goods Sold staging table (TS_GL_COGS)
- General Ledger Revenue staging table (TS_GL_REVENUE)
- General Ledger Other staging table (TS_GL_OTHER)

The Siebel General Ledger Analytics application prepackages logic to apportion header-level data to load the appropriate amounts into the corresponding staging table. The following examples illustrate the concept of apportioning data, by looking at how Siebel General Ledger Analytics apportions header-level invoice data to derive the detail-level amounts.

As previously stated, the General Ledger Header staging table (TS_STAGE_GL_IHD) stores header-level invoice data, while the IA_SALES_IVCLNS table stores line item invoice data, including the line item amounts for each invoice document. These line item amounts are aggregated by invoice number and loaded into the appropriate header record in the TS_STAGE_GL_IHD staging table.

Segment ratios in the Siebel General Ledger Analytics application apportion the header-level amounts into separate amounts based on each of the following segments—Revenue, Tax, and Freight. This allows the total amount of each invoice to be separated into those same segments of Revenue, Tax and Freight.

The segment ratios in Siebel General Ledger Analytics are created using two tables—TS_STAGE_GL and TS_STAGE_GL_IHD. The TS_STAGE_GL table stores total amounts for each segment for each order, and the TS_STAGE_GL_IHD stores the total amount for each Invoice. Revenue ratio equals the revenue amount divided by the total invoiced amount. Tax ratio equals the tax amount divided by the total invoiced amount. Freight ratio equals the freight amount divided by total invoiced amount.

The segmentation can vary in many ways depending on your business requirements. There can be a different number of segments, different segment types, or you may use segments to apportion quantities instead of amounts. In addition to the invoice data, the same concept also applies to sales order data that is posted at the header level.

About Loading Siebel General Ledger Analytics Staging Tables

After the sales order and invoice data is apportioned, the data is loaded into the applicable staging table. Using a lookup performed in the TS_STAGE_FIN_STMT staging table on Company Code and General Ledger Account number, the Siebel General Ledger Analytics application determines the financial statement that each segment amount belongs to. The possible financial statements are Accounts Payable, Accounts Receivable, Tax, General Ledger Revenue, General Ledger Cost of Goods Sold, and General Ledger Others. Others are defined as any financial statements that do not belong to one of the other five defined categories.

When the lookup determines the type of financial statement item, it loads the data into the appropriate staging table.

NOTE: The General Ledger Balance ID in the fact tables must be the same as the Key ID in the General Ledger Balance table. Keeping these two column values the same verifies that the same granularity, or incremental level, is maintained. If there is a disparity, the resulting balances may be skewed and misinterpreted. Also, note that Accounts Receivable and Accounts Payable have different Balance IDs due to their distinct granularity.

About Fact Table ETL Process for Detail-Level Information for SAP R/3

The Siebel General Ledger Analytics application uses the Group ID when data is posted at a detail level. The Group ID identifies one set of offsetting debits and credits. Each time the debits and credits are offset, the Group ID is reset.

In relation to Siebel General Ledger Analytics, the Group ID changes each time the debits and credits offset each other. Although the sales order number is the same, the Group ID changes. In this way, the Group ID relates a single record on the sales order side to multiple records in Siebel General Ledger Analytics. From the sales order perspective, the Group ID changes when the line item changes.

Similar to loading header-level data, the Siebel General Ledger Analytics application also loads detail-level data from the SAP R/3 tables BKPF and BSEG into six staging tables, then into their IA tables. However, the Siebel General Ledger Analytics application uses staging tables for detail-level data to determine three data streams—invoice, sales order, and others.

When loading data at the detail level, Siebel General Ledger Analytics first extracts and loads transactional data from the SAP R/3 tables BKPF and BSEG into the same three staging tables that it does for loading header level information—TS_STAGE_GL, TS_STAGE_SO_HDR, and TS_STAGE_IV_HDR. Before loading data into TS_STAGE_GL, the Group ID is generated by checking the debit and credit amounts.

In this next phase of loading detail-level information, the Siebel General Ledger Analytics application selects all invoice lines from IA_SALES_IVCLNS that exist in TS_STAGE_IV_HDR, and generates Group IDs for each combination of invoice and line items. A unique Group ID is created when the invoice line changes. The data is then loaded into the staging table TS_STAGE_GL_ILN.

For invoice lines data, sales invoice lines data is extracted from IA_SALES_INVLNS that exists in the TS_STAGE_IV_HDR table. The Group ID is selected from IA_SALES_HIST, and the data is then loaded into the staging table TS_STAGE_GL_SLN.

When creating header tables, the tables TS_STAGE_GL, TS_STAGE_IV_HDR, and TS_STAGE_GL_ILN join to get the keys, split the transactions, and set the Balance ID. In addition, a lookup is performed on the TS_STAGE_FIN_STMT staging table. The data then loads into the six staging areas, which are AP Transactions, AR Transactions, Tax Transactions, General Ledger Revenue, General Ledger Cost of Goods Sold, and General Ledger Others.

Similar to the Invoice Lines data, when creating header tables, the subitems of the sales order lines are created. The tables (TS_STAGE_GL, TS_STAGE_SO_HDR, and TS_STAGE_GL_SLN) join to retrieve the keys, split the transactions, and set the Balance ID. In addition, a lookup is performed on the TS_STAGE_FIN_STMT staging table. The data then loads into the six staging areas.

Siebel General Ledger Analytics Information for Nonsales Transactions for SAP R/3

Loading transaction information other than sales is handled in the same fashion as sales order lines and invoice lines data. All the transactions from TS_STAGE_GL that do not exist in TS_STAGE_IV_HDR are selected. Also the tables TS_STAGE_IV_HDR, TS_STAGE_SO_HDR, and TS_STAGE_GL join to get the keys, split the transactions and set the Balance ID. In addition, a lookup is performed on the TS_STAGE_FIN_STMT staging table. Data then loads into the six staging areas.

Process of Configuring Siebel Financial Analytics for SAP R/3

To configure Siebel Financial Analytics for SAP R/3, perform the following tasks:

- Extracting Data Posted at the Header Level for SAP R/3 on page 329
- Configuring the Group Account Number Categorization for Siebel General Ledger Analytics on page 330
- Configuring the Transaction Types for Siebel Financial Analytics on page 331
- Configuring the General Ledger Account Hierarchies on page 332
- Configuring Hierarchy ID in Source Adapter for Siebel General Ledger Analytics on page 334
- Configuring the Siebel General Ledger Analytics Balance Extract on page 335
- Configuring the Siebel Payables Analytics Balance Extract on page 336
- Configuring the Siebel Receivables Analytics Balance Extract on page 338
- Configuring the Customer Costs Lines and Product Costs Lines Tables for Siebel Profitability Analytics on page 337

Extracting Data Posted at the Header Level for SAP R/3

This task is a step in the Process of Configuring Siebel Financial Analytics for SAP R/3 on page 329.

NOTE: This section is only relevant if you are implementing Siebel Financial Analytics and Siebel Enterprise Sales Analytics.

The Siebel General Ledger Analytics application extracts sales information posted to the Siebel General Ledger Analytics at the detail level by default. However, you may configure Siebel General Ledger Analytics if your installation of SAP R/3 is configured to store data at the header level. For information on how the Siebel General Ledger Analytics loads data posted at the header level, see Fact Table ETL Process for Header-Level Sales Data for SAP R/3 on page 326.

To configure the fact extract to extract data posted at the header level

1. In PowerCenter Workflow Manager, open the Configuration for SAP R/3 folder.
2. Expand the `SAP_EnterpriseSales_Application` workflow and the `Extract_Facts` worklet.
3. Right -click on the worklet `S_FACTS_EXTRACT_FINANCE_DETAIL_SALES_COMBINATION_ONLY`, and select Edit.
4. In the Edit Tasks window, select the Disable this task check box.
5. Right -click on the worklet `S_FACTS_EXTRACT_FINANCE_HEADER_SALES_COMBINATION_ONLY`, and select Edit.
6. In the Edit Tasks window, clear the Disable this task check box.
7. Expand the `SAP_EnterpriseSales_Application` workflow and the `Load_Facts` worklet.
8. Right -click on the worklet `S_FACTS_LOADS_FINANCE_DETAIL_SALES_COMBINATION_ONLY`, and select Edit.
9. In the Edit Tasks window, select the Disable this task check box.
10. Right -click on the worklet `S_FACTS_LOADS_FINANCE_HEADER_SALES_COMBINATION_ONLY`, and select Edit.
11. In the Edit Tasks window, clear the Disable this task check box.

Configuring the Group Account Number Categorization for Siebel General Ledger Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for SAP R/3 on page 329.

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 *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.

The mappings to General Ledger Accounts Numbers are important for both Profitability Analysis (Income Statement) and General Ledger accounts.

The Group Account Number categorizes each Siebel General Ledger Analytics account record in the IA_GL_ACCOUNTS dimension table. Each Siebel General Ledger Analytics Account is assigned a Group Account Number. The logic for assigning the accounts is located in the `file_group_acct_codes_sapr3.csv` file. For example, this file might have the layout shown in Table 65.

In Table 65, all accounts within the account number range from 1000 to 139800 containing Hierarchy Code equal to GL_HIER and Company Code equal to 1 are assigned to Account Depreciation (ACCN DEPCN). Similarly, looking at the second row in the table, all accounts within the account number range from 140000 to 141099 containing Hierarchy Code equal to GL_HIER and Company Code equal to 1 are assigned to AR. Each row categorizes all accounts within the specified account number range and with the given combination of Hierarchy Code and Company Code.

Because different Hierarchy Code and Company Code combinations can contain the same account numbers, the Hierarchy Code and Company Code is also contained in this file. By keeping note of the account number as well as the Hierarchy Code and Company Code, you can uniquely identify each transaction in the data warehouse.

Table 65. Sample Layout of file_group_acct_codes_sapr3.csv File

HIER_CODE	COMPANY_CODE	FROM_ACCT	TO_ACCT	GROUP_ACCT_NUM
GL_HIER	1	1000	139800	ACC DEPCN
GL_HIER	1	140000	141099	AR
GL_HIER	1	141100	142400	CA
GL_HIER	1	143000	143000	AP
GL_HIER	1	143100	143100	AR
GL_HIER	1	143110	143200	CASH

NOTE: You must capitalize the letters for the GROUP_ACCT_NUM column.

To assign accounts to Group Account Numbers

1. Open the `file_group_acct_codes_sapr3.csv` file in the `$pmserver\srcfiles` folder.
2. Remove all entries in the file and insert your account ranges for each Group Account Number.

   For list of values for Group Account Numbers, please see the MASTER_CODE column in the `file_group_acct_names_sapr3.csv` file.
3. Save and close the file.

Configuring the Transaction Types for Siebel Financial Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for SAP R/3 on page 329.

Transaction types are stored in the IA_XACT_TYPES table. For a list of domain values in the Transaction Types table, see the *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.

In Siebel Financial Analytics, the transaction type of a record is determined by the following three attributes:

- POSTING_KEY
- ACCT_DOC_TYPE_CODE
- SPECIAL_G_L_INDICATOR

You can configure the transaction type by editing the xact_type_code_sap.csv file. Table 66 shows a sample layout of the xact_type_code_sap.csv file.

In Table 66, an accounting document item with a POSTING_KEY equal to 1, an ACCT_DOC_TYPE_CODE equal to AA, and a SPECIAL_G_L_INDICATOR with no value, appears in the Siebel Customer-Centric Enterprise Warehouse as an Invoice record.

Table 66. Sample Layout of xact_type_code_sap.csv File

POSTING_KEY	ACCT_DOC_TYPE_CODE	SPECIAL_G_L_INDICATOR	IA_XACT_TYPE_CODE	IA_XACT_STYPE_CODE	XACT_CODE	SOURCE_ID
1	AA		ORIGINAL	INVOICE	RECEIVABLE	SAPR3
1	DA		ORIGINAL	INVOICE	RECEIVABLE	SAPR3
1	DR		ORIGINAL	INVOICE	RECEIVABLE	SAPR3
1	RV		ORIGINAL	INVOICE	RECEIVABLE	SAPR3
1	SA		ORIGINAL	INVOICE	RECEIVABLE	SAPR3
1	DA		ORIGINAL	CR MEMO	RECEIVABLE	SAPR3
1	AB		ORIGINAL	OTHER	RECEIVABLE	SAPR3

To configure the transaction types

1. Open the xact_type_code_sap.csv file in the $pmserver\srcfiles folder.
2. Insert your transaction types.

   For a list of domain values in the Transaction Types table, see the *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.
3. Save and close the file.

Configuring the General Ledger Account Hierarchies

This task is a step in the Process of Configuring Siebel Financial Analytics for SAP R/3 on page 329.

The Siebel General Ledger Analytics application prepackages a hierarchy table (IA_HIERARCHIES) to store all hierarchy information. The Siebel General Ledger Analytics application uses this table to store Siebel General Ledger Analytics account hierarchy information. For the hierarchies to work, you first have to load the Siebel General Ledger Analytics account hierarchy information into IA_HIERARCHIES. Second, you have to configure the Source Adapter for the GL_ACCOUNTS_LOAD mapping so that the hierarchy key in the IA_GL_ACCOUNTS table links to the IA_HIERARCHIES table. After these two tasks are accomplished, you are ready to load General Ledger Account data into the IA_GL_ACCOUNTS table.

You can relate up to six possible hierarchy structures to each general ledger account, using the IA_HIERARCHIES table:

1. Load the primary hierarchy in the IA_GL_ACCOUNTS table.
2. If you want additional hierarchies, you begin by adding hierarchies to the HIERX_KEY ports.
3. Load the first additional hierarchy in the HIER1_KEY port, and then load the following hierarchy structure in the HIER2_KEY port, and the other hierarchies in sequence.

Loading the Hierarchy Definitions Into the IA_HIERARCHIES Table, The Siebel General Ledger Analytics application prepackages the M_S_GL_HIERARCHY_EXTRACT mapping to extract hierarchy definitions from SAP R/3. To better assist you in loading your hierarchy information into the IA_HIERARCHIES table, Table 67 provides descriptions of the columns presented in the table.

Table 67. Columns in the IA_HIERARCHIES Table

Column Name	Description
HIER_KEY	This surrogate key is generated for each hierarchy. This key must be linked to the HIERARCHY_KEY column in the IA_GL_ACCOUNTS table.
HIERARCHY_ID	This column uniquely identifies the hierarchy within a given category. For general ledger account hierarchies, the ID format is as follows: ‘GL_HIER’ || <top node of hierarchy> || HIER1_CODE... HIER10_CODE
HIER_CODE	This code represents a hierarchy. The hierarchy code defines the name of the hierarchy (for example, Balance Sheet, Profit and Loss, and so on).
HIER_NAME	This column provides the name of the entire hierarchy.
HIER_CAT_CODE	This code represents the category of the hierarchy. For general ledger account hierarchies, the category code is GL_HIER.
HIER_CAT_DESC	This is a description of the category to which the hierarchy belongs. For general ledger account hierarchies, the category description is General Ledger Hierarchy.
HIER_SUBCAT_CODE	This is the code of the subcategory of the hierarchy.
HIER_SUBCAT_DESC	This is a description of the subcategory of the hierarchy.

Table 67. Columns in the IA_HIERARCHIES Table

Column Name	Description
LEVEL_MIN_NUM	This is the first number, which is set to 0. For general ledger accounts, this is the first account number that is associated with this hierarchy. All accounts including and between the LEVEL_MIN_NUM and LEVEL_MAX_NUM are included in this category.
LEVEL_MAX_NUM	This is the last number, which is set to 99999999. For general ledger accounts, this is the last account number that is associated with this hierarchy. All accounts including and between the LEVEL_MIN_NUM and LEVEL_MAX_NUM are included in this category.
HIERX_CODE	Each of these code columns represents a level in the hierarchy, where [X] denotes the level. HIER1 is the highest level of the hierarchy, and HIER20 is the lowest. Each code name column (HIER[X]_NAME) corresponds to a code column (HIER[X]_CODE). The code of the highest level of the hierarchy is catenated with GL_ACCT to form the hierarchy of the IA_HIERARCHIES table.
HIER_ATTRX_CODE and HIER_ATTRX_NAME	There are five sets of extension columns for code name pairs. The X represents the level, where the same level is shared by each code name pair, such as HIER_ATTR1_CODE and HIER_ATTR1_NAME.
HIER_ATTRX_TEXT	These are extension columns to store additional text. There are three available for your use.

You must set the Hierarchy ID for every hierarchy structure. The following format is recommended:

```
'GL_ACCT' || <Hierarchy Code for top node of hierarchy structure>
```

where:

- GL_ACCT specifies that the hierarchy applies to general ledger accounts
- <Hierarchy Code for top node of hierarchy> is the hierarchy code that specifies the highest level of the hierarchy

For example, if one of the general ledger account hierarchies has the following structure:

```
Account (A)=> Current Asset (CA) => Fixed Asset (FA) => Balance Sheet (BS)
```

where the Balance Sheet hierarchy level is the highest level of the hierarchy, and BS denotes the hierarchy code, then set the Hierarchy ID to:

```
GL_HIER~BS~FA~CA~A
```

Configuring Hierarchy ID in Source Adapter for Siebel General Ledger Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for SAP R/3 on page 329.

To configure the Siebel General Ledger Analytics application to link to the General Ledger Account hierarchies from the IA_HIERARCHIES table, you must configure the Source Adapter in the GL_ACCOUNTS_LOAD mapping. By default, the HIERARCHY_KEY port is set to NULL. To populate the Siebel General Ledger Analytics account hierarchies when you load the IA_GL_ACCOUNTS table, use the same hierarchal structure that you used in IA_HIERARCHIES. For more information on the hierarchal structure used in IA_HIERARCHIES, see Configuring the General Ledger Account Hierarchies on page 332.

To configure the Hierarchy ID

1. In PowerCenter Designer, open the Configuration for SAP R/3 folder.
2. In Mapping Designer, open the mapping M_S_GL_ACCOUNTS_LOAD.
3. Double-click the Expression transformation to open the Edit Transformations window.
4. Edit the HIERX_KEY port (X denotes a distinct hierarchy structure).
5. Change NULL to reflect the Hierarchy ID specified in the HIERARCHY_KEY column in the IA_HIERARCHIES table.
6. Validate and save your changes to the repository.

Configuring the Siebel General Ledger Analytics Balance Extract

This task is a step in the Process of Configuring Siebel Financial Analytics for SAP R/3 on page 329.

The Balance and Key IDs set the grain at which you want to maintain the balances. There are three different IDs—General Ledger Balance ID, Accounts Payable Key ID, and Accounts Receivable Key ID. The default configurations are set to the most representative grains for maintaining the three different balances.

By default, the General Ledger Balance ID (BALANCE_ID) is configured as follows:

```
Account_number||’~’||Company_code||’~’||Business_area||’~’||Client
```

Therefore, the General Ledger Account, Company Code, Business Area, and Client Code maintain the General Ledger Balances. You use the Client Code to distinguish between different instances of SAP R/3. For example, if you are running one instance of SAP R/3 for your U.S. business, and another instance of SAP R/3 for your Japan business, you may have the same General Ledger Account numbers in each system referring to different accounts. To distinguish the same General Ledger Account numbers in different instances, the grain of the balance includes the Client Code.

To change the grain at which you accumulate the General Ledger balance, modify the Balance ID or Key ID definition in the Expression transformation in the applicable mapping. Note that there are two sets of mappings—the balance extract mapping and the initial fact load mapping. The extract moves the balance from the source to staging tables, and the initial fact load mappings move the data from staging tables to the data warehouse. The following procedure provides instructions on how to configure the balance extract.

To configure the balance extract for the General Ledger Balance

1. In PowerCenter Designer, open the Configuration for SAP R/3 folder.
2. In Mapplet Designer, open the `M_S_GL_XACTS_EXTRACT_SPLIT` mapping.
3. Double-click the Expression transformation to open the Edit Transformations window.
4. Edit `GL_BALANCE_ID`.
5. Validate and save your changes to the repository.
6. Repeat Step 2 to Step 5 for the following mappings, if you are configuring Siebel Enterprise Sales Analytics with Siebel Financial Analytics, and the sales and purchase subledger post to the General Ledger at the header level:

   - `M_S_GL_XACTS_HDR_SALES_IVCLNS_DERIVE`
   - `M_S_GL_XACTS_HDR_SALES_ORDLNS_DERIVE`
   - `MS_GL_XACTS_OTHERS_DERIVE`

   Repeat Step 2 to Step 5 for the following mappings, if you are configuring Siebel Enterprise Sales Analytics with Siebel Financial Analytics, and the sales and purchase subledger post to the General Ledger at the detail level:

   - `M_S_GL_XACTS_DET_SALES_IVCLNS_KEYS_DERIVE`
   - `M_S_GL_XACTS_DET_SALES_ORDLNS_KEYS_DERIVE`
   - `M_S_GL_XACTS_OTHERS_DERIVE`

Configuring the Siebel Payables Analytics Balance Extract

This task is a step in the Process of Configuring Siebel Financial Analytics for SAP R/3 on page 329.

The Balance and Key IDs set the grain at which you want to maintain a balance. There are three different IDs—General Ledger Balance ID, Accounts Payable Key ID, and Accounts Receivable Key ID. You set the default configurations to the most-representative grains for maintaining the three different balances.

By default, the Accounts Payable AP Key ID (KEY_ID) is configured as follows:

```
Vendor__creditor__account_number||’~’||Company_code||’~’||Client
```

Therefore, you use the vendor or creditor account number, company code, and client code to maintain the Accounts Payable Balance. You use the client code to distinguish between different instances of SAP R/3. For example, if you are running one instance of SAP R/3 for your U.S. business, and another instance of SAP R/3 in your Japan business, you may have the same AP account numbers in each system, which refer to different accounts. To distinguish the same GL account numbers in different instances, you set the grain of the balance to include the client code.

To configure the balance extract for the AP Balance

1. In PowerCenter Designer, open the Configuration for SAP R/3 folder.
2. In Mapplet Designer, open the M_S_GL_XACTS_EXTRACT_SPLIT mapping.
3. Double-click the Expression transformation to open the Edit Transformations window.
4. Edit AP_BALANCE_ID.
5. Validate and save your changes to the repository.
6. Repeat Step 2 to Step 5 for the following mappings, if you are configuring Siebel Enterprise Sales Analytics with Siebel Financial Analytics, and the sales and purchase subledger post to the General Ledger at the header level:

   - M_S_AP_XACTS_HDR_SALES_IVCLNS_DERIVE
   - M_S_AP_XACTS_HDR_SALES_ORDLNS_DERIVE
   - MS_AP_XACTS_OTHERS_DERIVE

   Repeat Step 2 to Step 5 for the following mappings, if you are configuring Siebel Enterprise Sales Analytics with Siebel Financial Analytics, and the sales and purchase subledger post to the General Ledger at the detail level:

   - M_S_AP_XACTS_DET_SALES_IVCLNS_KEYS_DERIVE
   - M_S_AP_XACTS_DET_SALES_ORDLNS_KEYS_DERIVE
   - M_S_AP_XACTS_OTHERS_DERIVE

Configuring the Customer Costs Lines and Product Costs Lines Tables for Siebel Profitability Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for SAP R/3 on page 329.

In Siebel Profitability Analytics, the Customer Costs Lines and Product Costs Lines fact tables store the costing and expenses for the Profitability functional area. You need to use these tables with General Ledger Revenue and General Ledger COGS fact tables.

The General Ledger Revenue and General Ledger COGS fact tables are populated by the or SAP R/3 source systems, but the Customer Costs Lines and Product Costs Lines fact tables are populated by the universal source system.

For more information on the Customer Costs Lines and Product Costs Lines fact tables, see About the Customer Costs Lines and Product Costs Lines Tables for Siebel Profitability Analytics on page 324.

To load the Customer Costs Lines and Product Costs Lines tables

1. Open the file_cust_costlns.csv file in the $pmserver\srcfiles folder.

2 Insert a record into the file for each customer costing transaction you want to load into the Customer Cost fact table.

   For more information on how to populate each field in the CSV file, please see the `Cust_Costlns_Interface_description.xls` file.

   For the SOURCE_ID column, you need to provide the same source identifier value as in the `file_parameters_sapr3.csv` file.

3 Save and close the file.

4 Repeat Step 1 to Step 3 for the `file_prod_costlns.csv` file.

5 Open Workflow Manager, open the Configuration for Universal Source folder, and run the Universal_Finance_Profitability workflow.

   NOTE: By default, the Universal_Finance_Profitability workflow loads the Transaction Type dimension and the Customer Costs and Product Costs fact tables. If you need to load other dimensions, open up the Extract_Dimensions and Load_Dimensions worklets and enable the corresponding dimension extract and load sessions.

Configuring the Siebel Receivables Analytics Balance Extract

This task is a step in the Process of Configuring Siebel Financial Analytics for SAP R/3 on page 329.

By default, the Accounts Receivable (AR) Key ID (KEY_ID) is configured as follows:

```
Customer_number||'~'||Company_code||'~'||Client
```

Therefore, the Accounts Receivable balance is maintained by customer number, company code, and client code. The client code is used to distinguish between different instances of SAP R/3. For example, if you are running one instance of SAP R/3 for your U.S. business, and another instance of SAP R/3 in your Japan business, you may have the same AR account numbers that refer to different accounts in each system. To distinguish the same GL account numbers in different instances the grain of the balance is set to include the client code.

To change the grain at which you accumulate the balances, modify the Key ID definition in the Expression transformation in the applicable mapping. Note that there are two sets of mappings—the balance extract mappings and the initial fact load mappings. The extract provides the balance from the source to staging tables. The initial fact load mappings move the data from staging tables to the data warehouse.

To configure the balance extract for the AR Balance

1 In PowerCenter Designer, open the Configuration for SAP R/3 folder.

2 In Mapplet Designer, open the `M_S_GL_XACTS_EXTRACT_SPLIT` mapping.

3 Double-click the Expression transformation to open the Edit Transformations window.

4 Edit Key_ID.

Make sure the AR Key ID in the extract mapping is the same precision as the AR Balance ID in the fact load mapping.

5 Validate and save your changes to the repository.

6 Repeat Step 2 to Step 5 for the following mappings, if you are configuring Siebel Enterprise Sales Analytics with Siebel Financial Analytics, and the sales and purchase subledger post to the General Ledger at the header level:

- M_S_AR_XACTS_HDR_SALES_IVCLNS_DERIVE
- M_S_AR_XACTS_HDR_SALES_ORDLNS_DERIVE
- MS_AR_XACTS_OTHERS_DERIVE

Repeat Step 2 to Step 5 for the following mappings, if you are configuring Siebel Enterprise Sales Analytics with Siebel Financial Analytics, and the sales and purchase subledger post to the General Ledger at the detail level:

- M_S_AR_XACTS_DET_SALES_IVCLNS_KEYS_DERIVE
- M_S_AR_XACTS_DET_SALES_ORDLNS_KEYS_DERIVE
- M_S_AR_XACTS_OTHERS_DERIVE

About PeopleSoft Trees in Siebel Financial Analytics

For PeopleSoft, the Siebel Financial Analytics application sources data from a data structure, called PeopleSoft Trees, to get information about the organization's General Ledger hierarchies, Profit hierarchies, Cost Centers’ hierarchies, and so on.

PeopleSoft Trees are a flexible, generic way of constructing hierarchical summarization of a particular database fields in PeopleSoft for reporting purposes. Typically, entities such as Chart of Account fields (Account, Dept, and Project, and so on), items, locations, and so on, are organized into user-defined trees.

Table 68 lists the PeopeSoft Trees the Siebel Financial Analytics application sources.

Table 68. PeopeSoft Trees for Siebel Financial Analytics

PeopleSoft Data Structure	Repository Source Name	Description
PeopleSoft Tree	ACCTROLLUP	Account Rollup
PeopleSoft Tree	AR_BUSINESS_UNITS	AR Business Unit Roll-Up
PeopleSoft Tree	DEPARTMENTS	Manufacturing Departments
PeopleSoft Tree	OPERUNIT	Operating Unit

Table 69 lists the mappings and sessions using PeopleSoft Trees.

Table 69. List of Mappings And Sessions Using PeopleSoft Trees

Repository Source Name	Mapping	Session
ACCTROLLUP	M_P_GL_ACCTS_HIERARCHY_EXTRACT	S_M_P_GL_ACCTS_HIERARCHY_EXTRACT
AR_BUSINESS_UNITS	M_P_BUSN_ORGS_COMPANY_HIERARCHY_EXTRACT	S_M_P_BUSN_ORGS_COMPANY_HIERARCHY_EXTRACT
DEPARTMENTS	M_P_PROFIT_CNTRS_HIERARCHY_EXTRACT	S_M_P_PROFIT_CNTRS_HIERARCHY_EXTRACT
OPERUNIT	M_P_COST_CENTERS_HIERARCHY_EXTRACT	S_M_P_COST_CENTERS_HIERARCHY_EXTRACT

Process of Configuring Siebel Financial Analytics for PeopleSoft 8.4

To configure Siebel Financial Analytics for PeopleSoft 8.4, perform the following tasks:

- Customizing the PeopleSoft Tree Names on page 340
- Importing PeopleSoft Trees Into the PowerCenter Repository on page 341
- Configuring the Group Account Number Categorization for Siebel General Ledger Analytics on page 342
- Configuring the Primary Ledger Name for Siebel General Ledger Analytics on page 343
- Configuring the Primary Ledger Name for Siebel Payables Analytics on page 343
- Configuring the Primary Ledger Name for Siebel Profitability Analytics on page 344

Customizing the PeopleSoft Tree Names

This task is a step in the Process of Configuring Siebel Financial Analytics for PeopleSoft 8.4 on page 340.

For a PeopleSoft environment with different tree names, you need to import these into the PowerCenter repository, and replace the old tree names with the new tree names wherever they are used in the mappings.

For example, if you are storing your Operating Unit information in a tree called BD_BUSUNIT, then you would need to:

1 Import the BD_BUSUNIT tree into the PowerCenter repository. For more information on importing a PeopleSoft tree into the PowerCenter repository, see Importing PeopleSoft Trees Into the PowerCenter Repository on page 341.

2. Edit the mapping that uses the OPERUNIT tree. Table 69 on page 340 lists the mappings for PeopleSoft Trees.
3. Replace the source definition in the mapping with BD_BUSUNIT, connect the columns appropriately, and validate and save the mapping.
4. Edit the corresponding session, and validate and save the session. Table 69 on page 340 lists the sessions for PeopleSoft Trees.

Importing PeopleSoft Trees Into the PowerCenter Repository

This task is a step in the Process of Configuring Siebel Financial Analytics for PeopleSoft 8.4 on page 340.

The Designer displays the following tree information in the Import From PeopleSoft dialog box, to identify the tree you need to import:

- **SetID.** If a tree is a SetID, it appears before the tree name.
- **Tree Name.** The name of the tree.
- **Effective Date.** The tree effective date appears after the tree name.

PeopleSoft uses the SetID and the Effective Date to identify trees. When importing a tree from PeopleSoft, you can use the SetID and the Effective Date to select the tree. The SetID and the Effective Date are displayed in the source definition in the Source Analyzer.

You can import strict-level trees from the Trees tab in the Import From PeopleSoft dialog box. Detail and Summary trees appear in the Trees folder, and Winter trees appear in the Winter Trees folder.

To import PeopleSoft Tree into the PowerCenter repository

1. Open Designer, and connect to your repository.
2. Open the Configuration for PeopleSoft 8.4 folder.
3. On the Sources menu, click Import from PeopleSoft.
4. Connect to the PeopleSoft OLTP.

   The Import From PeopleSoft dialog box appears.
5. Click the Trees tab
6. Click the Tree you want to import.
7. Click OK.

Configuring the Group Account Number Categorization for Siebel General Ledger Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for PeopleSoft 8.4 on page 340.

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 *Siebel Customer-Centric Enterprise Warehouse Data Model Reference*.

The mappings to General Ledger Accounts Numbers are important for both Profitability Analysis (Income Statement) and General Ledger accounts.

The Group Account Number categorizes each Siebel General Ledger Analytics account record in the IA_GL_ACCOUNTS dimension table. Each Siebel General Ledger Analytics account is assigned a Group Account Number. The logic for assigning the accounts is located in the `file_fin_stmt_item_codes_psft.csv` file. For example, this file might have the layout shown in Table 70.

In Table 70, all accounts within the account number range from 1000 to 139800 containing Business Unit equal to 1 are assigned to Account Depreciation (ACCN DEPCN). Similarly, looking at the second row in the table, all accounts within the account number range from 140000 to 141099 containing Business Unit equal to 1 are assigned to AR. Each row categorizes all accounts within the specified account number range and with the given combination of Hierarchy Code and Company Code.

Table 70. Sample Layout of file_fin_stmt_item_codes_psft.csv File

Business Unit	FROM_ACCT	TO_ACCT	GROUP_ACCT_NUM
1	1000	139800	ACC DEPCN
1	140000	141099	AR
1	141100	142400	CA
1	143000	143000	AP
1	143100	143100	AR
1	143110	143200	CASH

NOTE: You must capitalize the letters for the GROUP_ACCT_NUM column.

To assign accounts to Group Account Numbers

1. Open the `file_fin_stmt_item_codes_psft.csv` file in the `$pmserver\srcfiles` folder.
2. Remove all entries in the file and insert your account ranges for each Group Account Number.

   For list of values for Group Account Numbers, please see the MASTER_CODE column in the `file_fin_stmt_item_codes_psft.csv` file.

3. Save and close the file.

Configuring the Primary Ledger Name for Siebel General Ledger Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for PeopleSoft 8.4 on page 340.

The Primary Ledger is an accounting ledger in which you prepare your financial statements for your corporate headquarters.

By default, the name of the Primary Ledger is set to LOCAL for PeopleSoft. However, if the name of your Primary Ledger is not LOCAL, you can change this value by modifying the `file_parameters_psft84.csv` file. For example, you could have a different reporting ledger for a report of your financial statements in a different currency.

To modify the Primary Ledger name

1. Open the `file_parameters_psft84.csv` file using Notepad in the `$pmserver\srcfiles` folder.
2. Modify the Primary Ledger name as necessary.

   For example, if your Primary Ledger's name is MYPRIMARY, replace all instances of the default value S,LOCAL, with S,MYPRIMARY,.
3. Save and close the file.

Configuring the Primary Ledger Name for Siebel Payables Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for PeopleSoft 8.4 on page 340.

The Primary Ledger is an accounting ledger in which you prepare your financial statements for your corporate headquarters.

By default, the name of the Primary Ledger is set to LOCAL for PeopleSoft. However, if the name of your Primary Ledger is not LOCAL, you can change this value by modifying the `file_parameters_psft84.csv` file. For example, you could have a different reporting ledger for a report of your financial statements in a different currency.

To modify the Primary Ledger name

1. Open the `file_parameters_psft84.csv` file using Notepad in the `$pmserver\srcfiles` folder.
2. Modify the Primary Ledger name as necessary.

   For example, if your Primary Ledger's name is MYPRIMARY, replace all instances of the default value S,LOCAL, with S,MYPRIMARY,.

3 Save and close the file.

Configuring the Primary Ledger Name for Siebel Receivables Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for PeopleSoft 8.4 on page 340.

The Primary Ledger is an accounting ledger in which you prepare your financial statements for your corporate headquarters.

By default, the name of the Primary Ledger is set to LOCAL for PeopleSoft. However, if the name of your Primary Ledger is not LOCAL, you can change this value by modifying the `file_parameters_psft84.csv` file. For example, you could have a different reporting ledger for a report of your financial statements in a different currency.

To modify the Primary Ledger name

1. Open the `file_parameters_psft84.csv` file using Notepad in the `$pmserver\srcfiles` folder.
2. Modify the Primary Ledger name as necessary.

   For example, if your Primary Ledger's name is MYPRIMARY, replace all instances of the default value S,LOCAL, with S,MYPRIMARY,.
3. Save and close the file.

Configuring the Primary Ledger Name for Siebel Profitability Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for PeopleSoft 8.4 on page 340.

The Primary Ledger is an accounting ledger in which you prepare your financial statements for your corporate headquarters.

By default, the name of the Primary Ledger is set to LOCAL for PeopleSoft. However, if the name of your Primary Ledger is not LOCAL, you can change this value by modifying the `file_parameters_psft84.csv` file. For example, you could have a different reporting ledger for a report of your financial statements in a different currency.

To modify the Primary Ledger name

1. Open the `file_parameters_psft84.csv` file using Notepad in the `$pmserver\srcfiles` folder.
2. Modify the Primary Ledger name as necessary.

   For example, if your Primary Ledger's name is MYPRIMARY, replace all instances of the default value S,LOCAL, with S,MYPRIMARY,.

3 Save and close the file.

Process of Configuring Siebel Financial Analytics for Post-Load Processing

To configure Siebel Financial Analytics for post-load processing, perform the following tasks:

- Configuring Aging Buckets for Siebel Receivables Analytics on page 345
- Configuring the History Period for the Invoice Level for Siebel Receivables Analytics on page 346
- Configuring Aging Buckets for Siebel Payables Analytics on page 346
- Configuring the History Period for the Invoice Level for Siebel Payables Analytics on page 347

Configuring Aging Buckets for Siebel Receivables Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for Post-Load Processing on page 345.

The receivable aging information shows the open, due, and overdue amounts. It is split into four time intervals or buckets. You need to configure the values for the first three bucket start and bucket end days. The fourth bucket includes transactions outside the range of the first three buckets. The following procedure describes how to configure the bucketing range.

To configure the receivable aging bucket

1 Navigate to the $pmserver\srcfiles folder.

2 Open the sess_param_plp.par file using Notepad.

3 Modify the bucket start and the bucket end parameter values for the S_M_PLP_AR_SNAPSHOT_AGING_INV session.

The following table lists the default values for the S_M_PLP_AR_SNAPSHOT_AGING_INV session.

[S_M_PLP_AR_SNAPSHOT_AGING_INV]
$$BUCKET1_END=30
$$BUCKET1_START=0
$$BUCKET2_END=60
$$BUCKET2_START=31
$$BUCKET3_END=90
$$BUCKET3_START=61

4 Save and close the sess_param_plp.par file.

Configuring the History Period for the Invoice Level for Siebel Receivables Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for Post-Load Processing on page 345.

The Siebel Customer-Centric Enterprise Warehouse can store the history of the Invoice level aging information. The history period value is preconfigured with 2. You configure this value to match your business requirements. The following procedure describes how to configure the history period for the Invoice level.

To configure the history period for the Invoice level

1 Navigate to the $pmserver\srcfiles folder.

2 Open the sess_param_plp.par file using Notepad.

3 Modify the $$HISTORY_MONTH value in the following section:

```
[S_M_PLP_AR_SNAPSHOT_AGING_INV]

<Set of Bucketing parameters>

$$HISTORY_MONTHS=2
```

4 Save and close the sess_param_plp.par file.

Configuring Aging Buckets for Siebel Payables Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for Post-Load Processing on page 345.

The payable aging information shows the open, due, and overdue amounts. It is split into four time intervals or buckets. You need to configure the values for the first three bucket start and bucket end days. The fourth bucket includes transactions outside the range of the first three buckets. The following procedure describes how to configure the bucketing range.

To configure the payables aging bucket

1 Navigate to the $pmserver\srcfiles folder.

2 Open the sess_param_plp.par. file using Notepad.

3 Modify the bucket start and the bucket end parameter values for the S_M_PLP_AP_SNAPSHOT_AGING_INV session.

The following table lists the default values for the S_M_PLP_AP_SNAPSHOT_AGING_INV session.

[S_M_PLP_AP_SNAPSHOT_AGING_INV]
$$BUCKET1_END=30
$$BUCKET1_START=0
$$BUCKET2_END=60
$$BUCKET2_START=31
$$BUCKET3_END=90
$$BUCKET3_START=61

4 Save and close the sess_param_plp.par file.

Configuring the History Period for the Invoice Level for Siebel Payables Analytics

This task is a step in the Process of Configuring Siebel Financial Analytics for Post-Load Processing on page 345.

The Siebel Customer-Centric Enterprise Warehouse can store the history of the Invoice level aging information. The history period value is preconfigured with 2. You configure this value to match your business requirements. The following procedure describes how to configure the history period for the Invoice level.

To configure the history period for the invoice level

1 Navigate to the $pmserver\srcfiles folder.

2 Open the sess_param_plp.par. file using Notepad.

3 Modify the $$HISTORY_MONTH value in the following section:

```
[S_M_PLP_AP_SNAPSHOT_AGING_INV]

<Set of Bucketing parameters>

$$HISTORY_MONTHS=2
```

4 Save and close the sess_param_plp.par file.

17 Configuring Siebel Strategic Sourcing Analytics

This chapter describes how to configure certain objects for particular sources to meet your business needs.

It contains the following topics:

Overview of Siebel Strategic Sourcing Analytics

The Siebel Strategic Sourcing Analytics application allows you to analyze your procurement expenses and costs variances, as well as track your suppliers and supplier-related spending. It helps you to understand where and how spending occurs across your entire organization and to evaluate the price, quality, and delivery timing in procuring materials. Furthermore, it allows you to evaluate both direct and indirect spending (indirect spend being MRO and employee expenses). Through visibility into this complex data, you can reduce spending, intelligently select suppliers, and achieve a new balance and flexibility in your sourcing.

The Siebel Strategic Sourcing Analytics application is comprised of these functional areas:

- **Expenses.** The Expenses functional 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.
- **Spend.** The Spend functional area contains targeted reports and metrics that allow you to analyze both direct and indirect spend in detail to allow complete visibility of spending across your organization.
- **Suppliers.** 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.

Process of Configuring Siebel Strategic Sourcing Analytics for Oracle 11i

This section contains Siebel Strategic Sourcing Analytics configuration points that are specific to Oracle 11i. It contains the following topics:

Configuring the Handling of Currency Types

This task is a step in the Process of Configuring Siebel Strategic Sourcing Analytics for Oracle 11i on page 350.

The Siebel Strategic Sourcing Analytics application, like all other applications, uses the same method for handling currency conversions from document currency to local and group currencies. This guide provides a functional overview of how each of the local and group currencies is derived depending on what is supplied to the Siebel Customer-Centric Enterprise Warehouse. For more information on how to configure various components that relate to local, document, and group currencies see Chapter 8, “Configuring Common Components of the Siebel Customer-Centric Enterprise Warehouse.”

Configuring the Region Name

This task is a step in the Process of Configuring Siebel Strategic Sourcing Analytics for Oracle 11i on page 350.

This configuration point applies only to Plant locations, Storage locations, as well as Supplier locations. By default, the Region Name column (EXT_REGION_NAME) is populated using the same code value as the Region Code column (EXT_REGION_CODE). However, you can redefine the load mapping’s Source Adapter mapplet to load a source-supplied region name instead of the code. If you want to reconfigure the load in this manner, you can load the region code and region name into the IA_CODES table. For information on loading codes and code names into the IA_CODES table, see Chapter 8, “Configuring Common Components of the Siebel Customer-Centric Enterprise Warehouse.”

After you load the region code and region name into the IA_CODES table, you can remove the expression in the Source Adapter mapplet that defines the Region Name column. If you leave the Region Name's expression blank, the ADI looks up the Region Name in the IA_CODES table, using the supplied region code when the load occurs. The load mapping then inserts the region name and region code into the data warehouse table.

To configure the Region Name definition

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. Open the mapplet you want to edit.

   The following is a list of all Source Adapter mapplets that use the EXT_REGION_NAME column:

   - MPLT_SAI_SUPPLIERS
   - MPLT_SAI_BUSN_LOCS_PLANT
   - MPLT_SAI_BUSN_LOCS_STORAGE_LOC
3. Double-click the Expression transformation to open the Edit Transformations box, and select the Port tab to display the EXT_REGION_NAME port.
4. Edit the condition by removing the assigned value, if you want the lookup to occur.
5. Click Apply to save changes.
6. Validate the mapplet, and click OK to exit.

Configuring the State Name

This task is a step in the Process of Configuring Siebel Strategic Sourcing Analytics for Oracle 11i on page 350.

By default, the State Name column (EXT_STATE_NAME) is populated using the same code value as the State Code column (EXT_STATE_CODE). However, you can redefine the load mapping's Source Adapter mapplet to load a source-supplied state name instead of the code. If you want to reconfigure the load in this manner, you can load the state code and state name into the IA_CODES table. For information on loading codes and code names into the IA_CODES table, see Chapter 8, "Configuring Common Components of the Siebel Customer-Centric Enterprise Warehouse."

After you load the state code and state name into the IA_CODES table, you can remove the expression in the Source Adapter mapplet that defines the State Name column. If you set the State Name's expression to null, the ADI looks up the State Name in the IA_CODES table using the supplied state code, when the load occurs. The load mapping then inserts the state name and state code into the data warehouse table.

To configure the State Name definition

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. Open the MPLT_SAI_SUPPLIERS mapplet.

3. Double-click the Expression transformation to open the Edit Transformations box, and select the Port tab to display the EXT_STATE_NAME port.
4. Edit the condition by removing the assigned value, if you want the lookup to occur.
5. Click Apply to save changes.
6. Validate the mapplet, and click OK to exit.

Configuring the Country Name

This task is a step in the Process of Configuring Siebel Strategic Sourcing Analytics for Oracle 11i on page 350.

By default, the Country Name column (EXT_COUNTRY_NAME) is populated using the same code value as the Country Code column (EXT_COUNTRY_CODE). However, you can redefine the load mapping's Source Adapter mapplet to load a source-supplied country name instead of the code. If you want to reconfigure the load in this manner, you can load the country code and country name into the IA_CODES table. For information on loading codes and code names into the IA_CODES table, see Chapter 8, "Configuring Common Components of the Siebel Customer-Centric Enterprise Warehouse."

After you load the country code and country name into the IA_CODES table, you can remove the expression in the Source Adapter mapplet that defines the Country Name column. If you set the Country Name's expression to null, when the load occurs, the ADI looks up the country name in the IA_CODES table, using the supplied country code. The load mapping then inserts the country name and country code into the data warehouse table.

To configure the Country Name definition

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. Open the MPLT_SAI_SUPPLIERS mapplet.
3. Double-click the Expression transformation to open the Edit Transformations box, and select the Port tab to display the EXT_COUNTRY_NAME port.
4. Edit the condition by removing the assigned value, if you want the lookup to occur.
5. Click Apply to save changes.
6. Validate the mapplet, and click OK to exit.

Configuring the Make-Buy Indicator

This task is a step in the Process of Configuring Siebel Strategic Sourcing Analytics for Oracle 11i on page 350.

This configuration also applies to the Spend and Suppliers functional areas.

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 make (M). However, if the code is set to 2, then the indicator is set to B. 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 MPLT_SAI_PRODUCTS mapplet. 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 PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. Open the MPLT_SAI_PRODUCTS mapplet.
3. Double-click the Expression transformation to open the Edit Transformations box, and select the Port tab to display the EXT_MAKE_BUY_IND port.
4. Edit the condition by replacing the prepackaged condition with your desired logic.
5. Click Apply to save changes.
6. Validate the mapplet, and click OK to exit.

Extracting Particular Purchase Order Records

This task is a step in the Process of Configuring Siebel Strategic Sourcing Analytics for Oracle 11i on page 350.

This configuration also applies to the Spend functional area.

You may not want to extract particular types of records from purchase orders in your source system. In these cases, you can modify the filter condition in the Source Qualifier of the mapplet. By default, the filter condition is set to PLANNED, BLANKET, or STANDARD. However, you can change this value to some conditional statement that only allows particular types of records to be extracted.

To extract particular types of purchase order records

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. Open the MPLT_BCI_PURCH_ORDERS mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations box, and select the Properties tab to display the SQL Query.
4. Double-click the value in the SQL Query to open the SQL Editor box and edit the statement.
5. Replace the prepackaged filter condition with the new filter statement that reflects your business needs.
6. Edit the WHERE clause of the statement.
7. Click Apply to save the changes, and click OK to exit.

8. Validate the expression, and save your changes to the repository.
9. Repeat Step 2 to Step 8 for the MPLT_BCI_PURCH_SCHLNS mapplet.

Configuring the Purchase Organization Hierarchy

This task is a step in the Process of Configuring Siebel Strategic Sourcing Analytics for Oracle 11i on page 350.

This configuration also applies to the Spend functional area.

The Purchase Organization hierarchy can contain 10 different levels. Each level is denoted by a number, where 1 is the top of the hierarchy and 10 is the bottom of the hierarchy. By default, the first three levels of the Purchase Organization hierarchy are set as follows:

- `EXT_BORG_HIER1_CODE: Organization ID`
- `EXT_BORG_HIER2_CODE: Legal Entity ID`
- `EXT_BORG_HIER3_CODE: Set of Books ID`

The remainder of the hierarchy ports (`EXT_BORG_HIERX_CODE`) are populated using whatever values are input into the `INP_BORG_HIERX_CODE` ports (X denotes the level within the hierarchy).

If you want to configure your hierarchy differently than what is prepackaged, you must modify the `EXT_BORG_HIERX_CODE` ports in the Source Adapter mapplet.

To modify the Purchase Organization hierarchy

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. Open the MPLT_SAI_BUSN_ORGS mapplet.
3. Edit the expression for the `EXT_BORG_HIERX_CODE` port.
4. Validate the expression, and save your changes to the repository.

If applicable, you may need to modify the hierarchy in the front end so that end users can use the hierarchy levels for queries and reports.

Configuring the Siebel Business Analytics Repository for Siebel Strategic Sourcing Analytics

This task is a step in the Process of Configuring Siebel Strategic Sourcing Analytics for Oracle 11i on page 350.

A General Ledger account can contain cost center information. However, the Cost Center dimension table is not loaded for Oracle 11i.

The department segment of the General Ledger account is commonly used as a cost center. You can map this segment as a cost center in the Siebel Business Analytics Repository. Siebel Customer-Centric Enterprise Warehouse is preconfigured to use the `IA_GL_ACCOUNT.ACCT_SEG2_NAME` column as the cost center name and the `IA_GL_ACCOUNT.ACCT_SEG2_CODE` column as the cost center number.

To configure the Repository for Siebel Strategic Sourcing Analytics for Oracle 11i

1. Open the SiebelBusinessAnalytics.rpd in the `$SAHome\SiebelAnalytics\Repository` folder.
2. In the Mapping pane, expand Dim - Cost Centers, and expand Sources.
3. Double-click on Dim_IA_GL_ACCOUNTS_CostCenter_Segment.
4. In Column Mappings, change the expression column names to the appropriate segment.
5. Double-click on Dim_IA_COST_CENTERS.
6. Click the General tab and clear the Active check box.
7. Save the repository.
8. Click Yes to Check Global Consistency, and click OK when the Warnings are displayed.

Process of Configuring Siebel Strategic Sourcing Analytics for SAP R/3

This section contains Siebel Strategic Sourcing Analytics configuration points that are specific to SAP R/3. It contains the following topics:

- Configuring the Siebel Business Analytics Repository for SAP R/3 on page 355
- Configuring the Date Parameters for the SAP R/3 Parameter File on page 356

Configuring the Siebel Business Analytics Repository for SAP R/3

This task is a step in the Process of Configuring Siebel Strategic Sourcing Analytics for SAP R/3 on page 355.

The Requisition Cost and Purchase Cost fact tables are not loaded for SAP R/3. You need to disable these tables in the Siebel Business Analytics Repository.

To configure the repository for Siebel Strategic Sourcing Analytics for SAP R/3

1. Open the SiebelBusinessAnalytics.rpd in the `$SAHome\SiebelAnalytics\Repository` folder.
2. In the Mapping pane, expand Fact - Purchase Requisitions, and expand Sources.
3. Double-click on Fact_IA_RQLNS_COSTS, click the General tab, and clear the Active check box.
4. In the Mapping pane, expand Fact - Purchase Costs, and expand Sources.

5. Double-click on Fact_IA_PURCH_COSTS, click the General tab, and clear the Active check box.
6. In the Mapping pane, expand Dim - Cost Centers, and expand Sources.
7. Double-click on Dim_IA_GL_ACCOUNTS_CostCenter_Segment, click the General tab, and clear the Active check box.
8. Save the repository.
9. Click Yes to Check Global Consistency, and click OK when the Warnings are displayed.

Configuring the Date Parameters for the SAP R/3 Parameter File

This task is a step in the Process of Configuring Siebel Strategic Sourcing Analytics for SAP R/3 on page 355.

You need to set the PARM_NVALUE_1 value in the `file_parameters_sapr3.csv` file to the number of days that you expect your orders to be open. This configuration is necessary for ETL as SAP R/3 does not update the last changed date for a table when a user updates that table.

The PARM_NVALUE_1 value is used by the following session parameter codes:

- S_M_S_PURCH_ORDERS_EXTRACT:INCRDATE
- S_M_S_PURCH_RCPTS_EXTRACT:INCRDATE
- S_M_S_PURCH_RQLNS_EXTRACT:INCRDATE
- S_M_S_PURCH_SCHLNS_EXTRACT:INCRDATE

NOTE: There are always orders that are open for a long period of time. To make sure that ETL captures changes to these orders, it is recommended that you occasionally set the PARM_NVALUE_1 value to a value equivalent to that period of time.

To configure the date parameters for the SAP R/3 parameter file

1. Open the file_parameters_sapr3.csv file using Microsoft WordPad or Notepad in the `$pmserver\SrcFiles` folder.
2. Search for the column labeled PARM_NVALUE_1.
3. Change the default date value of parameter code to the date value you require.
4. Save your file.

About Configuring Siebel Strategic Sourcing Analytics for Universal Source

Expenses has one fact table (IA_EXPENSE) and one fact aggregate table (IA_EXPENSE_A1) that support metrics and reports for examining employee expenses. Several mappings populate these tables to complete extracts, loads and updates; you may configure these to suit your organization's business rules. The following sections discuss decisions you must make before you begin adapting individual PowerCenter objects, and provide specific configuration procedures for the universal source.

Universal Source Adapter mapplets extract data from a flat file interface to populate the Siebel Customer-Centric Enterprise Warehouse. In this phase of your project, you can configure the following:

- **System Flags and Indicators.** You may configure various system flags to indicate record rejection settings, as well as to indicate if your employees are using your preferred vendors, if you can forward expenses to your customers, and if receipts are available for expensed items.
- **Currency and Payment Options.** You may configure the date used to establish your exchange rates, determine if you allow expenses to be distributed across multiple cost centers, and define payment types in your data warehouse.

Before you begin, you must make the following decisions:

- **Cash Advances.** Cash advance records have a unique expense item number. If your system allows multiple cash advance records for one expense report, each of these advances must have their own identifiers.
- **Violations.** Many organizations capture violations of company expense policies at the item level (for example, the line item airfare exceeds $2000), cash advance level (for example, cash advance exceeds $500) and at the expense report level (for example, the report's total expenses exceed $5000). Currently the Siebel Customer-Centric Enterprise Warehouse stores item level violations within the corresponding item record, but the cash advance record stores both cash advance and report-level violations. Furthermore, each record has a VIOLATION_KEY that can point to IA_REASONS, where violation details are stored. Depending on how you want your analytic system to perform, you must edit your universal business adapter file to reflect the violation counts and keys appropriately. For example:
  - If a requestor violates a cash advance policy, but there are no other violations at the report level, the VIOLATION_ID refers to the cash advance violation only. The violation count equals the cash advance violation counts.
  - If a requestor violates company policy with their expense report, but has not taken a cash advance, you must add a dummy record in the flat file for a cash advance and set the cash advance amount to zero, and enter the violation count as the total number of expense report violations. In this scenario, VIOLATION_ID refers to the expense report violation data only.
  - If a requestor violates a cash advance policy and an expense report policy, you must total the violation counts and enter them in your flat file record, and the VIOLATION_ID has no value. However, if your organization wants to prioritize the violations and have the VIOLATION_ID point to that which is most important, you may point it to the appropriate entry in IA_REASONS.

- **Maintaining aggregate Information.** If you plan to run the ETL process that populates IA_EXPENSES at a different frequency than the process that populates IA_EXPENSES_A1, you must build additional mappings to retain incremental data in the aggregate table NU_EXPENSES. This is used as a source during post-load processing, required for updating invoice records. See the discussion on setting the S_M_PLP_EXPENSES_INVOICE_UPD_ALT session to run when the incremental invoice load frequency differs for the expense functional area, in Domain Values and CSV Worksheet Files for Siebel Strategic Sourcing Analytics on page 375.
- **Deciding to Run post-load processing Mappings.** Depending on your organization's needs and source system setup, you may not need to run the post-load process mappings packaged with Expenses:
  - If your source system or flat file is set up to supply credit invoice details (invoice number, invoice date, posted_on date, and so on.) with all other information, then the required fields are populated with corresponding invoice values. As a result, you need not schedule the following PLP sessions—S_M_PLP_EXPENSES_INVOICE_DERIVE, S_M_PLP_EXPENSES_INVOICE_UPD, and S_M_PLP_EXPENSES_INVOICE_UPD_ALT.
  - If your system or file does not supply invoice details of an expense report, you must use the PLP mappings and sessions as discussed in the previous bullet point. However, you only need to run S_M_PLP_EXPENSES_INVOICE_UPD or S_M_PLP_EXPENSES_INVOICE_UPD_ALT. The alternative mapping is provided if you decide to update your existing expense records in the Siebel Customer-Centric Enterprise Warehouse at a different frequency level than you load IA_EXPENSE and IA_EXPENSE_A1.

Process of Configuring Siebel Strategic Sourcing Analytics for a Universal Source

Universal business adapters are built to source data from a flat file interface and populate the data warehouse base tables.

To configure Siebel Strategic Sourcing Analytics for a universal source, perform the following tasks:

- Configuring the Preferred Merchant Flag on page 359
- Configuring the Customer Billable Indicator on page 359
- Configuring the Receipts Indicator on page 360
- Configuring Expense Payment Types on page 360
- Configuring the Default Expense Distribution Percentage on page 361
- Configuring Lookup Dates for Currency Conversion on page 361
- Configuring the Siebel Business Analytics Repository for Siebel Strategic Sourcing Analytics on page 362

Configuring the Preferred Merchant Flag

This task is a step in the Process of Configuring Siebel Strategic Sourcing Analytics for a Universal Source on page 358.

The Siebel Customer-Centric Enterprise Warehouse provides a preferred merchant flag to indicate whether the requestor used a preferred merchant for an expensed item. The flag can have only one value—Y (item acquired from a preferred merchant) or N (item acquired from a merchant not recorded). If you use custom logic to determine merchant status, you must include that logic in the expenses Source Adapter.

To configure the preferred merchant flag

1. In PowerCenter Designer, open the Configuration for Universal Source folder.
2. Open the `MPLT_SAF_EXPENSES` Source Adapter mapplet.
3. Select the Expression transformation to open the Edit Transformations box and select the Port tab.
4. Select the Add Port icon to add the `EXT_PREF_MERCHANT_FLAG_OUT` port.
5. Enter preferred merchant flag logic.
6. Validate the mapplet and click OK to exit.
7. Save your changes to the repository.

Configuring the Customer Billable Indicator

This task is a step in the Process of Configuring Siebel Strategic Sourcing Analytics for a Universal Source on page 358.

The Siebel Business Analytics provides a customer billable indicator that registers whether an expense item is billed to a customer or paid by your organization. The flag can have only one value—Y (cost is passed to the customer) or N (cost is paid by your organization). If you use custom logic to determine customer billable status, you must include that logic in the expenses Source Adapter.

To configure the customer billable indicator

1. In PowerCenter Designer, open the Configuration for Universal Source folder.
2. Open the `MPLT_SAF_EXPENSES` Source Adapter mapplet.
3. Select the Expression transformation to open the Edit Transformations box, and select the Port tab.

   Add a port called `EXT_CUST_BILLABLE_IND_OUT = <insert your expression here>`.
4. Validate the mapplet and click OK to exit.
5. Save your changes to the repository.

Configuring the Receipts Indicator

This task is a step in the Process of Configuring Siebel Strategic Sourcing Analytics for a Universal Source on page 358.

The Siebel Business Analytics provides a receipts indicator that registers whether requestors have submitted a receipt for a line item in their expense report. The flag can have only one value—Y (receipts are available) or N (receipts are not available). If you use custom logic to indicate receipt availability, you must include that logic in the expenses Source Adapter.

To configure the receipts indicator

1. In PowerCenter Designer, open the Configuration for Universal Source folder.
2. Open the MPLT_SAF_EXPENSES Source Adapter mapplet.
3. Select the Expression transformation to open the Edit Transformations box, and select the Port tab.

   Add a port called EXT_RECEIPT_IND_OUT = <insert your expression here>.
4. Validate the mapplet and click OK to exit.
5. Save your changes to the repository.

Configuring Expense Payment Types

This task is a step in the Process of Configuring Siebel Strategic Sourcing Analytics for a Universal Source on page 358.

The Siebel Customer-Centric Enterprise Warehouse supports analysis on three types of payment—Reimbursable Expense (type E), expenses prepaid by your company (type P), and cash advance (type C). All of your organization’s payment types must be mapped to one of these types described earlier; do this by modifying MPLT_SAF_EXPENSES.

To configure additional payment types

1. In PowerCenter Designer, open the Configuration for Universal Source folder.
2. Open the MPLT_SAF_EXPENSES Source Adapter mapplet.
3. Select the Expression transformation to open the Edit Transformations box, and select the Port tab to display the EXT_EXP_PAY_TYPE port.
4. Copy and paste this port, rename it EXT_EXP_PAY_TYPE_IND_OUT.
5. Select O to make your new port an output port.
6. Add a decode logic in the expression to decode source-supplied values to the Siebel Customer-Centric Enterprise Warehouse payment type of your choice.
7. Validate your mapping, and save your changes to the repository.

Configuring the Default Expense Distribution Percentage

This task is a step in the Process of Configuring Siebel Strategic Sourcing Analytics for a Universal Source on page 358.

At times, employee expenses may be distributed across multiple cost centers. For example, technical support associates frequently travel to work in an office with many cost centers; their expenses could be split between those who used their services. This cost center distribution is expected as a percentage from the source system or file; if it is not present a null value is returned. However, this prevents further calculations, so it is preferable to configure the default to be 100% if only one cost center is charged, rather than allow the system to return a null value.

To configure the default expense distribution percentage

1. In PowerCenter Designer, open the Configuration for Universal Source folder.
2. Open the `MPLT_SAF_EXPENSES` Source Adapter mapplet.
3. Select the Expression transformation to open the Edit Transformations box and select the Port tab.
4. Add a port named `EXT_DIST_PERCENTAGE_OUT = <expression that sets this to 100%>`.
5. Validate your mapplet.
6. Save your changes.

Configuring Lookup Dates for Currency Conversion

This task is a step in the Process of Configuring Siebel Strategic Sourcing Analytics for a Universal Source on page 358.

The Siebel Customer-Centric Enterprise Warehouse supports conversion of currency to document (transactional, or source, currency) and group (corporate umbrella currency) for exchange rates. The Siebel Business Analytics uses a specific lookup date to determine the rate of exchange on the date an expense was incurred (`ACTUAL_EXP_DT`). If you decide to use a different date as your currency conversion lookup, you must use the following procedure.

To configure the exchange rate lookup date

1. In PowerCenter Designer, open the Configuration for Universal Source folder.
2. Open the `MPLT_SAF_EXPENSES` Source Adapter mapplet.
3. Select the Expression transformation to open the Edit Transformations box, and select the Port tab to display the `EXT_XRATE_LKP_DATE` port.

   The expression contains the exchange rate lookup date logic.

4. Select the expression in the `EXT_XRATE_LOOKUP_DATE` port to open the Expression Editor box and edit the expression.

5 Edit the lookup date logic by substituting your lookup date for the prepackaged expression.

6 Validate the mapplet and click OK to exit.

7 Save your changes.

Configuring the Siebel Business Analytics Repository for Siebel Strategic Sourcing Analytics

This task is a step in the Process of Configuring Siebel Strategic Sourcing Analytics for a Universal Source on page 358.

Universal source provides the source data for the General Ledger Account (IA_GL_ACCOUNTS) and the Cost Center (IA_COST_CENTER) tables.

The following procedures configure the General Ledger Account and the Cost Center tables for universal source. The second procedure is used to configure a Cost Center for a different General Ledger Account column and when the Cost Center dimension is not populated. You need to identify the correct segment in the General Ledger Account which indicates the cost center information. Siebel Customer-Centric Enterprise Warehouse is preconfigured to use the `IA_GL_ACCOUNT.ACCT_SEG2_NAME` column as the cost center name and the `IA_GL_ACCOUNT.ACCT_SEG2_CODE` column as the cost center number.

To configure the repository when the Cost Center dimension is populated

1 Open the SiebelBusinessAnalytics.rpd in the `$SAHome\SiebelAnalytics\Repository` folder.

2 In the Mapping pane, expand Dim - Cost Centers, and expand Sources.

3 Double-click on DIM_IA_GL_ACCOUNTS_COSTCENTER_SEGMENT, click the General tab, and clear the Active check box.

4 Save the repository.

5 Click Yes to Check Global Consistency, and click OK when the Warnings are displayed.

To configure a Cost Center for a different General Ledger Account column

1 Open the SiebelBusinessAnalytics.rpd in the `$SAHome\SiebelAnalytics\Repository` folder.

2 In the Mapping pane, expand Dim - Cost Centers, and expand Sources.

3 Double-click on Dim_IA_GL_ACCOUNTS_CostCenter_Segment.

4 In Column Mappings, change the expression column names to the appropriate segment.

5 Double-click on Dim_IA_COST_CENTERS, click the General tab, and clear the Active check box.

6 Save the repository.

7 Click Yes to Check Global Consistency, and click OK when the Warnings are displayed.

Configuring Expenses for Post-Load Processing

Post-load processing procedures for Expenses allow expense cycle time analysis and update aggregate tables when data captures are performed at different frequencies. Each of these features require some configuration to best suit your organization’s needs.

About the S_M_PLP_EXPENSES_INVOICE_UPD_ALT Session

The Siebel Business Analytics prepackage M_PLP_EXPENSES_INVOICE_UPD to run at the same frequency as the NU_AP_XACTS table load, which is its source. NU_AP_XACTS is truncated before each load of the IA_AP_XACTS table. This truncation can lead to data loss if you decide to run the sessions that update NU_AP_XACTS at different times other than when you execute your post-load update processes.

CAUTION: To prevent data loss resulting from truncation of the NU_AP_XACTS table, you must run an alternate PLP mapping that uses IA_AP_XACTS as a source, as described in this topic. This is because this table always contains a set of invoice data.

Depending on how frequently you want to update data that is already loaded in the Siebel Customer-Centric Enterprise Warehouse, you must use one of the following tasks:

- If you change the update mapping frequency on a regular basis, disable the S_M_PLP_EXPENSES_INVOICE_UPD session and schedule S_M_PLP_EXPENSES_INVOICE_UPD_ALT session to run in its place.
- If you miss a single run of the S_M_PLP_EXPENSES_INVOICE_UPD session, run the S_M_PLP_EXPENSES_INVOICE_UPD session, but do not disable the S_M_PLP_EXPENSES_INVOICE_UPD_ALT session.

The S_M_PLP_EXPENSES_INVOICE_UPD_ALT session is found in the Sessions directory.

About Implementing Temporary Storage When Aggregate Load Frequencies Are Modified

The Siebel Business Analytics designed IA_EXPENSES and IA_EXPENSES_A1 loads to be synchronized in order to populate NU_EXPENSES at the same time. An incremental snapshot table, NU_EXPENSES is truncated before every load, which can cause data loss for the time period if IA_EXPENSES and IA_EXPENSES_A1 are loaded asynchronously.

CAUTION: To prevent data loss resulting from truncation of the NU_EXPENSES table, the incremental data from NU_EXPENSES must be stored temporarily until it is loaded into the aggregate table. Therefore, you must build an intermediate table and modify M_PLP_EXPENSES_A1_DERIVE to source from it. Your new intermediate table must have the same structure as the NU_EXPENSES table and be set to truncate before loading the IA_EXPENSES_A1.

Configuring the Extraction of Invoice Details

The Siebel Business Analytics prepackage post-load processing (PLP) mappings to populate the IA_EXPENSE table with up-to-date invoice information from the IA_AP_XACTS table. The IA_AP_XACTS table updates expense invoice information using the REF_DOC_ITEM and REF_DOC_NUM to uniquely identify the expense document:

```
IA_EXPENSE.INVOICE_DOC_NUM = IA_AP_XACTS.REF_DOC_NUM

IA_EXPENSE.INVOICE_DOC_ITEM = IA_AP_XACTS.REF_DOC_ITEM
```

However, if an expense record is distinctly identified by values other than the REF_DOC_ITEM and REF_DOC_NUM, you can use those values by adding a condition to the expression in the applicable PLP mapping.

NOTE: Using your values allows you to narrow the data set.

To configure the Expenses-Accounts Payable reference

1. In PowerCenter Designer, open the Configuration for Post Load Processing folder.
2. Open the M_PLP_EXPENSE_UPD post-load processing mapping.
3. Select the Source Qualifier to open the Edit Transformations box, and select the Properties tab to display the User Defined Join.
4. Select the value in the User Defined Join to open the SQL Editor box and edit the expression.
5. Edit the filter statement by adding your desired condition to the prepackaged expression.
6. Open the SQL Query, and then select Generate SQL.
7. Click OK, and then save your changes.

Process of Aggregating Siebel Strategic Sourcing Analytics Tables

This section contains Siebel Strategic Sourcing Analytics configuration points for aggregating the Purchase Receipts and Purchase Cycle Lines tables.

To aggregate the Purchase Receipts and Purchase Cycle Lines tables, perform the following tasks:

- Configuring the Purchase Receipts Aggregate Table on page 366
- Configuring the Purchase Cycle Lines Aggregate Table on page 371

Related Topics

- About Configuring the Purchase Receipts Aggregate Table on page 365
- About Configuring the Purchase Cycle Lines Aggregate Table on page 370

About Configuring the Purchase Receipts Aggregate Table

The Purchase Receipts aggregate table (IA_PURCH_RCPTS_A1) is used to capture information about the product receipts received from your suppliers and the purchase orders placed on them by your purchasing organization.

For your initial ETL run, you need to configure the GRAIN parameter for the time aggregation level in the Purchase Receipts Aggregate fact table.

For the incremental ETL run, you need to configure the time aggregation level and the source identification. The source identification value represents the source system you are sourcing data from.

You need to configure two parameters to aggregate the Purchase Receipts table for your incremental run:

- GRAIN
- SOURCE_ID

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

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

Table 71 lists the values for the SOURCE_ID parameter. The value of this parameter is preconfigured to reflect the ETL mapping's folder.

Table 71. Source System Parameter Values

Source System	Value
Oracle 11i	OAP11I
SAP R/3	SAPR3
Universal	GENERIC

NOTE: You can change the default value for the Source ID parameter if you use multiple instances of the same source system. For example, you can run multiple instances of SAP R/3 and use separate Source IDs for each instance. You can name the first instance SAPR3_1, the second instance SAPR3_2, and so on.

The Purchase Receipt Lines aggregate table is fully loaded from the base table in the initial ETL run. The table can grow to millions of records. Thus, the Purchase Receipts aggregate table is not fully reloaded from the base table after each incremental ETL run. Siebel Customer-Centric Enterprise 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. Siebel Customer-Centric Enterprise Warehouse finds the records to be deleted in the base table since the last ETL run, and loads them into the NU_PURCH_RCPTS table. The measures in these records are multiplied by (-1). The mapping responsible for this task is suffixed with PRE_D, and it is run before the records are deleted from the base table. The mapping is run in the source-specific workflow.
2. Siebel Customer-Centric Enterprise Warehouse finds the records to be updated in the base table since the last ETL run, and loads them into the NU_PURCH_RCPTS table. The measures in these records are multiplied by (-1). The mapping responsible for this task is suffixed with PRE_U, and it is run before the records are updated in the base table. It is run in the source-specific workflow.
3. Siebel Customer-Centric Enterprise Warehouse finds the inserted or updated records in the base table since the last ETL run, and loads them into the NU_PURCH_RCPTS table, without changing their sign. The mapping responsible for this task is suffixed with POST, 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. Siebel Customer-Centric Enterprise Warehouse aggregates the NU_PURCH_RCPTS table, and joins it with the IA_PURCH_RCPTS_A1 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 INCR.

Configuring the Purchase Receipts Aggregate Table

This task is a step in the Domain Values and CSV Worksheet Files for Siebel Strategic Sourcing Analytics on page 375.

To load the Purchase Receipts aggregate table (IA_PURCH_RCPTS_A1), 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 file_parameters_plp.csv parameters file

1. Open the file_parameters_plp.csv file using Microsoft WordPad or Notepad in the $pmserver\srcfiles folder.
2. Replace the default PARM_SVALUE2 values with your new values.

For a list of values for each parameter see the About Configuring the Purchase Receipts Aggregate Table on page 365.

The default values are shown in the following table:

PARM_TYPE	PARM_CODE	PARM_SVALUE_1	PARM_SVALUE_2	PARM_SVALUE_3
PLP	S_M_PLP_PURCH_RCPTS_POST:GRAIN	S	MONTH	MPLT_ADI_PURCH_RCPTS_PRE_POST
PLP	S_M_PLP_PURCH_RCPTS_A1_AGG_INIT:GRAIN	S	'MONTH'	—

NOTE: You need to use single quotes for the `S_M_PLP_PURCH_RCPTS_A1_AGG_INIT:GRAIN` session value.

3 Save and close the file.

To configure the Purchase Receipts aggregate table for Oracle 11i

1 Open the file_parameters_ora11i.csv file using Microsoft WordPad or Notepad in the `$pmserver\srcfiles` folder.

2 Replace the default PARM_SVALUE2 values with your new values.

For a list of values for each parameter see the About Configuring the Purchase Receipts Aggregate Table on page 365.

The default values are shown in the following table:

PARM_TYPE	PARM_CODE	PARM_SVALUE_1	PARM_SVALUE_2	PARM_SVALUE_3
ORA11I	S_M_I_PURCH_RCPTS_PRE_D:GRAIN	S	MONTH	MPLT_ADI_PURCH_RCPTS_PRE_POST
ORA11I	S_M_I_PURCH_RCPTS_PRE_D:SOURCE_ID	S	'OAP11I'	—
ORA11I	S_M_I_PURCH_RCPTS_PRE_U:GRAIN	S	MONTH	MPLT_ADI_PURCH_RCPTS_PRE_POST
ORA11I	S_M_I_PURCH_RCPTS_PRE_U:SOURCE_ID	S	'OAP11I'	—

NOTE: You need to use single quotes for the `S_M_I_PURCH_RCPTS_PRE_D:SOURCE_ID` and `S_M_I_PURCH_RCPTS_PRE_U:SOURCE_ID` session values.

3 Save and close the file.

To configure the Purchase Receipts aggregate table for SAP R/3

1 Open the `file_parameters_sapr3.csv` file using Microsoft WordPad or Notepad in the `$pmserver\srcfiles` folder.

2 Replace the default PARM_SVALUE2 values with your new values.

For a list of values for each parameter see the About Configuring the Purchase Receipts Aggregate Table on page 365.

The default values are shown in the following table:

PARM_TYPE	PARM_CODE	PARM_SVALUE_1	PARM_SVALUE_2	PARM_SVALUE_3
SAPR3	S_M_S_PURCH_RCPTS_PRE_D:GRAIN	S	MONTH	MPLT_ADI_PURCH_RCPTS_PRE_POST
SAPR3	S_M_S_PURCH_RCPTS_PRE_D:SOURCE_ID	S	'SAPR3'	—
SAPR3	S_M_S_PURCH_RCPTS_PRE_U:GRAIN	S	MONTH	MPLT_ADI_PURCH_RCPTS_PRE_POST
SAPR3	S_M_S_PURCH_RCPTS_PRE_U:SOURCE_ID	S	'SAPR3'	—

NOTE: You need to use single quotes for the `S_M_S_PURCH_RCPTS_PRE_D:SOURCE_ID` and `S_M_S_PURCH_RCPTS_PRE_U:SOURCE_ID` session values.

3 Save and close the file.

To configure the Purchase Receipts aggregate table for Universal Source

1 Open the `file_parameters_univ.csv` file using Microsoft WordPad or Notepad in the `$pmserver\srcfiles` folder.

2 Replace the default PARM_SVALUE2 values with your new values.

For a list of values for each parameter see the About Configuring the Purchase Receipts Aggregate Table on page 365.

The default values are shown in the following table:

PARM_TYPE	PARM_CODE	PARM_SVALUE_1	PARM_SVALUE_2	PARM_SVALUE_3
UNIV	S_M_F_PURCH_RCPTS_PRE_U:GRAIN	S	MONTH	MPLT_ADI_PURCH_RCPTS_PRE_POST
UNIV	S_M_F_PURCH_RCPTS_PRE_U:SOURCE_ID	S	'GENERIC'	—

> **NOTE:** You need to use single quotes for the `S_M_F_PURCH_RCPTS_PRE_U:SOURCE_ID` session value.

3 Save and close the file.

About Configuring the Purchase Cycle Lines Aggregate Table

To aggregate the Purchase Cycle Lines table (`IA_PURCH_CYCLNS_A1`), you need to configure the `file_parameters_plp.csv` parameters file and the source system parameters file, and run the initial ETL workflow and then the incremental ETL workflow.

For your initial ETL run, you need to configure the GRAIN parameter for the time aggregation level in the Purchase Cycle Lines Aggregate fact table.

For the incremental ETL run, you need to configure the time aggregation level and the source identification. The source identification value represents the source system you are sourcing data from.

You need to configure two parameters to aggregate the Purchase Cycle Lines table for your incremental run:

- GRAIN
- SOURCE_ID

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

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

Table 72 lists the values for the SOURCE_ID parameter. The value of this parameter is preconfigured to reflect the ETL mapping's folder.

Table 72. Source System Parameter Values

Source System	Value
Oracle 11i	OAP11I
SAP R/3	SAPR3
Universal	GENERIC

NOTE: You can change the default value for the Source ID parameter if you use multiple instances of the same source system. For example, you can run multiple instances of SAP R/3 and use separate

Source IDs for each instance. You can name the first instance SAPR3_1, the second instance SAPR3_2, and so on.

The Purchase Cycle Lines aggregate table is fully loaded from the base table in the initial ETL run. The table can grow to millions of records. The Purchase Cycle Lines aggregate table is not fully reloaded from the base table after an ETL run. Siebel Customer-Centric Enterprise 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. Siebel Customer-Centric Enterprise Warehouse finds the records to be deleted in the base table since the last ETL run, and loads them into the NU_PURCH_CYCLNS table. The measures in these records are multiplied by (-1). The mapping responsible for this task is suffixed with PRE_D, and it is run before the records are deleted from the base table. It is run in the source-specific workflow.
2. Siebel Customer-Centric Enterprise Warehouse finds the records to be updated in the base table since the last ETL run, and loads them into the NU_PURCH_CYCLNS table. The measures in these records are multiplied by (-1). The mapping responsible for this task is suffixed with PRE_U, and it is run before the records are updated in the base table. It is run in the source-specific workflow.
3. Siebel Customer-Centric Enterprise Warehouse finds the inserted or updated records in the base table since the last ETL run, and loads them into the NU_PURCH_CYCLNS table, without changing their sign. The mapping responsible for this task is suffixed with POST, 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. Siebel Customer-Centric Enterprise Warehouse aggregates the NU_PURCH_CYCLNS table, and joins it with the IA_PURCH_CYCLNS_A1 aggregate table to insert new or update existing buckets buckets to the aggregate table. This step is part of the post load-processing workflow, and the mapping is suffixed with INCR.

Configuring the Purchase Cycle Lines Aggregate Table

This task is a step in the Domain Values and CSV Worksheet Files for Siebel Strategic Sourcing Analytics on page 375.

To load the Purchase Cycle Lines aggregate table (IA_PURCH_CYCLNS_A1), 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 file_parameters_plp.csv parameters file

1. Open the `file_parameters_plp.csv` file using Microsoft WordPad or Notepad in the `$pmserver\srcfiles` folder.
2. Replace the default PARM_SVALUE2 values with your new values.

For a list of values for each parameter see the About Configuring the Purchase Cycle Lines Aggregate Table on page 370.

The default values are shown in the following table:

PARM_TYPE	PARM_CODE	PARM_SVALUE_1	PARM_SVALUE_2	PARM_SVALUE_3
PLP	S_M_PLP_PURCH_CYCLNS_A1_POST:GRAIN	S	MONTH	MPLT_ADI_PURCH_CYCLNS_NU
PLP	S_M_PLP_PURCH_CYCLNS_A1_LOAD_INIT:GRAIN	S	'MONTH'	—

NOTE: You need to use single quotes for the `S_M_PLP_PURCH_CYCLNS_A1_AGG_INIT:GRAIN` session value.

3 Save and close the file.

To configure the Purchase Cycle Lines aggregate table for Oracle 11i

1 Open the `file_parameters_ora11i.csv` file using Microsoft WordPad or Notepad in the `$pmserver\srcfiles` folder.

2 Replace the default PARM_SVALUE2 values with your new values.

For a list of values for each parameter see the About Configuring the Purchase Cycle Lines Aggregate Table on page 370.

The default values are shown in the following table:

PARM_TYPE	PARM_CODE	PARM_SVALUE_1	PARM_SVALUE_2	PARM_SVALUE_3
ORA11I	S_M_I_PURCH_CYCLNS_PRE_D:GRAIN	S	MONTH	MPLT_ADI_PURCH_CYCLNS_NU
ORA11I	S_M_I_PURCH_CYCLNS_PRE_D:SOURCE_ID	S	'OAP11I'	—

PARM_TYPE	PARM_CODE	PARM_SVALUE_1	PARM_SVALUE_2	PARM_SVALUE_3
ORA11I	S_M_I_PURCH_CYCLNS_PRE_U:GRAIN	S	MONTH	MPLT_ADI_PURCH_CYCLNS_NU
ORA11I	S_M_I_PURCH_CYCLNS_PRE_U:SOURCE_ID	S	'OAP11I'	—

NOTE: You need to use single quotes for the `S_M_I_PURCH_CYCLNS_PRE_D:SOURCE_ID` and `S_M_I_PURCH_CYCLNS_PRE_U:SOURCE_ID` session values.

3 Save and close the file.

To configure the Purchase Cycle Lines aggregate table for SAP R/3

1 Open the `file_parameters_sapr3.csv` file using Microsoft WordPad or Notepad in the `$pmserver\srcfiles` folder.

2 Replace the default PARM_SVALUE2 values with your new values.

For a list of values for each parameter see the About Configuring the Purchase Cycle Lines Aggregate Table on page 370.

The default values are shown in the following table:

PARM_TYPE	PARM_CODE	PARM_SVALUE_1	PARM_SVALUE_2	PARM_SVALUE_3
SAPR3	S_M_S_PURCH_CYCLNS_PRE_D:GRAIN	S	MONTH	MPLT_ADI_PURCH_CYCLNS_NU
SAPR3	S_M_S_PURCH_CYCLNS_PRE_D:SOURCE_ID	S	‘SAPR3'	—
SAPR3	S_M_S_PURCH_CYCLNS_PRE_U:GRAIN	S	MONTH	MPLT_ADI_PURCH_CYCLNS_NU
SAPR3	S_M_S_PURCH_CYCLNS_PRE_U:SOURCE_ID	S	‘SAPR3'	—

NOTE: You need to use single quotes for the `S_M_S_PURCH_CYCLNS_PRE_D:SOURCE_ID` and `S_M_S_PURCH_CYCLNS_PRE_U:SOURCE_ID` session values.

3 Save and close the file.

To configure the Purchase Cycle Lines aggregate table for Universal Source

1. Open the `file_parameters_univ.csv` file using Microsoft WordPad or Notepad in the `$pmserver\srcfiles` folder.
2. Replace the default PARM_SVALUE2 values with your new values.

   For a list of values for each parameter see the About Configuring the Purchase Cycle Lines Aggregate Table on page 370.

   The default values are shown in the following table:

PARM_TYPE	PARM_CODE	PARM_SVALUE_1	PARM_SVALUE_2	PARM_SVALUE_3
UNIV	S_M_F_PURCH_CYCLNS_PRE_U:GRAIN	S	MONTH	MPLT_ADI_PURCH_CYCLNS_NU
UNIV	S_M_F_PURCH_CYCLNS_PRE_U:SOURCE_ID	S	'GENERIC'	—

NOTE: You need to use single quotes for the `S_M_F_PURCH_CYCLNS_PRE_U:SOURCE_ID` session value.

3 Save and close the file.

Domain Values and CSV Worksheet Files for Siebel Strategic Sourcing Analytics

Table 73 lists the CSV worksheet files and the domain values for Siebel Strategic Sourcing Analytics in the $pmserver\LkpFiles folder.

For more information on configuring domain values with CSV worksheet files, see About Domain Values on page 154 and Configuring the Domain Value Set with CSV Worksheet Files on page 159.

Table 73. Domain Values and CSV Worksheet Files for Siebel Strategic Sourcing Analytics

Worksheet File Name	Description	Session
domainValues_Status_Purch_Approve_ora11i.csv	Lists the Purchasing Approval Status column and the corresponding domain values for the Oracle 11i application.	S_M_I_STATUS_PURCH_APPROVE_LOAD
domainValues_Status_Purch_Cycle_ora11i.csv	Lists the Purchasing Cycle Status column and the corresponding domain values for the Oracle 11i application.	S_M_I_STATUS_PURCH_CYCLE_LOAD
domainValues_Status_Purch_Cycle_sapr3.csv	Lists the Purchase Order Cycle Status column and the corresponding domain values for the SAP R/3 application.	S_M_S_PURCH_ORDERS_EXTRACT, S_M_S_PURCH_SCHLNS_EXTRACT
domainValues_Status_PurchRqsn_Cycle_sapr3.csv	Lists the Purchase Requisition Cycle Status column and the corresponding domain values for the SAP R/3 application.	S_M_S_PURCH_RQLNS_EXTRACT
domainValues_Xact_Types_Purch_Orders_ora11i.csv	Lists the Purchase Order Transaction Type column and the corresponding domain values for the Oracle 11i application.	S_M_I_XACT_TYPES_PURCH_ORDERS_LOAD
domainValues_Xact_Types_Purch_Orders_sapr3.csv	Lists the Purchase Order Transaction Type column and the corresponding domain values for the SAP R3 application.	S_M_S_XACT_TYPES_PURCH_ORDERS_LOAD, S_M_S_XACT_TYPES_PURCH_RQLNS_LOAD
domainValues_Xact_Types_Purch_Requisitions_ora11i.csv	Lists the Purchase Requisition Transaction Type column and the corresponding domain values for the Oracle 11i application.	S_M_I_XACT_TYPES_PREQ_LOAD
domainValues_Xact_Types_Purch_Requisitions_sapr3.csv	Lists the Purchase Requisition Transaction Type column and the corresponding domain values for the SAP R/3 application.	S_M_S_XACT_TYPES_PURCH_RQLNS_LOAD

18 Configuring Siebel Supply Chain Analytics

This chapter describes how to configure certain objects for particular sources to meet your business needs.

It contains the following topics:

Overview of Siebel Supply Chain Analytics

The Siebel Supply Chain Analytics applications allows you to analyze your organization's inventory to maintain a tight operating and audit control. Inventory is a considerable cost to most organizations, but it is important to adequately serve customer sales. Managing and minimizing your organization's inventory investment can reduce costs and ultimately contribute to the competitiveness of your organization. Because operating at the lowest cost while maintaining customer service allows for competitive advantage, minimizing your inventory costs can help your organization in the marketplace.

The Siebel Supply Chain Analytics applications has four functional areas:

- **Bill of Materials.** The Bill of Materials (BOM) functional area allows you to determine the profit margin of the components that comprise the finished goods. BOM allows you to keep up with the most viable vendors in terms of cost and profit, and to keep your sales organization aware of product delivery status, including shortages.
- **Inventory Transactions.** The Inventory Transactions functional area allows you to analyze the various types of events and tasks that occur. Examples of these activities include tracking inventory by type of movement. For example, transfer, issues, receipts, returns, sales, and so on. It allows the user to understand the impact of these activities on business operations, and allows the identification of problematic trends early. For example, large quantities of product in-transit.
- **Inventory Balances.** The Inventory Balances functional area allows you to analyze the inventory held by an organization in relation to a number of different dimensions. For example, Product type, Product number, Storage location, Plant, Consigned Inventory, Restricted, and so on. It allows the user the ability to understand and determine the optimal distribution of assets as well as identify potential issues such as unnecessary build up of inventories.

- **Customer and Supplier Returns.** The Customer and Supplier Returns functional area allows the user to specifically monitor the return of product by both Customers and Suppliers. At a Product level, it allows the user to identify early, potential, Customer-satisfaction issues relating to problematic Suppliers and Product.

Process of Configuring Siebel Supply Chain Analytics for Oracle 11i

This section contains Siebel Supply Chain Analytics configuration points that are specific to Oracle 11i. It contains the following topics:

- Configuring the Bill of Materials Explosion on page 378
- Configuring the Left Bound and Right Bound Calculation Option on page 382
- Configuring Quantity Types for Product Transactions on page 384
- Configuring the Region Name on page 385
- Configuring the State Name on page 386
- Configuring the Country Name on page 386
- Configuring the Make-Buy Indicator on page 387

Configuring the Bill of Materials Explosion

This task is a step in the Process of Configuring Siebel Supply Chain Analytics for Oracle 11i on page 378.

The Bill of Materials (BOM) functional area allows you to determine the profit margin of the components that comprise the finished goods. BOM allows you to keep up with the most viable vendors in terms of cost and profit, and to keep your sales organization aware of product delivery status, including shortages.

You can explode the BOM structure with three different options:

- **All.** All the BOM components are exploded regardless of their effective date or disable date. To *explode* a BOM component is to expand the BOM tree structure.
- **Current.** The incremental extract logic considers any changed components that are currently effective, any components that are effective after the last extraction date, or any components that are disabled after the last extraction date.
- **Current and Future.** All the BOM components that are effective now or in the future are exploded. The disabled components are left out.

These options are controlled by the `EXPLODE_OPTION` variable. The `EXPLODE_OPTION` variable is preconfigured with a value of 2, explode Current BOM structure.

In the M_I_BOM_Explode mapping, the bompexpl.exploder_userexit stored procedure explodes the BOM structure. Table 74 lists the variables for the bompexpl.exploder_userexit stored procedure.

Table 74. Variables for the bompexpl.exploder_userexit Stored Procedure

Input Variable	Preconfigured Value	Description
VERIFY_FLAG	0	A verify flag with a value of 1 only applies to standard BOM.
ORG_ID	ORGANIZATION_ID	Organization ID
ORDER_BY	1	Controls the order of the records. 1—Operation Sequence Number, Item Number 2—Item Number, Operation Sequence Number
GRP_ID	Negative Sequence ID -1, -2, and so on.	Unique value to identify current explosion.
SESSION_ID	Negative Sequence ID -1, -2, and so on.	Unique value to identify current session.
LEVELS_TO_EXPLODE	10	Levels to explode.
BOM_OR_ENG	1	1—BOM 2—ENG
IMPL_FLAG	1	1—Implemented Only 2—Implemented and Unimplemented
PLAN_FACTOR	2	1—Yes 2—No
EXPLODE_OPTION	2	1—All 2—Current 3—Current and Future
MODULE	2	1—Costing 2—BOM 3—Order Entry 4—ATO 5—WSM

Table 74. Variables for the bompexpl.exploder_userexit Stored Procedure

Input Variable	Preconfigured Value	Description
CST_TYPE_ID	0	Cost type ID for cost explosion.
STD_COMP_FLAG	0	1—Explode only standard components 2—All components
EXPL_QTY	1	Explosion quantity
ITEM_ID	ROUND(TO_DECIMAL(PRODUCT_ID))	Item ID of assembly to explode.
ALT_DESG	ALTERNATE_BOM_DESIGNATOR	Alternate routing designator
COMP_CODE	NULL	Concatenated component code
REV_DATE	TO_CHAR(CREATION_DT, 'YYYY/MM/DD HH24:MI')	Explosion date YYYY/MM/DD HH24:MI

There are five different BOM types in a source system—1- Model, 2 - Option Class, 3 - Planning, 4 - Standard, and 5 - Product Family. By default, only the Standard BOM type is extracted and exploded.

To configure the BOM explosion to the All option

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. Open the `M_I_BOM_EXPLODE` mapping.
3. Double-click the `EXP_BOMPEXPL_EXPLODER_USEREXIT` expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the `EXPLODE_OPTION` port.
4. Change the value to 1, and click Apply.
5. Validate the mapping, and save your changes to the repository.
6. Open the `MPLT_BCI_STAGE_BOM_COMPONENTS` mapplet.
7. Double-click the SQL qualifier `SQ_BOM_INVENTORY_COMPONENTS` to open the Edit Transformations dialog box, and click on Properties tab, open value for SQL Query.
8. Modify the INV.EFFECTIVITY_DATE and INV.DISABLE_DATE portions in the following default Where statement.

```
INV.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY HH24:MI:SS')
AND (INV.EFFECTIVITY_DATE <= TO_DATE('$$CURRENT_DATE','MM/DD/YYYY HH24:MI:SS')
and (INV.DISABLE_DATE > TO_DATE('$$CURRENT_DATE','MM/DD/YYYY HH24:MI:SS') OR
INV.DISABLE_DATE IS NULL))
```

OR

```
INV.EFFECTIVITY_DATE between TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY
HH24:MI:SS') and TO_DATE('$$CURRENT_DATE','MM/DD/YYYY HH24:MI:SS')

 OR

INV.DISABLE_DATE between TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY HH24:MI:SS')
and TO_DATE('$$CURRENT_DATE','MM/DD/YYYY HH24:MI:SS')

)

OR BOM.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY
HH24:MI:SS'))

GROUP BY
```

as follows:

```
INV.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY HH24:MI:SS')

 OR BOM.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY
HH24:MI:SS'))

GROUP BY
```

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

To configure the BOM explosion to the Current and Future option

1 In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.

2 Open the `M_I_BOM_EXPLODE` mapping.

3 Double-click the `EXP_BOMPEXPL_EXPLODER_USEREXIT` expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the `EXPLODE_OPTION` port.

4 Change the value to 3, and click Apply.

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

6 Open the `MPLT_BCI_STAGE_BOM_COMPONENTS` mapplet.

7 Double-click the SQL qualifier `SQ_BOM_INVENTORY_COMPONENTS` to open the Edit Transformations dialog box, and click on Properties tab, open value for SQL Query.

8 Modify the INV.EFFECTIVITY_DATE and INV.DISABLE_DATE portions in the following default Where statement.

```
INV.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY HH24:MI:SS')
AND (INV.EFFECTIVITY_DATE <= TO_DATE('$$CURRENT_DATE','MM/DD/YYYY HH24:MI:SS')
and (INV.DISABLE_DATE > TO_DATE('$$CURRENT_DATE','MM/DD/YYYY HH24:MI:SS') OR
INV.DISABLE_DATE IS NULL))

 OR

INV.EFFECTIVITY_DATE between TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY
HH24:MI:SS') and TO_DATE('$$CURRENT_DATE','MM/DD/YYYY HH24:MI:SS')

 OR
```

```
INV.DISABLE_DATE between TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY HH24:MI:SS')
and TO_DATE('$$CURRENT_DATE','MM/DD/YYYY HH24:MI:SS')
```

as follows

```
INV.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY HH24:MI:SS')
AND ((INV.DISABLE_DATE > TO_DATE('$$CURRENT_DATE','MM/DD/YYYY HH24:MI:SS') OR
INV.DISABLE_DATE IS NULL))

 OR

INV.DISABLE_DATE between TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY HH24:MI:SS')
and TO_DATE('$$CURRENT_DATE','MM/DD/YYYY HH24:MI:SS')

)

 OR BOM.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE','MM/DD/YYYY
HH24:MI:SS'))
```

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

To configure the BOM type

1 In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.

2 Open the `MPLT_BCI_STAGE_BOM_COMPONENTS` mapplet.

3 Double-click the SQL qualifier `SQ_BOM_INVENTORY_COMPONENTS` to open the Edit Transformations dialog box, and click on Properties tab, open value for SQL Query.

4 Modify `BOM_ITEM_TYPE` section in Where statement.

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

```
Where INV.BOM_ITEM_TYPE = 3 AND

M.BOM_ITEM_TYPE = 3 AND
```

NOTE: You can also remove these two filters to extract all types of BOM.

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

Configuring the Left Bound and Right Bound Calculation Option

This task is a step in the Process of Configuring Siebel Supply Chain Analytics for Oracle 11i on page 378.

You can use the left bound and the right bound calculation to expedite some reporting requirements. For example, you can find the components in a subassembly within a finished product. Left bound and right bound are stored in the `IA_BOM_ITEMS` table for each BOM node, and they have one row of data in the `IA_BOM_ITEMS` table. The `COMPUTE_BOUNDS` stored procedure traverses the exploded BOM tree structure and calculates the left bound and right bound. By default, the `COMPUTE_BOUNDS` stored procedure is off and the IA_BOM_ITEMS.LEFT_BOUNDS and IA_BOM_ITEMS.RIGHT_BOUNDS columns are empty.

Figure 58 illustrates a sample BOM structure with the left bound and right bound values listed for each node. To find all the components under node B, you select the components with a top product key value of A, the left bound value is greater than 2, and the right bound value is less than 17.

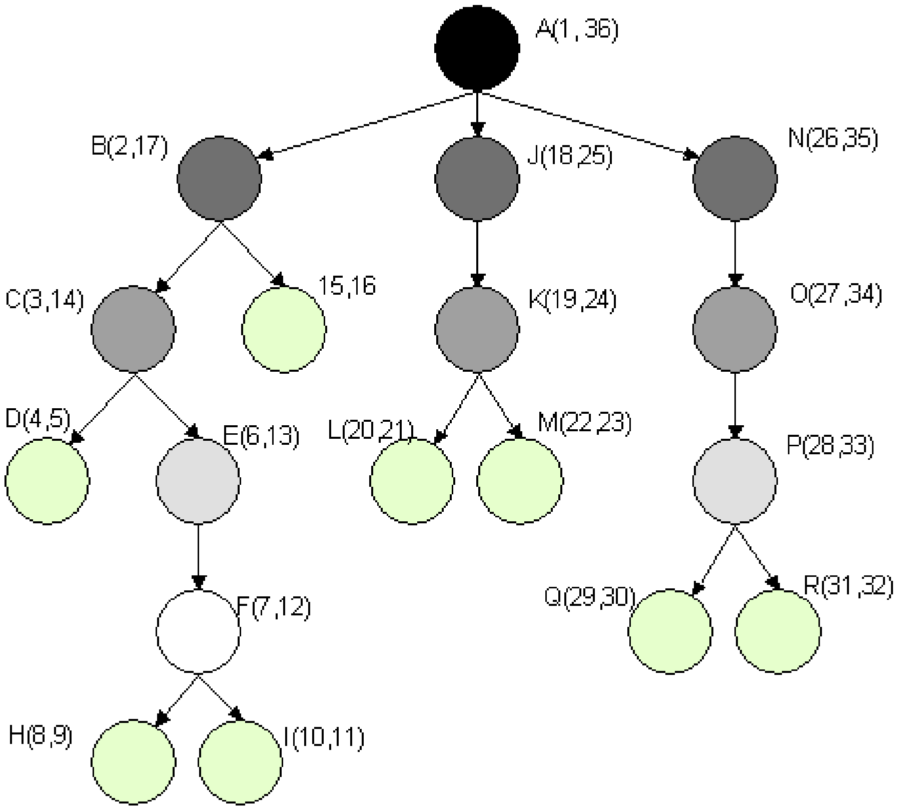


Figure 58. Sample BOM Structure

You can use the following procedure to turn on the left bound and the right bound calculation and populate the `IA_BOM_ITEMS.LEFT_BOUNDS` and `IA_BOM_ITEMS.RIGHT_BOUNDS` columns.

To configure the left bound and right bound calculation option

1 In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.

2 Open the `M_I_BOM_ITEMS_LOAD` mapping.

3. Double-click the COMPUTE_BOUNDS stored procedure transformation to open the Edit Transformations dialog box, and click the Properties tab.
4. Change the value for Transformation Attributes Call Text to compute_bounds_ora11i(1).
5. Click Apply.
6. Validate the mapping, and save your changes to the repository.

Configuring Quantity Types for Product Transactions

This task is a step in the Process of Configuring Siebel Supply Chain Analytics for Oracle 11i on page 378.

Oracle 11i categorize quantities into three different types:

- **Goods Received quantities.** *Goods Received quantity* refers to the number of goods received.
- **Delivery quantities.** *Delivery quantity* refers to the number of goods delivered.
- **Base quantities**. *Base quantity* refers to any transaction quantity.

The Siebel Customer-Centric Enterprise Warehouse extracts the transaction type and loads this value into the XACT_SRC_TYPE column. In this column, the value 1 denotes a Goods Received quantity, and 2 denotes a Delivery quantity.

All quantities extracted from the source system are always loaded into the Base quantity column (EXT_BASE_QTY). However, only the receipt quantity is loaded into the Goods Received quantity column (EXT_GR_QTY), and only delivered quantities are loaded into the Delivery quantity column (EXT_DELIVERY_QTY).

If your definition of goods received or delivery quantity is different from the prepackaged condition, then you can edit the condition to suit your business needs.

To configure the Quantity type

1. In PowerCenter Designer, open the Configuration for Oracle Applications 11i folder.
2. Open the MPLT_SAI_PROD_XACTS mapplet.
3. Double-click the Expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the EXT_GR_QTY and EXT_DELIVERY_QTY port.
4. Edit the quantity types by substituting your desired condition for the prepackaged expression.
5. Click Apply.
6. Validate the mapplet, and save your changes to the repository.

About Configuring the Handling of Currency Types

The Siebel Supply Chain Analytics applications, like all other applications, uses the same method for handling currency conversions from document currency to local and group currencies. The only distinction is how currency is handled for product transactions. In a product transaction, if the document currency code is not supplied, the ADI uses the same code that it uses for the local currency code, and the local currency is derived using the Set of Books ID.

This guide provides a functional overview of how local and group currencies are derived depending on what data is supplied to the Siebel Customer-Centric Enterprise Warehouse. For more information on how to configure various components that relate to local, document, and group currencies, see About Document, Local, and Group Currencies on page 136.

Configuring the Region Name

This task is a step in the Process of Configuring Siebel Supply Chain Analytics for Oracle 11i on page 378.

For Oracle 11i, you can reconfigure the region, state, and country names. This configuration information applies only to plant, storage, and supplier locations. By default, the Region Name column (EXT_REGION_NAME) is populated using the same code value as the Region Code column (EXT_REGION_CODE). However, you can redefine the load mapping's Source Adapter mapplet to load a source-supplied region name instead of the code. If you want to reconfigure the load in this manner, you can load the region code and region name into the IA_CODES table. For information on loading codes and code names into the IA_CODES table, see Codes Lookup on page 147.

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

To configure the Region Name

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. Open the mapplet you want to edit.

   The following is a list of all Source Adapter mapplets that use the EXT_REGION_NAME column:

   MPLT_SAI_SUPPLIERS

   MPLT_SAI_BUSN_LOCS_PLANT

   MPLT_SAI_BUSN_LOCS_STORAGE_LOC
3. Double-click the Expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the EXT_REGION_NAME port.
4. Edit the condition by removing the assigned value if you want the lookup to occur.
5. Click Apply.

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

Configuring the State Name

This task is a step in the Process of Configuring Siebel Supply Chain Analytics for Oracle 11i on page 378.

For Oracle 11i, you can reconfigure the region, state, and country names that apply to the Supplier locations only. By default, the State Name column (EXT_STATE_NAME) is populated using the same code value as the State Code column (EXT_STATE_CODE). However, you can redefine the load mapping's Source Adapter mapplet to load a source-supplied state name instead of the code. If you want to reconfigure the load in this manner, you can load the state code and state name into the IA_CODES table. For information on loading codes and code names into the IA_CODES table, see Codes Lookup on page 147.

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

To configure the State Name

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. Open the MPLT_SAI_SUPPLIERS mapplet.
3. Double-click the Expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the EXT_STATE_NAME port.
4. Edit the condition by removing the assigned value if you want the lookup to occur.
5. Click Apply.
6. Validate the mapplet and save your changes to the repository.

Configuring the Country Name

This task is a step in the Process of Configuring Siebel Supply Chain Analytics for Oracle 11i on page 378.

For Oracle 11i, you can reconfigure the region, state, and country names that apply to supplier locations only. By default, the Country Name column (EXT_COUNTRY_NAME) is populated using the same code value as the Country Code column (EXT_COUNTRY_CODE). However, you can redefine the load mapping's Source Adapter mapplet to load a source-supplied country name instead of the code. If you want to reconfigure the load in this manner, you can load the country code and country name into the IA_CODES table. For information on loading codes and code names into the IA_CODES table, see Codes Lookup on page 147.

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

To configure the Country Name

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. Open the MPLT_SAI_SUPPLIERS mapplet.
3. Double-click the Expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the EXT_COUNTRY_NAME port.
4. Edit the condition by removing the assigned value if you want the lookup to occur.
5. Click Apply.
6. Validate the mapplet, and save your changes to the repository.

Configuring the Make-Buy Indicator

This task is a step in the Process of Configuring Siebel Supply Chain Analytics for Oracle 11i on page 378.

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

Your organization may require different indicator codes. If so, you can modify the indicator logic by reconfiguring the condition in the mapplet MPLT_SAO_PRODUCTS. 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 PowerCenter Designer, open the Configuration for Oracle Applications v11i folder.
2. Open the MPLT_SAI_PRODUCTS mapplet.
3. Double-click the Expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the EXT_MAKE_BUY_IND port.
4. Edit the condition by replacing the prepackaged condition with your desired logic.
5. Click Apply.
6. Validate the mapplet, and save your changes to the repository.

Process of Aggregating Siebel Supply Chain Analytics Tables

This section contains Siebel Strategic Sourcing Analytics configuration points for aggregating the Inventory Balance and Product Transaction tables.

It contains the following topics:

- Configuring the Inventory Balance Aggregate Table on page 389
- Configuring the Product Transaction Aggregate Table on page 391

Related Topics

- About Configuring the Inventory Balance Aggregate Table on page 388
- About Configuring the Product Transaction Aggregate Table on page 390

About Configuring the Inventory Balance Aggregate Table

To configure the Inventory Balance (IA_INV_BALANCE_A1) aggregate table, you need to consider the aggregation level, the time period to update the aggregation, and the time period to keep records in the Inventory Balance table.

You need to configure three parameters to aggregate the Inventory Balance table:

- GRAIN
- KEEP_PERIOD
- NUM_OF_PERIOD

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

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

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

- DAY
- WEEK
- MONTH

- QUARTER
- YEAR

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

Configuring the Inventory Balance Aggregate Table

This task is a step in the Process of Aggregating Siebel Supply Chain Analytics Tables on page 388.

You need to configure the `file_parameters_plp.csv` parameters file, and run the initial ETL session or incremental ETL sessions to load the Inventory Balance aggregate table.

To configure the file_parameters_plp.csv parameters file

1. Open the `file_parameters_plp.csv` file using Microsoft WordPad or Notepad in the `$pmserver\srcfiles` folder.
2. Replace the default PARM_SVALUE2 values with your new values.

   For a list of values for each parameter see the About Configuring the Inventory Balance Aggregate Table on page 388.

   The default values for the `file_parameters_plp.csv` file are shown in the following table.

PARM_TYPE	PARM_CODE	PARM_SVALUE_1	PARM_SVALUE_2
PLP	S_M_PLP_INV_BALANCE_A1_AGG:GRAIN	S	'MONTH'
PLP	S_M_PLP_INV_BALANCE_TRIM:KEEP_PERIOD	S	'MONTH'
PLP	S_M_PLP_INV_BALANCE_TRIM:NUM_OF_PERIOD	S	1

   NOTE: You need to use single quotes for the values of the KEEP_PERIOD and GRAIN parameters.

3. Save and close the file.

To configure he Inventory Balance aggregate table

1. Delete the records from the Inventory Balance (IA_INV_BALANCE_A1) aggregate table for a certain time.

   The GRAIN parameter determines the time period for the deletion. For example, if GRAIN=MONTH, and the date is May 15, 2005, then all records for April and the current month (May) are deleted in the IA_INV_BALANCE_A1 table.

   Running the S_M_PLP_INV_BALANCE_A1_AGG session, and the M_PLP_INV_BALANCE_A1_AGG mapping implements this step.

2. Retrieve the records in the Inventory Balance (IA_INV_BALANCE) fact table and aggregate the records to the IA_INV_BALANCE_A1 table at a certain grain level.

   For example, if GRAIN=MONTH then the records in the IA_INV_BALANCE fact table are retrieved and aggregated to the IA_INV_BALANCE_A1 table at a monthly level.

   Running the S_M_PLP_INV_BALANCE_A1_AGG session, and the M_PLP_INV_BALANCE_A1_AGG mapping implements this step.

3. Remove the old records from the IA_INV_BALANCE fact table.

   To remove old records you need to use the KEEP_PERIOD and the NUM_OF_PERIOD parameters. For example, if KEEP_PERIOD=MONTH, NUM_OF_PERIOD=1, and the date is May 15, 2005, then the records for April and the current month (May) are kept and the older records are deleted.

   Running the S_M_PLP_INV_BALANCE_TRIM session, and the M_PLP_INV_BALANCE_TRIM mapping implements this step.

About Configuring the Product Transaction Aggregate Table

There are two aggregation scenarios to configure the Product Transaction aggregate (IA_PROD_XACTS_A1) table—the initial ETL run and then the incremental ETL run.

For your initial ETL run, you need to configure the aggregation level, and the length of history kept in the Product Transaction fact table.

You need to configure three parameters to aggregate the Product Transaction table for you initial run:

- GRAIN
- KEEP_PERIOD
- NUM_OF_PERIOD

For the incremental ETL run, you need to configure the aggregation level, the update period in aggregation, and the length of history kept in the Product Transaction fact table.

You need to configure five parameters to aggregate the Product Transaction aggregate table for you initial run:

- GRAIN
- REFRESH_PERIOD
- NUM_OF_PERIOD (S_M_PLP_PROD_XACTS_A1_AGG)
- KEEP_PERIOD
- NUM_OF_PERIOD (S_M_PLP_PROD_XACTS_TRIM)

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

- DAY
- WEEK

- MONTH
- QUARTER
- YEAR

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

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

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

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

The NUM_OF_PERIOD parameter for the `S_M_PLP_PROD_XACTS_A1_AGG` session has a preconfigured value of 1. The value for the NUM_OF_PERIOD parameter is a positive integer, for example, 1, 2, 3, and so on.

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

Configuring the Product Transaction Aggregate Table

This task is a step in the Process of Aggregating Siebel Supply Chain Analytics Tables on page 388.

You need to configure the `file_parameters_plp.csv` parameters file, and run the initial ETL and then the incremental ETL to load the Product Transaction aggregate table.

To configure the file_parameters_plp.csv parameters file

1. Open the `file_parameters_plp.csv` file using Microsoft WordPad or Notepad in the `$pmserver\srcfiles` folder.
2. Replace the default PARM_SVALUE2 values with your new values.

   For a list of values for each parameter see the About Configuring the Product Transaction Aggregate Table on page 390.

The default values are shown in the following table.

PARM_TYPE	PARM_CODE	PARM_SVALUE_1	PARM_SVALUE_2
PLP	S_M_PLP_PROD_XACTS_TRIM:KEEP_PERIOD	S	'YEAR'
PLP	S_M_PLP_PROD_XACTS_TRIM:NUM_OF_PERIOD	S	3
PLP	S_M_PLP_PROD_XACTS_A1_AGG_INCR:REFRESH_PERIOD	S	'MONTH'
PLP	S_M_PLP_PROD_XACTS_A1_AGG_INCR:GRAIN	S	'MONTH'
PLP	S_M_PLP_PROD_XACTS_A1_AGG_INCR:NUM_OF_PERIOD	S	1
PLP	S_M_PLP_PROD_XACTS_A1_AGG_INIT:GRAIN	S	'MONTH'

NOTE: You need to use single quotes for the values of the KEEP_PERIOD, GRAIN, and REFRESH_PERIOD parameters. The KEEP_PERIOD value must be equal to or greater than the GRAIN value. The REFRESH_PERIOD value must equal the GRAIN value.

3 Save and close the file.

To configure the Product Transaction aggregate table for the initial ETL run

1 Retrieve the records in the Product Transaction fact (IA_PROD_XACTS) table, and aggregate the records to the Product Transaction aggregate (IA_PROD_XACTS_A1) table at a certain grain level.

For example, if GRAIN=MONTH then the records in the IA_PROD_XACTS fact table are retrieved and aggregated to the IA_PROD_XACTS_A1 table at a monthly level.

Running the S_M_PLP_PROD_XACTS_A1_AGG_INIT session, and the M_PLP_PROD_XACTS_A1_AGG_INIT mapping implements this step.

2 Remove the old records from the IA_PROD_XACTS fact table.

To remove old records you need to use the KEEP_PERIOD and the NUM_OF_PERIOD parameters. For example, if KEEP_PERIOD=YEAR, NUM_OF_PERIOD=3, and the date is May 1, 2005, then the records for the years 2002, 2003, and 2004, and the current year (2005), are kept and the older records are deleted.

Running the S_M_PLP_PROD_XACTS_TRIM session, and the M_PLP_PROD_XACTS_TRIM mapping implements this step.

To configure the Product Transaction aggregate table for the incremental ETL run

1 Delete the refreshed records from the Product Transaction aggregate (IA_PROD_XACTS_A1) table for a certain time.

   The REFRESH_PERIOD and the NUM_OF_PERIOD parameters determine the time period for the deletion.

   For example, if REFRESH_PERIOD=MONTH, NUM_OF_PERIOD=1, and the date is May 15, 2005, then all records for April and the current month (May) are deleted in the IA_PROD_XACTS_A1 table.

   Running the S_M_PLP_PROD_XACTS_A1_AGG_INCR session, and the M_PLP_PROD_XACTS_A1_AGG_INCR mapping implements this step.

2 Retrieve the records in the Product Transaction fact (IA_PROD_XACTS) table, and aggregate the records to the IA_PROD_XACTS_A1 table at a certain grain level.

   For example, if GRAIN=MONTH then the records in the IA_PROD_XACTS fact table are retrieved and aggregated to the IA_PROD_XACTS_A1 table at a monthly level.

   Running the S_M_PLP_PROD_XACTS_A1_AGG_INCR session, and the M_PLP_PROD_XACTS_A1_AGG_INCRT mapping implements this step.

3 Remove the old records from the IA_PROD_XACTS fact table.

   To remove old records you need to use the KEEP_PERIOD and the NUM_OF_PERIOD parameters. For example, if KEEP_PERIOD=YEAR, NUM_OF_PERIOD=3, and the date is May 1, 2005, then the records for the years 2002, 2003, and 2004, and the current year (2005), are kept and the older records are deleted.

   Running the S_M_PLP_PROD_XACTS_TRIM session, and the M_PLP_PROD_XACTS_TRIM mapping implements this step.

About the SAP R/3 Inventory Transfer Process for Siebel Supply Chain Analytics

In SAP R/3 you can post a stock transfer from plant to plant using a two-step procedure— post the stock removal from storage at the issuing plant and post the stock placement into storage at the receiving plant.

There are certain movement types you would associate with this two-step process. By default, SAP R/3 uses:

- 303 and 305 in conjunction to carry out a two-step process for transfer posting from a plant to another plant.
- 313 and 315 in conjunction to carry out a two-step process for transfer posting from a storage location to another storage location.

Table 75 lists three records for the 303-305 combination in the IA_PROD_XACTS table in Siebel Customer-Centric Enterprise Warehouse.

Table 75. 303—305 Movement Type

MVMT_TYPE_CODE	BASE_QTY
303	250
303	-250
305	250

The 305 movement type in Table 75 acknowledges the receipt of 250 items in the receiving plant. If you are interested in seeing how much a plant received in a day at gross level, you should exclude the third record from the calculation. Otherwise the quantity is doubled, and your result is 500.

Siebel Supply Chain Analytic is preconfigured to filter out movement types which are associated with the second step in a two-step transfer process. The filtration occurs at the reporting level and not at the ETL level. Therefore, the physical table stores these three records, but the Siebel Analytics Server applies a filter for the 305 movement type. For the 313 and 315 movement types, the filtration occurs at the 315 movement type. If you do not need these filters, remove the content filter (front end metadata) in the Fact - Inventory Transactions logical fact table in the Siebel Business Analytics repository.

Index