Siebel Connector for Oracle Applications

Version 8.0

December 2006

ORACLE

Contents

Chapter 1: What's New in This Release

Chapter 2: Overview of Siebel Connector for Oracle Applications

About the Siebel Connector for Oracle Applications 9

Using the Siebel Connector for Oracle Applications 9

Connector Features 10

- Packaged Integration 10
- Integration Infrastructure 11
- Mobile User Support 11

Architecture 12

- Integration Objects 13
- Integration Object Editor 13
- Integration Object Wizards 13
- Data Transformation Maps and Data Transformation Engine 14
- Business Service Editor 14
- Data Mapper 14
- Integration Workflows 15
- Business Process Designer 15
- Business Integration Manager 15
- Oracle Alerts 15
- Oracle PL/SQL Scripts 15
- EAI Siebel Adapter 16
- Database Adapter 16
- Oracle Procedural Adapter 16
- Buffer Tables 16
- Notification Table 16
- Oracle Receiver 17
- Run-Time Architecture 17

Chapter 3: Installing the Connector

Installation Steps and Dependencies 19

Installation and Configuration Activities 20

Preliminary Requirements and Activities 21
Systems 21
Software 21
Verifying Preliminary Tasks 23
Configuring the Connector 24
Creating the Oracle Database Login 24
Installing the Oracle Client 25
Installing and Configuring the Siebel Enterprise Server and Web Clients 25
Creating the Net Service Name 26
Creating the ODBC Data Source 27
Configuring the Oracle Receiver 29
Configuring the Business Integration Manager 30
Activating Workflows 30
Configuring the Siebel Web Client to Handle Server Requests 31
Running Scripts 31
Post-Installation Oracle Applications Alerts Setup 32

Chapter 4: Setting Up the Connector

Configure EAI Value Maps 33
Creating EAI Value Maps 33
Configuring EAI Value Maps 34
Loading Oracle Applications Data 34
Loading Oracle Applications Data 35
Loading Payment Terms 36
Verifying the Oracle Data Load 36

Chapter 5: System Administrator Procedures

Monitoring Integrations 37
Monitoring Outbound Integrations 37
Monitoring Inbound Integrations 38
Troubleshooting Oracle Connector Messages 39

Chapter 6: User Procedures

Working with Accounts 43
Creating a New Account 43
Updating an Account 44
Working with Orders 44
Creating an Order 45
Updating an Order 45

Verifying an Order Status 46
Understanding Integrations 46

Chapter 7: Developer Procedures

Installing the Development Environment 47
Installing the Oracle Client 48
Installing Siebel Mobile Web Client and Siebel Tools 48
Creating the Net Service Name 48
Creating the ODBC Data Source 49
Editing the Configuration File for Siebel Mobile Web Client 51
Editing the Configuration File for Siebel Tools 51
Integration Overview 52
General 52
EAI Value Mappings 53
Predefined Integration Points 53
Integration Dependencies and Steps 54
Siebel and Oracle Applications Integrations 56
Entity Attribute Mappings (EAMs) 56
Integration Workflows 57
Unicode to Codepage Deployments 65
Deployment Scenarios 66
Character Conversion and Error Detection 67
Understanding the Database Adapter 68
Methods and Arguments 69
Additional Information 72
Understanding the Oracle Receiver 74
Running the Oracle Receiver 74
Operational Summary 75
Notification Table 76
Delete Processing 80
Understanding the Oracle Applications Wizard 80
Operational Summary 80
Working with Integration Objects 81
Running the Wizard 82
Understanding the Oracle Procedure Adapter 83
Operational Summary 84
Methods and Arguments 84

Appendix A: Detailed Data Mappings

Entity Attribute Mappings for Oracle Applications 85
Organization Integration EAM Mappings 85
Inventory Location Integration EAM Mapping 86
Product Integration EAM Mappings 87
Order Integration EAM Mapping 89
EAI Value Mappings for Oracle Applications 97

Appendix B: Oracle Scripts

Oracle Applications Scripts 119

Index

1 What's New in This Release

What's New in Siebel Connector for Oracle Applications, Version 8.0

This guide has been updated to reflect product name changes. It was previously published as Siebel Connector for Oracle Applications, Version 7.5.

2 Overview of Siebel Connector for Oracle Applications

This chapter gives an overview of Oracle's Siebel Connector for Oracle Applications.

About the Siebel Connector for Oracle Applications

Siebel Connector for Oracle Applications provides a complete solution for integrating Oracle's Siebel Business Applications with Oracle Applications. It provides everyone in an organization with a consistent set of data, allowing them to serve customers and cross application boundaries to connect business processes.

Siebel Connector for Oracle Applications allows organizations to manage the total cost of deploying Siebel Business Applications by incorporating packaged integration functionality that reduces the need to define custom integration. Oracle's Siebel Enterprise Applications Integration (EAI) also includes a set of tools that allows organizations to configure and extend the connector.

Using the Siebel Connector for Oracle Applications

You can approach your work with the Siebel Connector for Oracle Applications in three ways, depending upon your needs. You can use the connector as is; you can configure the provided integration points to meet specific business needs; or you can create new integration points based on additional objects.

This guide explains how to use and modify the connector. Chapters provide directions for the following high level procedures.

Checklist		
❑	Install and configure the connector.	For details, see Chapter 3, "Installing the Connector."
❑	Set up the connector after installation.	For details, see Chapter 4, "Setting Up the Connector."
❑	Administer the connector.	For details, see Chapter 5, "System Administrator Procedures."
❑	Perform end user tasks.	For details, see Chapter 6, "User Procedures."
❑	Perform developer tasks.	For details, see Chapter 7, "Developer Procedures."

It also includes detailed descriptions of EAI value and Entity Attribute Mappings (EAMs) - see "Detailed Data Mappings". These connector-related operations are listed in the following sections.

Connector Features

The Siebel Connector for Oracle Applications includes the following features:

- Packaged Integration
- Integrated Infrastructure
- Mobile User Support

These features and associated functions are introduced below.

Packaged Integration

Siebel Connector for Oracle Applications includes pre-defined integration points that facilitate business processes involving extensive data exchanges between Siebel Business Applications and Oracle Applications. The following discussions focus on the business process automation, data synchronization, integration infrastructure, and mobile user support that permit these integrations.

Business Process Automation

Siebel Connector for Oracle Applications allows orders to be processed (captured and fulfilled) between Siebel Applications and Oracle Applications R11i. Using Siebel Business Applications, business users can generate quotes or create orders using information that is synchronized with Oracle Applications, submit orders into Oracle Applications for fulfillment and invoicing, and monitor order fulfillment.

The connector allows several auxiliary business processes to be conducted in conjunction with order capture and fulfillment. These processes include:

- **Account Management.** Users of both Siebel Applications and Oracle Applications can update customer account information.
- **Product Management.** Item definitions can be maintained in Oracle Applications and published to Siebel Applications.
- **Invoicing.** An invoice can be generated from Oracle Applications after an order is fulfilled, using synchronized customer account information from Siebel Applications.
- **Customer Service.** Sales and service professionals can respond to customer inquiries about products, prices, and order status with information obtained from Siebel Applications.

Synchronizing Data

The Connector supports the integration of business processes by synchronizing data objects between the Siebel Business Applications and Oracle Applications. When the system is first set up, an initial data load is used to bring information from Oracle Applications into Siebel Applications. After that, periodic transfers of information between the two applications allow a high level of data consistency. These data integrations are identified in the next sections.

Integrating Siebel Application and Oracle Application Data

Siebel Connector for Oracle Applications provides packaged integration for Siebel Applications with Oracle Applications data. It synchronizes the data between the two applications, as listed in Table 1.

NOTE: See *Siebel System Requirements and Supported Platforms* on Siebel SupportWeb for information on which version of Oracle Applications is supported in Siebel 8.0.

Table 1. Synchronized Business Objects

Siebel Business Object	Oracle Applications Data Object	Direction
Order	Order	Bi-directional data exchanges
Account	Customer	Data exported from Siebel
Product	Item	Data imported from Oracle
Organization	Operating Unit	Data imported from Oracle
Inventory Location	Inventory Organization	Data imported from Oracle

The connector includes mappings of equivalent entities between the two applications. These mappings of commonly-required data types take into account the complicated entity relationships existing in both applications. You can modify these mappings to accommodate your business needs. For specific information about these mappings, see Appendix A, “Detailed Data Mappings.”

Integration Infrastructure

Siebel Connector for Oracle Applications provides tools for customizing its packaged integration capabilities. You can configure the provided integration points or create new ones. In addition to design-time tools, the Connector provides run-time components that can be used to manage data exchanges between the two applications.

Mobile User Support

Siebel Connector for Oracle Applications allows Siebel Business Applications users to access imported Oracle Applications data and generate quotes or create orders when they are disconnected from the network. Here, features of Oracle’s Siebel Remote are used to synchronize information on the Siebel Server with local databases maintained by the Siebel Mobile Web Client.

Incoming transactions from Oracle Applications to Siebel Applications are routed to mobile users according to visibility rules defined in the Siebel Repository for Oracle's Siebel Remote. Outgoing transactions, such as order creation and account updates from mobile users, are queued in their local databases. When these mobile users synchronize with the Siebel Server, their transactions are placed in the server queue and subsequently routed to Oracle Applications through the interfaces described in this guide.

Architecture

The Siebel Connector for Oracle Applications includes the design and run-time tools and components listed in Table 2.

Table 2. Siebel Connector for Oracle Applications Components

Function	Design Time Component	Design Time Tool	Run Time Component
Data Definition	■ Integration Objects ■ Oracle Database Views	■ Integration Object Editor ■ Integration Object Wizards	
Data Transformation	■ Data Transformation Maps	■ Data Mapper ■ Business Service Editor	■ Data Transformation Engine (DTE)
Business Process Integration	■ Integration Workflows ■ Oracle Alerts ■ Oracle PL/SQL Scripts	■ Business Process Designer	■ Business Integration Manager (BIM)
Data Transport			■ EAI Siebel Adapter ■ Database Adapter ■ Oracle Procedural Adapter ■ Buffer Tables ■ Notification Table ■ Oracle Receiver

NOTE: The Database Adapter is also known as the EAI SQL Adapter, and the Oracle Procedural Adapter is also known as EAI ODBC Service.

These Connector components are discussed in the following pages.

Integration Objects

Integration objects are application-neutral representations of common data elements that are exchanged between Siebel Applications and Oracle Applications. When instantiated during run-time, they are used to hold the content of the data being sent from one application to the other. Two integration objects are typically defined for each data flow—an internal data object that represents Siebel Applications data and an external object that represents the corresponding data object in Oracle Applications.

For example, to import customer data from Oracle Applications into Siebel Applications as accounts, you must define an external integration object that represents the Oracle Applications' Customer data and another internal integration object that represents the corresponding Siebel Account business component. The Customer data from Oracle Applications may be imported directly from Oracle base tables or from database views.

On the other hand, to export Siebel account information to Oracle Applications through Oracle's Open Interface Tables, the external business objects must represent data definitions of the Interface Tables.

Two integration objects are provided for each integration point of the Siebel Connector for Oracle Applications' packaged integration. You may configure these objects for your specific needs or create new objects as part of a new integration point definition.

Integration Object Editor

Integration objects can be defined manually using the Integration Object Editor in Oracle's Siebel Tools. The procedure for defining an integration object is similar to the procedure for creating a Siebel Business Object. For more information, see *Integration Platform Technologies: Siebel Enterprise Application Integration* for instructions on defining integration objects manually.

Integration Object Wizards

You can use Wizards within Siebel Tools to create integration objects. These wizards can be used to ascertain the meta-data definitions of applications or data sources, including customizations to data models, and generate appropriate integration objects. You can use the Integration Object Editor to further edit the generated objects.

The following Integration Object Wizards are provided for integrating Siebel Applications with Oracle Applications:

- **Siebel Wizard.** This wizard is used to examine meta-data definitions of Siebel Business Applications and generate internal integration objects that are based on Siebel Business Objects. See *Integration Platform Technologies: Siebel Enterprise Application Integration* for instructions on using this wizard.
- **Oracle Applications Wizard.** This wizard is used to examine meta-data definitions of Oracle Applications that are stored in Oracle FND tables and generate external integration objects based on Oracle base tables and interface tables. See Chapter 7, "Developer Procedures," for instructions on using this wizard.

- **Database Wizard.** If an Oracle Applications element (for example, database views) is not maintained by FND tables, the Database Wizard can be used to examine the native Oracle Database data dictionary and extract the definition. These definitions can be used to create external integration objects that represent Oracle Applications data. See Chapter 7, "Developer Procedures," for instructions on using this wizard.

Data Transformation Maps and Data Transformation Engine

After the internal and external integration objects are created, a Data Transformation Map is used to transform data contained in the two objects. During run-time, a data transformation map is passed to the Siebel Data Transformation Engine (DTE) along with an internal and an external integration object. The map defines the relationships of data elements between the two integration objects. The DTE transforms the data following the map definitions.

Depending on the direction of data transfer, the internal and external integration objects may serve as either the DTE input or output. For example, when information flows inbound from an Oracle to a Siebel application, the external integration object is the DTE input and the output goes to the internal integration object. The arrangement is reversed for an outbound data flow.

A data transformation map is provided for each integration point in the Siebel Connector for Oracle Applications' packaged integration. You may customize a map to meet your requirements or create a new one as part of a new integration point definition. Data transformation maps are defined two ways: by using the Business Service Editor to write an eScript or by using the Siebel Data Mapper.

Business Service Editor

You can write a Siebel Business Service in Oracle's Siebel eScript to process the integration object input to the Siebel Data Transformation Engine (DTE) and invoke DTE data transformation functions that put the transformed data into the DTE output. For more about the Business Service Editor, see *Siebel Tools Online Help*, *Using Siebel Tools*, and *Siebel Developer's Reference*.

Data Mapper

The Siebel Data Mapper is used to define data transformation maps declaratively. Instead of writing eScripts to traverse through integration objects and transform the data, the Data Mapper allows you to define the relationships between the components and fields of the internal and external integration objects. The Siebel Data Transformation Engine (DTE) interprets the relationships and performs the data transformation on your behalf. You may access the Data Mapper through the Siebel Web Client. See *Business Processes and Rules: Siebel Enterprise Application Integration* for instructions on using the Siebel Data Mapper.

Integration Workflows

An Integration Workflow, a type of Siebel Workflow Process, defines the processing steps for data that will be exchanged between Siebel Applications and Oracle Applications. For example, an integration workflow for outbound account data from Siebel Applications to Oracle Applications may contain steps that invoke the EAI Siebel Adapter to query the account object from the Siebel Object Manager, use the DTE to transform the data, and call the Database Adapter to write the transformed account information to Oracle's interface tables. It may also include business logic and error handling steps. A set of pre-defined workflow processes is included in the Siebel Connector for Oracle Applications' integration points. See Chapter 7, "Developer Procedures," for descriptions of these workflow processes.

Business Process Designer

Integration workflows are created and maintained with the Siebel Business Process Designer, a graphical tool accessed through Siebel Tools. See *Siebel Business Process Framework: Workflow Guide* for instructions on using the Designer.

Business Integration Manager

The Siebel Business Integration Manager (BIM) is the run-time component that executes integration workflows. It can handle both interactive and batch data exchanges.

Oracle Alerts

Oracle Alerts are used to capture information that is updated in Oracle Applications. Unlike other Connector components discussed above, Oracle Alerts are installed on Oracle Applications. When a piece of data that must be synchronized with Siebel Business Applications is updated in an Oracle Application, an alert is raised. Then a SQL script is invoked to process the alert. The integration points supplied with the Connector include scripts to create these alerts on Oracle Applications. See the Oracle Applications documentation for instructions on registering alerts.

Oracle PL/SQL Scripts

During the installation of the Siebel Connector for Oracle Applications, PL/SQL scripts are installed in Oracle Applications. During an outbound data flow, these PL/SQL scripts check the data being passed from Siebel Applications to Oracle Applications for validity. See Chapter 7, "Developer Procedures," and the Oracle Applications documentation for instructions on using PL/SQL scripts.

EAI Siebel Adapter

EAI Siebel Adapter moves data into and out of the Siebel Object Manager. When outbound data flows from Siebel Applications to Oracle Applications, EAI Siebel Adapter is the first step of the workflow. It queries the Object Manager for data and puts it into an instantiated integration object for additional processing. When inbound data flows from an Oracle to a Siebel application, EAI Siebel Adapter is the last step of the workflow. It inserts or updates the incoming data into the Siebel Object Manager. See *Integration Platform Technologies: Siebel Enterprise Application Integration* for instructions for using the EAI Siebel Adapter.

Database Adapter

The Database Adapter allows data exchanges between Siebel Applications and Oracle Applications database tables. During an outbound data flow, the Database Adapter writes data to buffer tables in the Oracle Applications database. During an inbound data flow, the Database Adapter reads data from the Oracle Applications' base table or database views and passes it into Siebel Applications. See Chapter 7, "Developer Procedures," for instructions on using this Adapter.

NOTE: The Database Adapter is also known as the EAI SQL Adapter.

Oracle Procedural Adapter

In conjunction with the Database Adapter, the Oracle Procedural Adapter is used to invoke PL/SQL scripts to perform additional business logic validation, move data from buffer tables into Oracle Open Interface Tables, or invoke the Oracle Applications API to send data into Oracle Applications. See Chapter 7, "Developer Procedures," for instructions on using this Adapter.

NOTE: The Oracle Procedural Adapter is also known as EAI ODBC Service.

Buffer Tables

Buffer Tables are used to temporarily hold outbound data from Siebel Applications while it is being prevalidated. After that process, the data is written to the Oracle base tables. The Connector's integration points include scripts to create these tables. See Chapter 7, "Developer Procedures," for instructions on using the Buffer Tables.

Notification Table

A Notification Table stores the Unique Id of a record that is created or updated in Oracle Applications to facilitate inbound data flow. When a record is changed in the Oracle Application, an alert is raised. A SQL script responds to the alert and registers the Unique Id into the Notification Table. The Siebel Connector for Oracle Applications includes a PL/SQL script to install this table into the Oracle Application's database. See Chapter 7, "Developer Procedures," for instructions on using this table.

Oracle Receiver

A constantly running Siebel Server process, the Oracle Receiver regularly polls the Notification Table to look for new Unique Ids that reflect updates to the Oracle Applications base tables. When the Receiver retrieves a Unique Id from the Notification Table, it invokes an Integration Workflow Process to use the Database Adapter to retrieve the actual record. You can use the Siebel Server Manager to configure the Oracle Receiver. See Chapter 7, "Developer Procedures," for more information.

Run-Time Architecture

The component architecture and paths for one-way and two-way synchronous data integrations between Siebel Applications and Oracle Applications are represented in Figure 1.

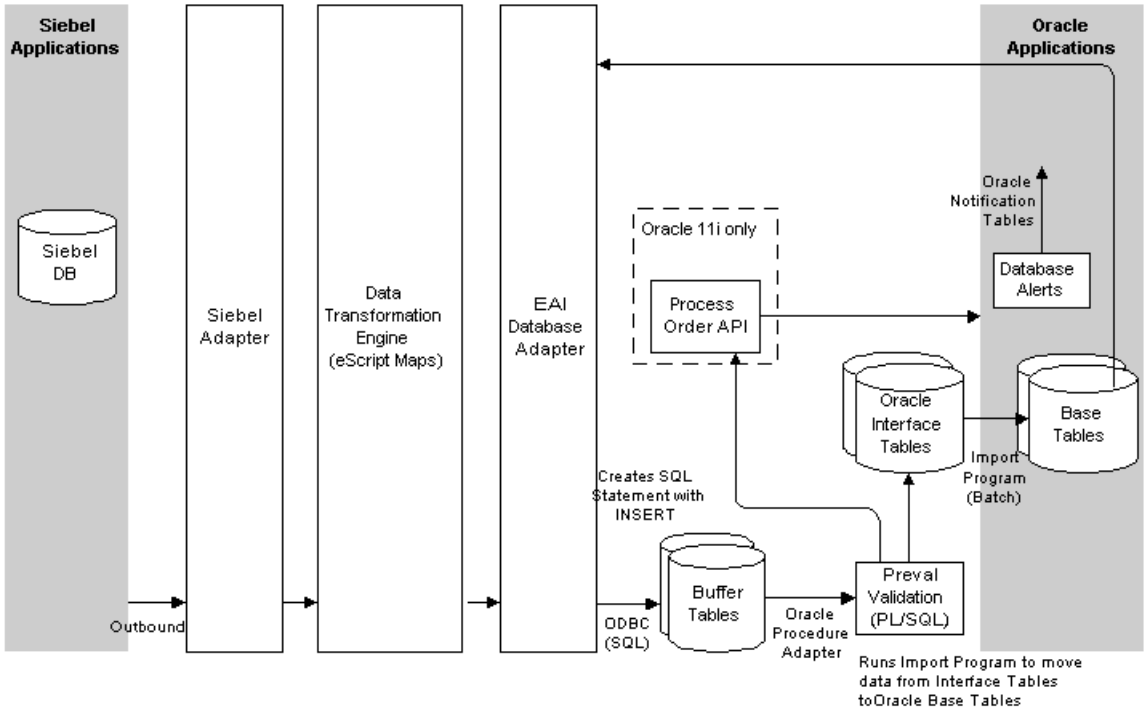


Figure 1. Siebel and Oracle Integration: Inbound and Outbound Process Flow

Outbound Data

For outbound data sent from Siebel Applications to Oracle Applications:

1. A user action in a Siebel Applet, a Siebel Workflow Event, or a Business Integration Manager batch run invokes an Integration Workflow to execute the data transfer.
2. The workflow invokes the EAI Siebel Adapter to query the updated business object through Siebel Object Manager and put the data into an instantiated internal integration object.

3 The integration object is passed to the Data Transformation Engine (DTE). The DTE transforms the object content from Siebel Applications representation into an equivalent Oracle Applications representation and puts it into an instantiated external integration object.

4 The external integration object is passed to the Database Adapter. The Adapter writes the object data to a buffer table.

5 The Oracle Procedural Adapter invokes a PL/SQL script to run a business logic validation.

6 The script sends all data other than Order information to the Open Interface Table. It sends Order information to an Oracle Applications API.

7 If the data goes to the Open Interface Table, the Oracle Concurrent Program Manager runs a program to import the data into the Application's base tables. If the data goes to an API, the API writes the data directly to the base table.

Inbound Data

For inbound data from Oracle Applications to Siebel Applications:

1 An alert is raised when a record is changed in Oracle Applications.

2 To handle the alert, a SQL script runs to register the unique Id of the changed record in the Notification Table.

3 The Oracle Receiver picks up the new table entry and invokes a workflow when it polls the Notification Table for changes.

4 The workflow uses the Unique Id, retrieved by the Oracle Receiver, to invoke the Oracle Adapter. The Adapter queries for the actual changed record from the Oracle Application's base table. The query results are put into an instantiated external integration object.

5 The integration object is passed to the Data Transformation Engine. The DTE transforms the object content from an Oracle Applications representation into an equivalent Siebel Applications representation and puts it into an instantiated internal integration object.

6 The internal integration object is passed to the Siebel Adapter. The Adapter "upserts" (inserts or updates) the data to the Siebel Object Manager and puts it into the Siebel Database.

3 Installing the Connector

This chapter identifies the tasks involved in installing the Siebel Connector for Oracle Applications.

Installation Steps and Dependencies

The Siebel Connector for Oracle Applications is installed in a specific sequence of activities. Figure 2 shows the interdependencies of individual tasks. Arrows indicate dependency. For example, the Siebel Server must be installed and configured (Step 3) before running scripts, activating workflows, creating the ODBC Data Source, and configuring the Siebel Client.

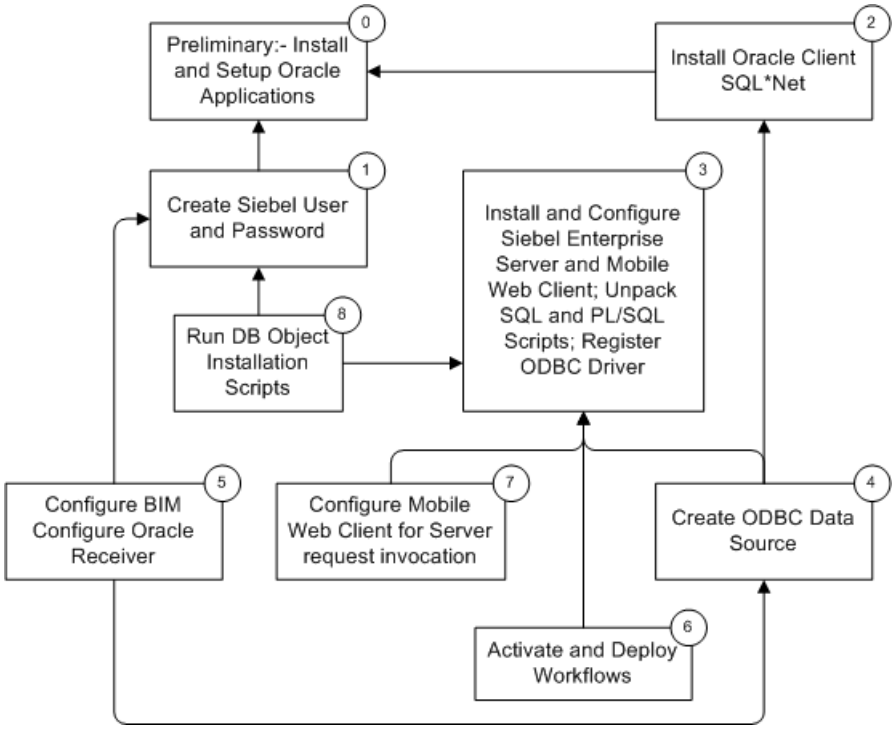


Figure 2. Installing the Connector - Dependencies and Tasks

As Figure 2 on page 19 shows, Siebel Connector for Oracle Applications is installed in eight logically sequential steps. Table 3 identifies these steps, the machine involved in the process, and the person responsible for performing the task.

Installation and Configuration Activities

The installation and configuration activities involved in setting up the connector are identified in the following table.

Table 3. Connector Installation and Configuration Activities

Step	Activity and Discussion	Machine	Responsible Person
(Pre)	This group of preliminary tasks includes installing the Oracle Applications software, and setting it up to enable certain integrations. See "Preliminary Requirements and Activities" on page 21 for a description of the preliminary requirements for the connector installation.	Oracle Database Server	Oracle Applications DBA
(1)	Create the Siebel User and Password for the Oracle Relational Database Management System (RDBMS). For instructions, see "Creating the Oracle Database Login" on page 24.	Oracle Database Server	Oracle Applications DBA
(2)	Install the Oracle Client. For instructions, see "Installing the Oracle Client" on page 25.	Siebel Enterprise Server	Oracle Applications DBA
(3)	Install and configure the Siebel Enterprise Server and Mobile Web Client. This process involves installing and configuring Siebel Enterprise Server and Mobile Web Client software, registering the bundled ODBC driver for accessing Oracle Applications database, and unpacking SQL and PL/SQL scripts that will be used in a later step of the installation. Install Siebel Tools. For a summary, see "Installing and Configuring the Siebel Enterprise Server and Web Clients" on page 25. For detailed instructions, see *Siebel Installation Guide* for the operating system you are using.	Siebel Enterprise Server Siebel Web Clients	Siebel System Administrator
(4)	Create the ODBC Data Source. For instructions, see "Creating the Net Service Name" on page 26.	Siebel Enterprise Server	Siebel System Administrator

Table 3. Connector Installation and Configuration Activities

Step	Activity and Discussion	Machine	Responsible Person
(5)	Configure the Oracle Receiver and the Business Integration Manager (BIM). For instructions, see "Configuring the Oracle Receiver" on page 29 and "Configuring the Business Integration Manager" on page 30.	Siebel Enterprise Server	Siebel System Administrator
(6)	Activate and Deploy Workflows. For instructions, see "Activating Workflows" on page 30.	Siebel Enterprise Server	Siebel System Administrator
(7)	Configure Server Request Invocation if you will be using the Siebel Dedicated Web Client or the Siebel Mobile Web Client. For a summary, see "Configuring the Siebel Web Client to Handle Server Requests" on page 31.	Siebel Web Clients	Siebel System Administrator
(8)	Run DB object installation scripts. For instructions, see "Running Scripts" on page 31.	Oracle Database Server	Oracle Applications DBA

These tasks are discussed in the following sections.

Preliminary Requirements and Activities

Before the Siebel Connector for Oracle Applications is installed, during the Siebel Server installation, your system must include the hardware and software listed below.

Systems

The connector setup is performed on the following systems:

- Siebel Enterprise Server
- Siebel Dedicated Web Client or Siebel Mobile Web Client (if you plan to deploy these types of client)
- Oracle Applications Server

Software

The DBA or System Administrator must be certain that the following software is properly installed and configured before the Connector installation starts:

- Oracle Applications (on the Oracle Database Server)
- Oracle Client latest version (on the Siebel Server)

Oracle SQL*Net database connectivity software is installed during the client software installation. SQL*Net enables database messaging over a variety of network protocols.

- Siebel Web Client software (on the Siebel Mobile Web Client or Siebel Dedicated Web Client if you plan to deploy these types of client)
- ODBC drivers (on the Siebel Server)

For details of the versions of the software, see the *Siebel System Requirements and Supported Platforms* on Siebel SupportWeb.

See the *Siebel Installation Guide* for the operating system you are using for directions on installing and configuring the Siebel Server, Siebel Gateway Name Server, and Siebel Web Client software. For descriptions of Oracle Applications installations, refer to the current Oracle Applications installation guide.

Installing and Configuring Oracle Applications

Install Oracle Applications (English settings) on the Oracle Applications Server running Oracle/ Windows. If you are running a version of Oracle that is not supported for use with Siebel Connector for Oracle Applications, or Oracle in a different language or platform, ask your systems integrator for assistance in configuring your implementation. In addition, your Oracle system must include the latest database release, Oracle hot patches, and the Oracle kernel. For more information about these steps, see *Siebel Installation Guide* for the operating system you are using.

Enabling Account Outbound Integration

You must set up the conditions listed below before using the Account Outbound integration.

- Add the Lookup Code GENERAL under the lookup type SITE_USE_CODE on the Oracle Side in the following screen:
  - \Navigator\Setup\System\QuickCode\Receivables

  The HQ site used for an address on the Siebel side will be mapped into the site use code GENERAL on the Oracle side.
- Set Automatic Site Numbering to Yes in the following screen in the Oracle application:
  - \Navigator\Setup\Customers\System Options\Trans and Customers

    The flag is set in the Trans and Customers Tab.

  Next, click on \Zone\Next to go to the zone Customers. Then, set Automatic Site Numbering to Yes and save the screen. If it is not set to Yes, Location becomes a required column that is based on the Location. Site numbering is performed automatically.
- County is a required column in the Sales Tax Location Flexfield. However, it is not a mandatory column in the Siebel database and is also not displayed in the User Interface (UI). For the Oracle Import program to pick up the County automatically for a particular state and city, it must be set up in the following screen. Otherwise, the Siebel application must pass the value for County in the following application:

- Navigator > Setup > Tax > Location

- For Address Validation problem setup, navigate to Receivables > Navigate > Setup > System > System Options >Tax. Modify the Location Flexfield Structure. State County City is the default value. Change to State City, then SAVE.

Enabling Order Outbound Integration

In order to make the Accounts Receivable and Orders sent from Siebel Applications visible in Oracle Applications, you must perform the following setup before any user enters an Outbound Order.

To enable an outbound integration

1. Create an Order Source Siebel (Siebel should be mixed case) in the following screen in the Oracle application:
   - Order Source is created in `\Navigator\Setup\Orders\Import Sources`
2. Set up the Operating Unit for both Receivables Super User and Order Entry Super User.
   a. Login as the System Administrator.
   b. Go to \Navigate\Profile\System.
   c. Select Responsibility as the level.
   d. For the Name, enter the name that will be used for Order Entry and Accounts Receivable.
   e. Check to be certain that MO: Operating Unit is not null and is the same for both the System Administrator and the Name that will be used for Order Entry and Accounts Receivable.

Installing the Mobile Web Client Application

To install the Mobile Web Client, follow the instructions provided in the *Siebel Installation Guide* for the operating system you are using.

Verifying Preliminary Tasks

After the preliminary setup, check the SQL*Net database connectivity and network connectivity. These tasks are discussed below.

SQL*Net Database Connectivity

Follow the directions in the Oracle documentation to verify the Oracle SQL*Net software installation. For additional information, see the *Siebel Installation Guide for Microsoft Windows*.

Network Connectivity

The *Siebel Installation Guide for Microsoft Windows* explains how to verify network connectivity among the Siebel Servers, Gateway/Name Server, Database Server, and the File System.

Configuring the Connector

Next, you can configure the connector. These activities are described in order below.

Creating the Oracle Database Login

After verifying the preliminary setups, start the connector installation by creating an Oracle Database login (named SIEBEL) on the Oracle Applications database. The Connector creates new objects (procedures, views and database triggers) in the Oracle Applications database using this user ID.

Create an Oracle Database User SIEBEL with the following system privileges:

- CREATE SESSION
- RESOURCE
- CREATE SYNONYM
- CREATE SEQUENCE
- CREATE TABLE
- CREATE PROCEDURE
- CREATE VIEW

Create Oracle Applications User ID

You should also create an Oracle Application user named SIEBEL for the Connector. This User ID is used to populate the created by and updated by columns of the open interface tables.

When the connector is being used with Oracle Applications, the SIEBEL user should have at least one responsibility defined for each operating unit in which orders are created using Siebel Applications. These responsibilities enable the connector to pass an Order outbound to call the Process Order API. For more information, see Oracle Applications documentation.

Default User Names and Passwords

Unless you change them in the configuration file, the connector will be installed with the default user name and password shown in Table 4.

Table 4. Default User Names and Passwords

Default User Name	Default Password
apps	apps

NOTE: The connector uses the user name and password to connect to Oracle Applications to exchange data.

Installing the Oracle Client

Next, the DBA or System Administrator installs Oracle client software on the Siebel Server and configures connectivity to the Oracle Applications instance. For directions, see the Oracle Client software installation guide.

Installing and Configuring the Siebel Enterprise Server and Web Clients

In the third step, you install and configure the Siebel Enterprise Server and Web Clients. These activities are summarized below. For details, see the *Siebel Installation Guide* for the operating system you are using and the *Siebel System Administration Guide*.

During the installation, the various components of the Siebel Enterprise Server software and the Siebel Connector for Oracle Applications are installed on the Siebel Enterprise Server.

Unpacking SQL and PL/SQL Scripts

SQL and PL/SQL scripts, which will be used in a later installation step, are packaged with the Siebel Enterprise Server software. You need to unpack them to the Siebel Server and then move these scripts to the Oracle Applications Server in order to install them.

To unpack SQL and PL/SQL scripts

1. Start installing the Siebel Enterprise Server.

   For directions, see the *Siebel Installation Guide* for the operating system you are using.

2. When the Select Components dialog window appears, select Siebel EAI Connectors; underneath, make sure that you install the Oracle Connector, and then click Next.

   The Siebel EAI Connectors selection unpacks SQL and PL/SQL scripts.

3. Select Typical install and click Next. Follow instructions on the remaining dialog box to unpack the scripts.

Verifying the Unpacking of SQL and PL/SQL Scripts

You may want to verify the unpacking of the SQL and PL/SQL scripts.

To verify

1. Open the directory where the Siebel Server is installed.
2. Open the eaiconn folder.
3. Open the ORACLE folder.

   The R11i Scripts folder is shown.

4 Open the folder 11ISCRIPTS to examine Oracle Applications scripts.

5 Scroll down the list, comparing it with the list of Oracle Scripts in Appendix B, “Oracle Scripts.”

Registering the ODBC Driver

The ODBC Driver for accessing the Oracle Applications database is packaged with the Siebel Enterprise Server software. You must register it with Microsoft Windows before it can be used by the connector to communicate with Oracle Applications. Use the Siebel Software Configuration Utility to perform this task.

To register the ODBC Driver

1 Start the Siebel Software Configuration Utility. The utility is automatically launched at the conclusion of the Siebel Server install. You may also start it manually by selecting Microsoft Windows Start Menu > Programs > Siebel Enterprise Server > Configure Siebel Server.

2 When the dialog box Configure EAI Connectors: Register External Oracle DB ODBC Driver appears, check Yes under the Register External Oracle DB ODBC Driver prompt.

NOTE: If you are manually registering the ODBC driver, you must stop the Siebel Service before registering the driver and restart the service after registering the driver.

Verifying the Registration of the ODBC Driver

After the connector is installed, you should verify the ODBC driver installation.

To verify

1 Open the ODBC Data Source Administrator and select the Drivers tab.

2 Look for Siebel Oracle90 in the Name column on the left.

Creating the Net Service Name

To enable you to access an Oracle database across a network, you can create a net service name using Oracle Net Manager, which is part of Oracle Net Service. The main feature of Oracle Net Manager that you use is the Net Service Name Wizard.

The example that follows illustrates how to create a service name for an Oracle database to be accessed across a TCP/IP connection.

To create a net service name

1 From the Windows Programs window, navigate the path: Oracle 9i > Configuration and Migration > Net Manager.

2 In the Net Manager window, select Oracle Net Configuration > Local > Service Naming.

This launches the Net Service Name Wizard.

3 As you progress through each of the first 4 pages of the wizard, enter or select the appropriate data, then click Next:

- On page 1, specify the Net Service Name.
- On Page 2, select the Network Protocol, as for example, TCP/IP (Internet Protocol).
- On Page 3, if you previously selected the TCP/IP protocol, enter the TCP/IP host where the database is located. You can also change the default Port Number.
- On Page 4, you must provide either the Service Name (for Oracle 8i or later), or, for Oracle 8.0, a system identifier (SID).

4 On Page 5, you can optionally click Test to verify that the database can be reached using the data that you have provided. If you successfully access the database, you can change the user name and password for the test in the Change Login window.

5 Click Finish to complete the creation of the net service name.

Creating the ODBC Data Source

Next, you create an ODBC data source on the Siebel Server.

To create an ODBC data source

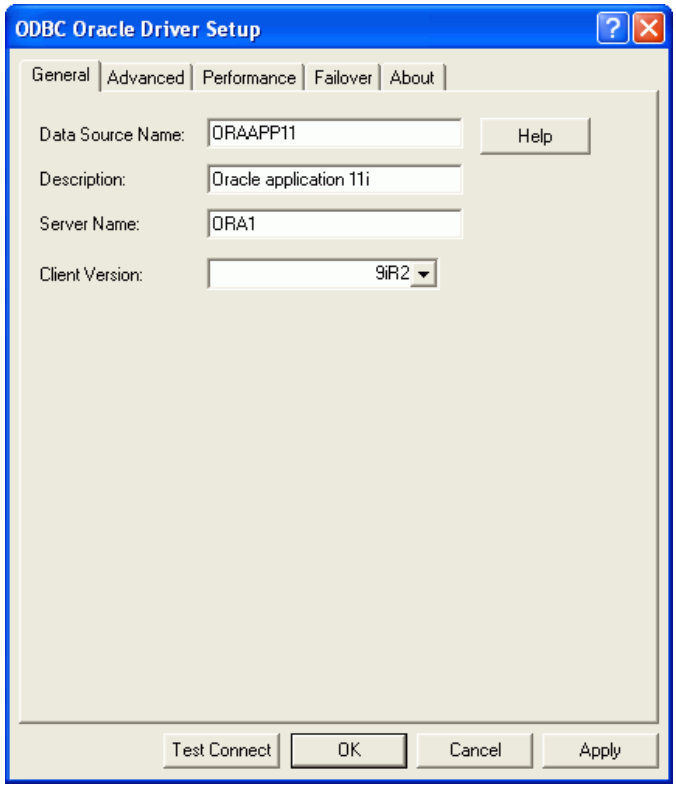
1 In the Microsoft Windows ODBC Data Source Administrator window, select the System DSN tab. Click Add.

The Create New Data Source window appears.

2 Select the ODBC driver to be added and click Finish.

Depending on which driver you select, there may be slight differences in the setup windows that appear.

3 If the ODBC Oracle Driver Setup window appears, fill in the Data Source Name, Description, and Server Name, select the Client Version, and click OK. For example:



If the Microsoft ODBC for Oracle Setup window appears, fill in the Data Source Name, Description, User Name, Server Name, and click OK.

The Oracle data source is created.

To test the data source connection

1 If you entered data in the ODBC Oracle Driver Setup window, you can now test your connection to the data source. Choose Test Connect.

2 When the Logon dialog window appears, enter the Server Name, user name, and password for logging onto Oracle Applications. Then click OK.

You should be connected to the Oracle database.

Configuring the Oracle Receiver

The Oracle Receiver is configured so that it can detect the Oracle RDBMS. This task is summarized below.

To configure the Oracle Receiver

1 Navigate to the Administration - Server Configuration screen > Enterprise Servers > Component definitions view.

2 Select Oracle Receiver and search for Component Parameters.

3 Supply the Current Value and Value on Restart for the following Oracle parameters:

- `ExtDBODBC Data Source`

  This is the ODBC data source name for the Oracle database.

- `ExtDBPassword`

  This is the password for the data source.

- `ExtDBTableOwner`

  The is the Table owner for the DSN.

- `ExtDBUserName`

  This is the Username for the data source.

- `ORCL Batch Size`

  If this parameter is specified, the Oracle Receiver will extract (in a single pass) the maximum number of records from the Notification Table, up to ORCLBatchSize.

NOTE: Current Value and Value on Restart may have the same value.

Starting the Oracle Receiver

You may start the Oracle Receiver immediately after you install and configure the Oracle Receiver. Once it is running, the Receiver continues unless it is shut down by a system or power failure. In this case, you must restart the Oracle Receiver as part of the general restarting process.

Guidelines for Starting the Oracle Receiver

The following guidelines apply to starting the Oracle Receiver:

- You can start the Oracle Receiver in DOS through the Server Manager.
- The Oracle Receiver server component name is ORCLRcvr.
- The following example shows the parameters to start the Oracle Receiver:

```
srvrmgr > start task for component ORCLRcvr /g GatewayName /e EnterpriseName /s
ServerName /U LoginName /P Pswd
```

Configuring the Business Integration Manager

Next, configure the Business Integration Manager (BIM). Here also, you are configuring parameters so that the BIM can detect the Oracle RDBMS. This task is nearly identical to configuring the Oracle Receiver.

To configure the BIM

1. Navigate to the Administration - Server Configuration screen > Enterprises > Component definitions view.
2. Select Business Integration Manager and search for Component Parameters.
3. Supply the Current Value and Value on Restart for the following Oracle parameters:
   - ExtDBODBCDataSource
   - ExtDBPassword
   - ExtDBTableOwner
   - ExtDBUserName

Activating Workflows

Next, activate the workflows that connect each data flow in the Connector. This operation is outlined below.

To review the predefined workflows, see “Siebel and Oracle Applications Integrations” on page 56 and “Understanding the Database Adapter” on page 68.

To activate workflows for the connector

1. Navigate to the Administration - Business Process screen > Workflow Deployment > Repository Workflow Processes view.
2. Query for all Oracle workflows: enter *Oracle*11i* in the Name search field.
3. Select each workflow returned by the query, and click Activate.
4. Verify all the activated workflows in the Active Workflow Processes applet.

Configuring the Siebel Web Client to Handle Server Requests

If you plan to deploy the Siebel Mobile Web Client or Siebel Dedicated Web Client, you need to enable the client to handle Server requests by changing parameters in the application configuration file. Refer to *Siebel System Administration Guide* for detailed instructions.

Running Scripts

Next, run the install scripts to create database tables and load the SQL and PL/SQL scripts that you unpacked earlier onto the Oracle Applications Server. For this procedure, you must already have a SIEBEL user account with the required system privileges. The steps for creating this Oracle user ID are described in "Creating the Oracle Database Login" on page 24.

The Siebel Connector for Oracle Applications uses the APPS User ID to connect to the Oracle Applications database (to send and receive data); access rights on Connector objects are granted to APPS database users.

To run SQL and PL/SQL scripts

You run Oracle setup scripts to grant necessary privileges, create synonyms, create all Connector objects, and create alerts and triggers under appropriate database user accounts. All supplied Connector objects are to be created under the SIEBEL user. Necessary privileges on these objects are granted to the APPS user. All Siebel Connector for Oracle Applications alerts and triggers are created under the APPS user.

To set up Siebel Connector for Oracle Applications Objects in Oracle Applications

1. Log into the Oracle Applications database using Oracle SQL*Plus as the APPS user.
2. Run the script siebelgeneral11isetup.sql at the prompt.
3. At the prompt for the SIEBEL user, enter SIEBEL.
4. At the password prompt, enter the password for this SIEBEL user.
5. At the connectstring prompt, enter the connect string for your Oracle Applications database. It connects to the database as the SIEBEL user and runs the siebelgeneral11isetup2.sql script.

6 At the prompt to log in again, log in as the super user APPS.

7 At the password prompt, enter the password for the APPS user.

8 At the connect string prompt, enter the connect string for your Oracle database. You are connected to the database as the APPS user and run the script siebelgeneral11isetup3.sql'.

The setup scripts for Oracle Applications are available under \eaiconn\ora\orascripts\oracle11i\.

Post-Installation Oracle Applications Alerts Setup

After the Oracle Applications installation, the Oracle Applications Administrator should enable all Connector Alerts in every operating unit that will supply data to the Siebel application. For more information, refer to the Oracle Alert User's Guide.

4 Setting Up the Connector

This chapter explains how to set up the Siebel Connector for Oracle Applications after installation. It provides directions for configuring Siebel Applications and Oracle Applications EAI value maps, loading Oracle Applications data, loading payment terms, and verifying the data load. These activities are listed in Table 5.

Table 5. Setup Activities

Step	Activity and Discussion	Machine	Responsible Person
(1)	Configure EAI value maps. For directions, see "Configure EAI Value Maps" on page 33.	Siebel Client machine	Siebel Administrator and Business Analysts
(2)	Load Oracle Applications data. For directions, see "Loading Oracle Applications Data" on page 34.	Siebel Server	Siebel Administrator
(3)	Populate the Integration ID column. For directions, see "Loading Payment Terms" on page 36.	Siebel Server	Siebel Administrator
(4)	Verify the Oracle data load. For directions, see "Verifying the Oracle Data Load" on page 36.	Siebel Server	Siebel Administrator and Business Analyst

Configure EAI Value Maps

Working together, the Siebel Administrator and Business Analysts must configure the Siebel Applications to Oracle Applications EAI value maps. The connector provides a set of predefined maps that correlate Siebel data values with Oracle Applications data values. The EAI Value Map entries are stored in the EAI Value Map table. You may modify these maps according to your specific business needs or define entirely new maps.

Creating EAI Value Maps

This section explains how to create an EAI Value Map. For more information, see *Business Processes and Rules: Siebel Enterprise Application Integration*.

To create EAI value mappings

1. Navigate to the Administration - Integration screen > EAI Value Maps > EAI Lookup Map view.
2. Click New.
3. Select Direction (Siebel Inbound or Siebel Outbound) and Type from the pick list.

   Type is pre-defined in the system. If a new type needs to be added, you must add it to the List of Values view under Applications Administration.
4. Depending on the Direction and Type selected, enter the Siebel Value, External System Value. Enter any comments (if necessary) and click Save.

   NOTE: For more information about EAI Value Maps, see *Business Processes and Rules: Siebel Enterprise Application Integration*.

Configuring EAI Value Maps

Out of the box, the Oracle Connector provides a set of EAI value maps that translate some Oracle Applications lookup values to Siebel lookup values and the reverse. "Oracle Scripts" provides a complete list of these EAI value maps that are provided with the Oracle Connector. When the Connector is set up, Siebel and Oracle Applications Business Analysts should verify that these mappings accurately reflect their business processes and correct the existing values or add new values if necessary.

Loading Oracle Applications Data

After the Siebel Connector for Oracle Applications is installed, the System Administrator must load Oracle Applications data into the Siebel database prior to using Siebel Business Applications to capture orders. Follow the following sequence for the data load.

To load Oracle Applications data

1. Load Operating Units definitions into Siebel Applications as Organizations.
2. Load Inventory Organization definitions into Siebel Applications as Inventory Locations.
3. Load Item definitions into Siebel Applications as Products.
4. Load existing Customer data into Siebel Applications as Accounts.
5. Load existing Order data into Siebel Applications as Orders.

Again, these tasks involve running a SQL script to register data from Oracle Applications' base tables with the Notification Table and starting the Oracle Receiver to receive the data. These activities are described in the next two sections.

Loading Oracle Applications Data

This section explains how to initially load data from an Oracle Applications database to the Siebel database. It provides step-by-step directions for creating Siebel organizations and inventory locations and loading Oracle product and account data from Oracle Applications. Each task involves running a SQL script and starting the Oracle Receiver to load the data. These activities are described next.

Loading Operating Units and Inventory Organization from Oracle Applications

When you initially load data from Oracle Applications to Siebel Applications, you must first load Operating Unit definitions and then load Inventory Organization definitions in order to maintain the referential integrity of other data that you will be loading. These activities are summarized below.

To load Oracle Operating Units into Siebel Applications

1 At the SQL prompt, run the script siebelgeneral11iscript_notify_org.sql.

This script inserts records into the Notification Table identifying all Oracle Operating Units.

2 Start the Oracle Receiver.

The Oracle Receiver automatically invokes the Oracle Receiver Dispatch Workflow which calls sub-process Organization - Receive Oracle11i Operating Unit and loads all Oracle Organizations into Siebel Applications. The workflow for this process is described in "Organization Inbound Workflow" on page 57.

To load Oracle Inventory Organizations into Siebel Applications

1 At the SQL prompt, run the script siebelgeneral11iscript_notify_invloc.sql.

This script inserts records into the Notification Table identifying all Oracle Inventory Organizations.

2 Start the Oracle Receiver.

The Oracle Receiver automatically invokes the Oracle Receiver Dispatch Workflow which calls sub-process InvLoc - Receive Oracle11i Organization and loads all Oracle Inventory Organizations into the Siebel application. The workflow for this process is described in "Inventory Locations Inbound Workflow" on page 58.

Loading Pricing and Discounts Definitions from Oracle Applications

Pricing and Discount integrations are not provided with Oracle Applications at this time. You can define custom integration points to achieve integration, or use the Siebel Enterprise Integration Manager to load the data. For more information, see *Siebel Enterprise Integration Manager Administration Guide*.

Loading Items from Oracle Applications

After Oracle Applications' Operating Units and Inventory Locations are loaded, you can load Item definitions from Oracle Applications. This process is summarized below.

To load Item Definitions

1 At the SQL prompt, run this script:

```
siebelgeneral11iscript_notify_product.sql
```

This script inserts records into the Notification Table for all Oracle Product Items.

2 Start the Oracle Receiver.

The Oracle Receiver invokes the Oracle Receiver Dispatch Workflow which calls sub-process Product - Receive Oracle11i Item and loads all Products into Siebel Applications. The workflow for this process is described in "Product Inbound Workflow" on page 59.

Loading Customer Data from Oracle Applications

Next, existing customer data is loaded from Oracle Applications into Siebel Applications. Use the Siebel Enterprise Integration Manager to load the data. The Integration Manager is described in *Siebel Enterprise Integration Manager Administration Guide*.

Loading Orders from Oracle Applications

Finally, existing order data is loaded from the Oracle Applications into Siebel Applications. Here again, the Siebel Enterprise Integration Manager is used to load the data. The Integration Manager is described in *Siebel Enterprise Integration Manager Administration Guide*.

Loading Payment Terms

There is no inbound dataflow for loading payment terms into Siebel Applications. Before you can use Siebel Connector for Oracle Applications, you must populate the Integration ID column in the Siebel payment terms (S_PAYMENT_TERM) table with the value of the TERM_ID column in Oracle Applications table RA_TERMS. Use the Enterprise Integration Manager (EIM) for this setup task.

Verifying the Oracle Data Load

After loading Oracle data into the Siebel database, you can verify the data load by comparing the Siebel data with the Oracle data. See *Siebel Applications Administration Guide* and Oracle Applications user guides to determine which reports can be used to compare Organization, Account, Product, and Order data.

5 System Administrator Procedures

The System Administrator is responsible for maintaining and monitoring Connector operations. This chapter explains how to perform the typical administrative tasks listed in Table 6.

Table 6. Administration Activities

Activity	Machine	Responsible Person
Monitoring outbound data integrations. See "Monitoring Outbound Integrations" on page 37.	Siebel Server	Siebel Administrator and Oracle Systems Administrator
Monitoring inbound data integrations. See "Monitoring Inbound Integrations" on page 38.	Siebel Server	Siebel Administrator and Oracle Systems Administrator
Troubleshooting messages. See "Troubleshooting Oracle Connector Messages" on page 39.	Siebel Server	System Administrator

Monitoring Integrations

You can monitor integrations of Siebel Applications with Oracle Applications. Follow the steps in the next section.

Monitoring Outbound Integrations

The Oracle Administrator and the end user can monitor outbound integrations. These activities are summarized below.

Monitoring Account to Oracle Applications Integration

The Oracle Administrator can monitor outbound integrations. The following approach is used to monitor the user-initiated outbound integration:

- Siebel Account business object with Oracle Applications.

The following activity takes place:

1 The user submits the Account to Oracle Applications through the Siebel User Interface (for example, Sales window). The data is entered into Oracle Applications interface tables.

   If a failure occurs, the user is immediately informed by an error message that indicates the cause of the failure.

2 The Oracle Import program runs to import the data into Oracle Applications.

   If a failure occurs, the failed rows are marked with the error status by the import program. For more information, see the appropriate Oracle Open Interface manual.

To monitor outbound integrations

- Monitor the Oracle interface tables for failures.

  When appropriate, take corrective actions. For more information, refer to the Oracle Applications documentation set.

Monitoring Order to Oracle Applications Integration

The user monitors a Siebel Order to Oracle Applications integration by examining any error messages that appear after the order is submitted.

Monitoring Inbound Integrations

The Siebel Administrator can monitor any inbound integrations from Oracle Applications to Siebel Applications.

The Oracle Receiver Server Task performs the Inbound integration. Because the Task is driven by the Siebel Notification Table (SIE_NOTIFY_TBL), you can monitor the integration by running a SQL query against the Notification Table.

For example, the following query will yield a list of current entries in the Notification Table:

```
select seq_num, status, type, object_name, tbl_name, PRI_KEY1, PRI_KEY2,
original_system_reference, operation

from sie_notify_tbl order by seq_num;
```

For a high level overview, the following query will yield a summary of the entries in the Notification Table:

```
select object_name, tbl_name, status, count(*) from sie_notify_tbl

group by object_name, tbl_name, status;
```

Entries waiting to be processed have STATUS='NEW'. When a notification entry is successfully completed, the entry is removed from the Notification Table.

If an error occurs while an entry or set of entries is processing, the entries are marked with STATUS='ERROR'. In this case, the Oracle Administrator can read the Oracle Receiver Task Log to learn why the process failed. Indicative trace files can be found in the log directory of the Siebel Server installation.

Troubleshooting Oracle Connector Messages

You can troubleshoot Siebel Applications-to/from-Oracle Applications integration problems by reviewing the messages displayed by the Siebel and Oracle applications. The Siebel messages are discussed in Table 7. For a description of the Oracle messages, refer to Oracle documentation.

Table 7. Siebel Connector for Oracle Applications Messages

Message	Meaning	Action
<value> not in value map <value>	Missing EAI Value Map.	Set up the value in EAI value map.
Contact First Name is NULL.	First name of the Contact sent from Oracle is null.	Enter Contact with a first name.
Delete not supported on <value>.	Deletion for this object not supported.	Contact System Administrator.
Discount Id is NULL.	Missing Discount.	Enter discount on Siebel side.
Empty Input Message <value>.	There is no Integration Object in the input message.	Contact System Administrator.
Failed to insert customer.	Insert into RA_CUSTOMERS_INTERFACE_ALL fails.	Contact System Administrator.
From Order Process API <value>	This error message is returned by Process Order API (Oracle Applications).	Please refer to the Oracle Applications reference manuals for further information.
Header Reference is NULL.	Missing Order Header	Contact System Administrator.
Interface Status Error.	This has been submitted already to Oracle but has an error in the import process.	Correct the error in the Oracle Applications interface table and submit the import program again.
Invalid Bill To Contact <value>.	This Bill To Contact does not exist in Oracle Applications.	Submit this Account/Address/ Contact to Oracle Applications before submitting the order.
Invalid Contact Title <value>.	Contact Title not defined in (Oracle Applications) AR LOOKUPS.	Define Contact Title in Oracle Applications.
Invalid Currency <value>.	The Currency is not defined in Oracle Applications.	Define the Currency in Oracle Applications.
Invalid Cust Int. Id <value>.	Invalid Customer Integration Id	Contact System Administrator.

Table 7. Siebel Connector for Oracle Applications Messages

Message	Meaning	Action
Invalid Customer Status <value>.	The Customer Status is invalid.	Select Customer Status on the Siebel side for this account.
Invalid Discount Line Id <value>.	Discount Line is missing.	Select Discount Line in Siebel.
Invalid Insert/Update <value>.	Invalid insert/Update Flag	Contact System Administrator.
Invalid InvLoc <value> : Item <value>.	This item is enabled in the warehouse.	Select a valid item for this order.
Invalid Invoice Contact <value>.	The Invoice to Contact does not exist in Oracle Applications.	Submit this Contact to Oracle Applications before submitting the order.
Invalid Order Source Id <value>.	SIEBEL is not defined as Order Source in Oracle Applications.	Setup SIEBEL order source in Oracle Applications.
Invalid Order Type <value>.	Invalid Order Type	Set up the value in EAI value map.
Invalid Payment Term <value>.	The Payment Term does not exist in Oracle Applications.	Select a Payment Term that exists in Oracle Applications or complete integration for that payment term.
Invalid Sales Rep Id <value>.	Missing Sales Rep Id for a Booked Order.	Set up the value in EAI value map.
Invalid Ship FOB Code <value>.	The FOB Code is invalid.	Set up the value in EAI value map.
Invalid Ship Prior. Code <value>.	The Ship Priority Code is invalid.	Set up the value in EAI value map.
Invalid Ship To Contact <value>.	Ship To Contact does not exist in Oracle Applications.	Create the Ship To Contact.
Invalid Site Use Code <value>.	Site Use Code not defined in (Oracle Applications) AR LOOKUPS.	Define Site Use Code in Oracle Applications.
Invalid Tax Exem. Reason <value>.	Tax exempt Reason is invalid.	Set up the value in EAI value map.
Invalid User Id <value>.	Siebel user is not defined in Oracle Applications.	Define Siebel user in Oracle Applications.
Multiple Cust. Entries <value>.	More than one record exists for this Original system reference in RA_CUSTOMERS.	Contact System Administrator.
Order does not have a Line.	No line items in Order.	Create line Items for the Order.

Table 7. Siebel Connector for Oracle Applications Messages

Message	Meaning	Action
Sales Rep Id is NULL.	Sales Rep Id cannot be null for a booked order.	Set up the EAI value map for Sales Rep.
Tax Exempt Number is NULL.	Tax exempt number is Null.	Enter Tax exempt number.
Unspecified Site Use Code <value>.	Valid site use code values are BILL_TO, SHIP_TO, GENERAL.	Contact System Administrator.
User Id, Resp. Mismatch<value>.	No responsibility is associated with the operating unit for SIEBEL user.	For SIEBEL user, associate at least one responsibility whose ORG_ID profile value matches the id of the operating unit.

6 User Procedures

This chapter explains how to create, update, and verify Order and Account activity. These everyday activities are usually performed by a Siebel Applications user (for example, Sales Representative).

Table 8. End User Activities

Activity and Discussion	Machine	Responsible Person
Create an account. See “Creating a New Account” on page 43.	Siebel Client machine	Siebel user
Update an account. See “Updating an Account” on page 44.	Siebel Client machine	Siebel user
Create an order. For directions, see “Creating an Order” on page 45.	Siebel Client machine	Siebel user (Sales, and so on)
Update an order. See “Updating an Order” on page 45.	Siebel Client machine	Siebel user
Verify an order. See “Verifying an Order Status” on page 46.	Siebel Client machine	Siebel user

Working with Accounts

Using a Siebel Applications application (for example, Sales), you can create a new account or update an existing account and send it to the Oracle application. As you create or change an account or import customer account data from an Oracle application, your input starts an integration process. This section explains how to work with Accounts. For descriptions of the associated workflows, refer to Chapter 7, “Developer Procedures,”

Creating a New Account

To create an account in Siebel Applications and send it to Oracle Applications

1. Using a Siebel application (for example, Call Center), click the Accounts tab.
2. Select My Accounts in the pull down box.

3 Click New to create a new account.

4 When the New Accounts display appears, enter the values for each applicable field.

5 Click Save to save the changes.

6 Enter additional details such as Addresses.

7 Select the Back Office tab for Oracle Applications.

8 Click Update Back Office to send the account record to the back office.

9 In a moment, the screen displays the synchronization status and indicates that the account was received in the back office.

Updating an Account

To update an existing account in Siebel Applications and send it to Oracle Applications

1 Using a Siebel application (for example, Call Center), click on Accounts.

2 Select My Accounts in the pull down box.

3 Click on a listed account.

4 Enter new values for account fields.

5 Click Save to save the account record.

6 Make any additional necessary changes to the account.

7 Select the Back Office tab for Oracle Applications.

8 Click Update Back Office to send the account record to the back office.

9 When the screen is refreshed, it displays the synchronization status and indicates that the account was received in the back office.

The Oracle Receiver automatically brings all new and updated account records created through Oracle Applications to Siebel Applications at the next scheduled run.

Working with Orders

Using a Siebel Applications application such as Sales, you can create an order or update an existing order, send it to Oracle Applications, and verify the order. These user activities are explained in the following sections. For descriptions of the associated workflows, refer to Chapter 7, “Developer Procedures,”

Creating an Order

You can create a new order in Siebel Applications, send it to Oracle Applications, and verify the integration.

To create an order in Siebel Applications and send it to Oracle Applications

1. Using a Siebel application (for example, Sales), click the Sales Orders tab.
2. Select All Sales Orders in the pull down box.
3. Click on the Sales Order tab.
4. Click New and enter order details.
5. Under the Line Item tab, click New at the bottom, and enter line item details.
6. Click Save to save the order details.
7. Click on the Back Office tab for Oracle Applications to display more information.
8. Click Update Back Office to send the order record to the Oracle back office application.
9. At any point, if you want to check the status of the order in the back office, click Get Status.

Updating an Order

You can update an existing order using a Siebel application such as Sales.

NOTE: You can submit an order only once.

To update an existing order in Siebel Applications and send it to Oracle Applications

1. Using a Siebel application (for example, Call Center), click the Sales Orders tab.
2. Select All Sales Orders in the pull down box.
3. Click on the Sales Order tab.
4. Select an existing order.
5. Click Edit and enter changes to the order.
6. Click Save to save your changes.
7. Click on the Back Office tab for Oracle Applications to display more information.
8. Click New and enter changes to the Line Item details.
9. Click Save to save your changes.
10. Click the Update Back Office button to send the new order information to the Oracle back office application.

    After the order is sent to Oracle, it cannot be updated through a user interface (UI) in a Siebel Application.

11 At any point, if you want to check the status of the order in the back office, click Get Status.

Verifying an Order Status

You can use a Siebel application such as Sales to check the status of an order.

To verify an order status

The Oracle Receiver automatically imports any Oracle-generated order updates to Siebel Applications during its next scheduled run.

1 Open a Siebel application and select the Sales Order tab.

2 Click on the line displaying the particular order for details.

Understanding Integrations

For descriptions of the workflows controlling Siebel Applications and Oracle Applications integrations (Organization, Inventory Location, Account, Order, and Product integrations), see Chapter 7, “Developer Procedures,”

See “Detailed Data Mappings” for descriptions of associated EAIs and EAMs.

7 Developer Procedures

Designed for developers, this chapter provides summary descriptions of the Entity Attribute Maps (EAMs), workflows, and Siebel Enterprise Application Integration (EAI) Value Maps associated with the integration of Siebel Applications and Oracle Applications data. It also includes technical information about the DB Adapter, Oracle Receiver, and Oracle Procedure Adapter.

NOTE: The DB Adapter is also known as the EAI SQL Adapter, and the Oracle Procedure Adapter is also known as EAI ODBC Service.

Table 9 summarizes developer activities.

Table 9. Developer Activities

Activity	Machine	Responsible Person
Running the Oracle Receiver. For directions, see "Understanding the Oracle Receiver" on page 74.	Siebel Client machine	Siebel Administrator
Working with integration objects. See "Working with Integration Objects" on page 81.	Siebel Client machine	Developer
Running the Oracle Application Wizard. See "Running the Wizard" on page 82.	Siebel Client machine	Developer

Installing the Development Environment

You must install a development environment before you modify or extend integration points. Here are the installation steps:

To install the environment

1. Install the Oracle Client.
2. Install the Siebel Mobile Web Client.

   You will run the Siebel Workflow Simulator using this client in your development environment. See *Siebel Business Process Framework: Workflow Guide*.
3. Install Siebel Tools.

   For directions, see *Siebel Installation Guide* for the operating system you are using.
4. Set up the ODBC Data Source.

5 Edit the configuration files for the Siebel Mobile Web Client. See *Siebel Installation Guide* for the operating system you are using to get the name of the configuration file that you should edit.

6 Edit the configuration file for Siebel Tools (tools.cfg).

Installing the Oracle Client

Next, the DBA or System Administrator installs Oracle client software on the Siebel Server and configures connectivity to the Oracle Applications instance. For directions, see the Oracle Client software installation guide.

Installing Siebel Mobile Web Client and Siebel Tools

You must use Custom Install when you install Siebel Mobile Web Client and Siebel Tools when you set up a development environment for configuring and extending the connector. This procedure will direct the installers to register the ODBC Driver needed for accessing the Oracle Applications database with Microsoft Windows.

To install

1 Launch the installer.

2 Choose Custom Install.

3 Select ODBC Driver for External Oracle Database in the Select Components dialog.

After you make the selection, follow the normal procedures for installing Siebel Mobile Web Client and Siebel Tools.

Verifying the Registration of ODBC Drivers

After the connector is installed, you may want to verify the installation of ODBC drivers.

To verify

1 Open the ODBC Data Source Administrator and select the Drivers tab. Look for Siebel Oracle90 in the Name column on the left.

Creating the Net Service Name

To enable you to access an Oracle database across a network, you can create a net service name using Oracle Net Manager, which is part of Oracle Net Service. The main feature of Oracle Net Manager that you use is the Net Service Name Wizard.

The example that follows illustrates how to create a service name for an Oracle database to be accessed across a TCP/IP connection.

To create a net service name

1. From the Windows Programs window, navigate the path: Oracle 9i > Configuration and Migration > Net Manager.
2. In the Net Manager window, select Oracle Net Configuration > Local > Service Naming.

   This launches the Net Service Name Wizard.
3. As you progress through each of the first 4 pages of the wizard, enter or select the appropriate data, then click Next:
   - On page 1, specify the Net Service Name.
   - On Page 2, select the Network Protocol, as for example, TCP/IP (Internet Protocol).
   - On Page 3, if you previously selected the TCP/IP protocol, enter the TCP/IP host where the database is located. You can also change the default Port Number.
   - On Page 4, you must provide either the Service Name (for Oracle 8i or later), or, for Oracle 8.0, a system identifier (SID).
4. On Page 5, you can optionally click Test to verify that the database can be reached using the data that you have provided. If you successfully access the database, you can change the user name and password for the test in the Change Login window.
5. Click Finish to complete the creation of the net service name.

Creating the ODBC Data Source

Next, you create an ODBC data source on the Siebel Server.

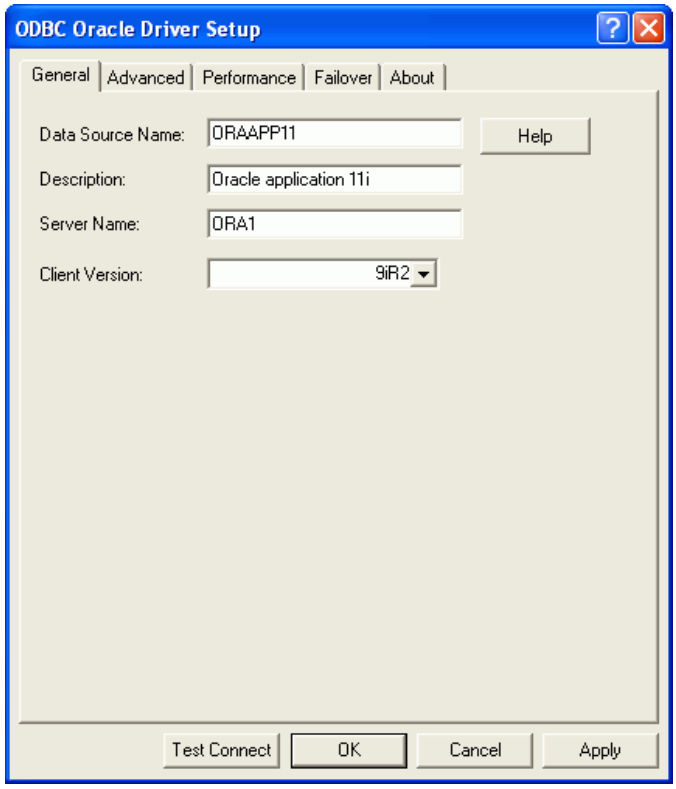
To create an ODBC data source

1. In the Microsoft Windows ODBC Data Source Administrator window, select the System DSN tab. Click Add.

   The Create New Data Source window appears.
2. Select the ODBC driver to be added and click Finish.

   Depending on which driver you select, there may be slight differences in the setup windows that appear.

3 If the ODBC Oracle Driver Setup window appears, fill in the Data Source Name, Description, and Server Name, select the Client Version, and click OK. For example:



If the Microsoft ODBC for Oracle Setup window appears, fill in the Data Source Name, Description, User Name, Server Name, and click OK.

The Oracle data source is created.

To test the data source connection

1 If you entered data in the ODBC Oracle Driver Setup window, you can now test your connection to the data source. Choose Test Connect.

2 When the Logon dialog window appears, enter the Server Name, user name, and password for logging onto Oracle Applications. Then click OK.

You should be connected to the Oracle database.

Editing the Configuration File for Siebel Mobile Web Client

To edit the configuration file

1 Get the name of the file that you need to edit, by referring to the *Siebel System Administration Guide*.

2 Supply the values for the following parameters under the [ExtDBSubSys] section.

- ExtDBUserName

  This is the Username for the data source.

- ExtDBPassword

  This is the password for the data source.

- ExtDBODBCDataSource

  This is the ODBC data source name for the Oracle database.

- ExtDBTableOwner

  This is the Table owner for the data source.

Editing the Configuration File for Siebel Tools

Supply the values for the following parameters under the [DBWizSubSys] section of tools.cfg.

- DBWizUserName

  This is the ODBC data source name for the Oracle database.

- DBWizPassword

  This is the password for the data source.

- `DBWizODBCDataSource`

  This is the Table owner for the data source.

- `DBWizTableOwner`

  This is the Username for the data source.

Integration Overview

This section provides background information about Siebel Applications and Oracle Applications integrations. If you are already familiar with these concepts, you may skip this section and go to more detailed discussions in "Siebel and Oracle Applications Integrations" on page 56.

General

Siebel Applications and Oracle Applications are integrated using Oracle's Siebel EAI infrastructure. Integration objects represent data exchanges, data is transformed by the Data Transformation Engine (DTE), and the entire process is coordinated by workflows. These key features are discussed below.

- **Integration Objects**

  The common data exchanged between Siebel Applications and Oracle Applications is depicted by integration objects. Oracle Applications data is represented by Oracle Integration Objects (also known as External Integration Objects) while Siebel application data is represented by Siebel Integration Objects (also known as Internal Integration Objects). For example, an Oracle Order integration object represents the structure of the Oracle table used to import orders in Oracle Applications. On the other hand, a Siebel Order integration object defines the data structures of the Order business components applied in an integration flow.

- **Data Transformation Engine**

  The Data Transformation Engine (DTE) handles the two-way transformation of integration objects. Siebel Data Transformation functions are used to construct transformation maps. Data transformation maps function as import and export filters. You can customize and extend the data transformation rules and define new transformation maps using the Siebel Tools Data Transformation Service to adapt the supplied predefined integration points or create new integration points.

- **Siebel Integration Workflow**

  Siebel Integration Workflows coordinate the processing steps involved in exchanging data between Siebel Applications and Oracle Applications. These workflow definitions include exception handlers to manage error conditions, ensuring a reliable data exchange. You can use the Siebel Workflow Designer to modify and extend workflows, modeling them more closely to actual business processes.

EAI Value Mappings

EAI value mappings associate Siebel Applications and Oracle Applications data values. For example, Siebel Account Status values are mapped to Oracle Customer Status values.

Stored in the EAI Value Map table, these can be viewed, created, and changed using the Siebel client Integration Administration screens. You can view the EAI value mappings shipped with the Connector or create new mappings using a Siebel application (for example, Sales) Integration Administration screen.

Predefined Integration Points

Oracle provides a set of predefined integration points (listed in Table 10 on page 53) with the Siebel Connector for Oracle Applications. You can use the Siebel EAI Integration Object Wizard to create additional points.

These predefined integration points promote seamless transactions across application boundaries by accessing a consistent set of data on customers, products, pricing, and discount rules. This allows sales and service professionals to generate quotes and place orders immediately, as they interact with customers using Siebel applications. For example, sales orders are easily transferred to the Oracle Applications back office where they are handled by Oracle Applications Order Entry and Manufacturing modules.

The Siebel Connector for Oracle Applications includes the following predefined integration points listed in Table 10.

Table 10. Synchronized Business Objects for Oracle Applications

Siebel Business Object	Oracle Applications R11i Data Object	Direction
Order	Order	Bi-directional data exchanges
Account	Customer	Bi-directional data exchanges
Product	Item	Data imported from Oracle
Organization	Operating Unit	Data imported from Oracle
Inventory Location	Inventory Organization	Data imported from Oracle

Integration Dependencies and Steps

The interdependencies and recommended sequence of data loads are shown in Figure 3.

NOTE: The diagram includes two dependencies, Price List and Discounts, that, while recommended for a complete solution, are not implemented with out-of-the-box Siebel Connector for Oracle Applications version 8.0.

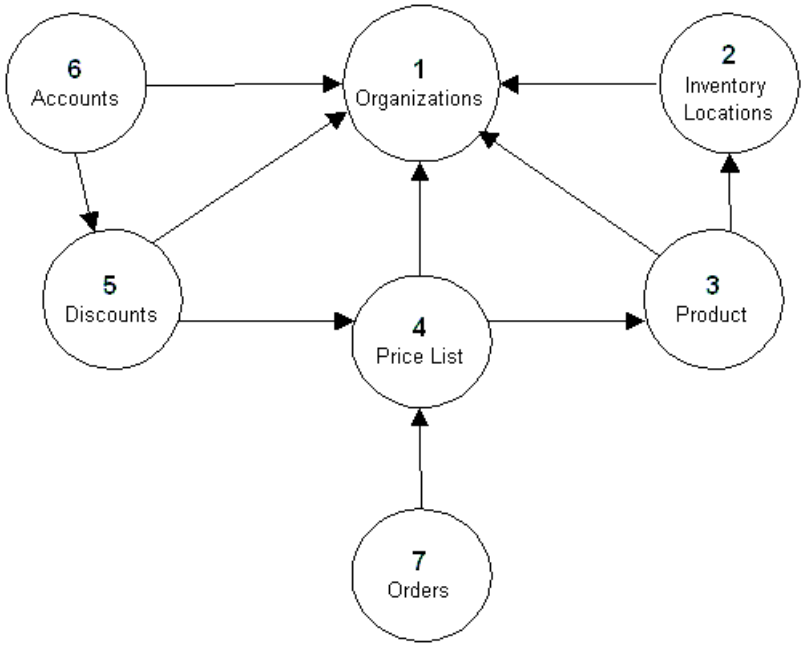


Figure 3. Integration Dependencies

Figure 3 shows the integrations initiated by the System Administrator and by Users. As the figure indicates, integrations should be run in the following order:

1. **Organizations.** This integration takes place during the setup, before all other integrations. During the initial setup, Organizations are created in Siebel Applications and loaded with Oracle Operating Unit data.
2. **Inventory Locations**. This integration take places during the Setup, well before the Product integration. During the initial setup, Inventory Locations are created in Siebel applications and loaded with Oracle Inventory Organizations data.
3. **Product.** This integration associates Siebel integration object Product - Get Oracle Item (Siebel Applications) with the Oracle Applications integration object Product - Get Oracle11i Item (Oracle Applications). Product integration (also known as Product Catalog Import):
   - Incorporates Oracle multi-organization support. Items are transferred from the Oracle inventory master organization. Associations with other Inventory Organizations are also created in Siebel Applications.

4 **Price List.**

If you perform this operation, you should integrate Oracle Price Lists, Discounts, and Price List Line Items into the Siebel Admin Price List business object.

For example, Price List integration:

- Should import an Oracle price list. For data consistency, Oracle Applications should be the Price List master. Changes to Price Lists should be performed in the Oracle application and brought into Siebel Applications through Connector processes.
- Should use a base unit-of-measure.

NOTE: The Price List integration is not implemented with out-of-the-box Siebel Connector for Oracle Applications version 8.0.

5 **Discounts.**

Discount data should be pulled from the Oracle Applications and integrated into Siebel Applications. This integration should associate Siebel business components with Oracle Discount information through the following mappings:

- Oracle Discount Lines to be mapped to the Siebel Volume Discount business component.
- Oracle Discounts Price Breaks to be mapped to the Siebel Volume Discount Item business component. These are Unit, not Monetary, Volume Price Breaks.

Generally, this integration (also known as Discount Rule Import):

- Should import Oracle Applications discount rules.
- Should create Discount Price List.

In Oracle Applications, a new Discount Line can be created for an item that exists under the same Discount but at a different time period. By contrast, in Siebel Applications there is a one-to-one relationship between a Price List item and a Discount. For this reason, only active Oracle Discount Lines should be pulled into a Siebel application as Volume Discounts.

NOTE: The Discounts integration is not implemented with out-of-the-box Siebel Connector for Oracle Applications version 8.0.

6 **Accounts.** This two-way integration maps Siebel Account-to-Oracle Customer data and Oracle Customer-to-Siebel Account data. See Chapter 6, "User Procedures," for information about the steps and workflow for this integration.

In summary, this integration:

- Starts with data captured by Siebel Applications. After that, changes are made at either side.
- Permits synchronization of customer contacts, telephone numbers, and addresses between the two applications.
- Maintains Oracle Applications address, customer, and site numbering hierarchy.
- Involves near real-time synchronization, triggered by update events at Siebel Applications and alerts at Oracle Applications.

7 **Orders.** Performed by Users, this is the final integration. All other integrations must precede it. This two-way integration maps Siebel Order-to-Oracle Order data and Oracle Order-to-Siebel Order Status data. Chapter 6, "User Procedures," describes the steps and workflow for this integration.

In summary, this integration (also referred to as Ordering and Order Status Update):

- Supports the creation of an order without a quote, deferring to Oracle Applications for price calculation or creation of an order with a quote using a price list imported from Oracle Applications.
- Allows tax and shipping costs to be estimated within the Siebel application. However, the final invoice figure is determined by Oracle Applications.
- Uses a back office applet to submit orders (using the Submit button) into Oracle Applications in near real-time. Orders can be submitted as Enter or Booked.
- Freezes an order in Siebel Applications once it is submitted. Changes are made only in Oracle Applications.
- Checks credit in Oracle Applications. The result of a credit check is reported in the Order Status field.
- Allows order line changes in Oracle Applications.
- Transmits order status and updates asynchronously back to the Siebel application at regular intervals, or transmits order status and updates synchronously on-demand in real-time using a Get Status button on the Siebel Applications back office applet.

Siebel and Oracle Applications Integrations

This section provides summary descriptions of the integrations that associate Siebel and Oracle Applications, focusing upon EAMs, workflows, and EAIs.

Entity Attribute Mappings (EAMs)

The following predefined Siebel and Oracle Applications integrations have EAM mappings:

- Organization—This mapping defines the propagation of Oracle Applications Operating Units data to the Siebel Internal Division business object.
- Inventory Location—This mapping defines the propagation of Oracle Applications Inventory Organizations data to the Siebel FS Inventory Location business component.
- Product—This mapping integrates Oracle Items and Siebel Internal Product business objects.
- Account—This mapping integrates the Siebel Account business object with the Oracle Customer.
- Order—This mapping integrates the Siebel Order business object with Oracle Order.

Entity Mapping Details—In each case, the Entity Mappings are displayed in a table containing values for the Siebel Business Component (for example, FS Inventory Location), associated Oracle Base Table (for example, hr_organization_units), and Oracle to Siebel Integration (for example, SIE_INVLOC_V).

Attribute Mapping Details—The Attribute Mappings table contains values for the associated Siebel Field (for example, Integration Id), Oracle Base Column (for example, ORGANIZATION_ID), and Oracle to Siebel Integration (for example, ORGANIZATION_ID).

Refer to "Detailed Data Mappings" for detailed descriptions of these EAM values.

Integration Workflows

Each integration is driven by a workflow. This section provides an annotated workflow diagram for each Siebel application and Oracle Applications integration.

NOTE: There are alternate names for some of the business services described in this section. The Database Adapter (or DB Adapter) is also known as the EAI SQL Adapter, and the Oracle Procedure Adapter is also known as EAI ODBC Service.

Organization Inbound Workflow

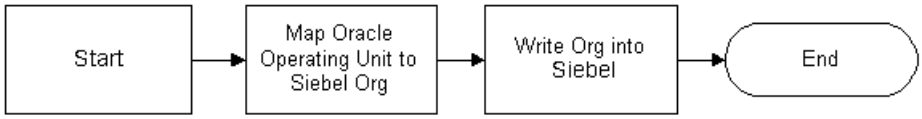
Organization Inbound integration transforms Oracle Operating Units into Siebel Internal Division business objects.

This integration involves the following processes:

1. The SQL script siebelgeneral11iscript_notify_org.sql is run.
2. The script inserts records into the Notification Table for all the Oracle Organizations with the following attributes:
   - OBJECT_NAME: Organization - Get Oracle11i Operating Unit (Oracle Applications)
   - TBL_NAME: SIE_ORG_V
3. The Oracle Receiver:
   a. Polls the Notification Table and gets the entry.
   b. Creates a query specification integration object instance that locates the particular operating unit.
   c. Calls the DB Adapter to extract details of the operating units into an instance of the Oracle integration object.
   d. Executes the Oracle Receiver Dispatch Workflow, passing the instance of the integration object as a parameter.
4. The Oracle Receiver Dispatch Workflow calls:

- Organization – Receive Oracle11i Operating Unit workflow, passing the instance of the Oracle integration object as a parameter.

The workflow diagram for this process is shown in the following figure.



5 The Organization - Receive Oracle Operating Unit workflow:

a Maps the instance of the Oracle integration object to an instance of the Siebel integration object:

- Organization - Get Oracle11i Operating Unit (Siebel Applications), using the Org Maps - Siebel Inbound (OracleR11i) DTE business service.

b Runs the EAI Siebel Adapter with the Execute method to enter the new Oracle organizations into Siebel Applications.

NOTE: The Siebel Connector for Oracle Applications provides scripts to load Oracle operating units into Siebel Applications. These scripts must be run every time a new operating unit is created in Oracle Applications so that it can be created in Siebel Applications.

Inventory Locations Inbound Workflow

Inventory Locations integration transforms Oracle Inventory Organizations data into the Siebel Inventory Location business component. This integration involves the following processes:

1 The SQL script siebelgeneral1iiscript_notify_invloc.sql is run.

2 The script inserts records into the Notification Table for all the Oracle Inventory Organizations with the following attributes:

- OBJECT_NAME: InvLoc- Get Oracle11i Org (Oracle Applications)
- TBL_NAME: SIE_INVLOC_V

3 The Oracle Receiver:

a Polls the Notification Table and gets the entry.

b Creates a query specification integration object instance that locates the particular inventory Organization.

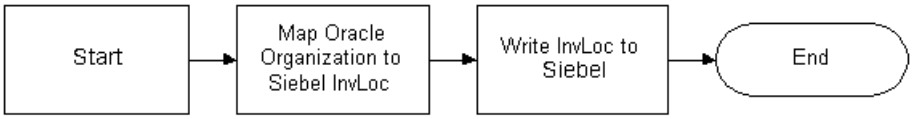
c Calls the Database Adapter to extract details of the Inventory Organization into an instance of the Oracle integration object.

d Executes the Oracle Receiver Dispatch Workflow, passing the instance of the integration object as a parameter.

4 The Oracle Receiver Dispatch Workflow calls:

- InvLoc – Receives Oracle11i Organization workflow and loads all inventory Locations into the Siebel application.

The workflow diagram for this process is shown below.



5 The InvLoc-Receive Oracle Organization workflow:

a Maps the instance of the Oracle Applications integration object to an instance of the Siebel Applications integration object:

- InvLoc - Get Oracle11i Org (Siebel Applications), using the InvLoc Maps - Siebel Inbound DTE business service.

b Runs the EAI Siebel Adapter with the Execute method to enter the new inventory locations into Siebel Applications.

NOTE: The Siebel Connector for Oracle Applications provides scripts to load Oracle Applications Inventory Organizations into Siebel Applications. These scripts must be run every time a new Inventory Organization is created in Oracle Applications so that it can be created in Siebel Applications.

Product Inbound Workflow

Product Inbound integration transforms an Oracle Applications orderable global master material item into a Siebel Internal Product business object.

The integration involves the following processes.

1 An insertion or change to a material item in Oracle Applications fires an Oracle alert or trigger. The material item alert is Siebel Product Update.

NOTE: The Connector also incorporates a trigger SIE_UPDATE_ITEMS_TR. The alert makes an entry into the Notification Table for every update in the Oracle base table. On the other hand, the trigger checks for the CUSTOMER_ORDER_ENABLED_FLAG and makes entries accordingly. You can disable the alert and enable the trigger to reduce the volume of entries in the Notification Table.

2 The alert creates an entry in the Siebel Notification Table with the following attributes:

- OBJECT_NAME = Product - Get Oracle11i Item (Oracle Applications)
- TBL_NAME = SIE_PRODUCT_V

3 The Oracle Receiver:

a Polls the Notification Table and gets the entry.

b Creates a query specification integration object instance that locates the particular material item.

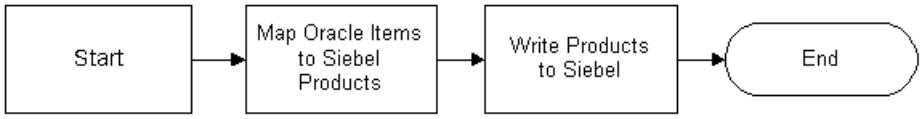
c Calls the DB Adapter to extract details of the material item into an instance of the integration object Product - Get Oracle Item (Oracle Applications). This object is based on the SIE_PRODUCT_V view.

d Executes the Oracle Receiver Dispatch Workflow, passing the instance of the integration object as a parameter.

The Oracle Receiver Dispatch workflow calls:

- Product – Receive Oracle 11i Item workflow and loads the item into the Siebel application.

The diagram for this workflow process is shown in the following figure.



4 The Product – Receive Oracle Item workflow:

a Maps the instance of the Oracle Applications integration object to an instance of the Siebel Applications integration object Product - Get Oracle Item (Siebel Applications), using the Product Maps - Siebel DTE business service.

b Runs EAI Siebel Adapter with the Execute method to enter the new product or product changes into Siebel Applications.

To verify that a product created in Oracle Applications has passed to Siebel Applications

The Oracle Receiver automatically brings new and updated Product data to Siebel Applications at the next scheduled run. To verify the data transfer:

1 Open the Call Center or another Siebel Business application.

2 Select the Products view and query for the product created in Oracle Applications.

3 Click the row containing the product.

4 Select a tab for additional information (for example, Details).

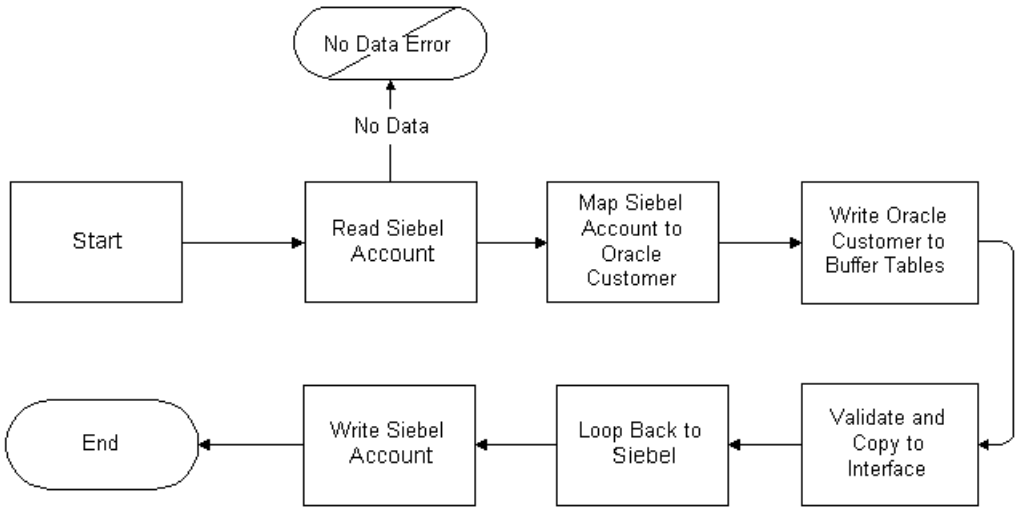
Account Outbound Workflow

Account outbound integration propagates the Siebel Applications integration object Account Put Oracle11i Customer (Siebel Applications) to the Oracle Application R11i integration object Account - Put Oracle 11i Customer (Oracle Applications). The predefined workflow for this integration is described below.

CAUTION: Oracle does not support site use changes on the Siebel or Oracle side after the address is created in the Oracle application.

Account - Create or Update Oracle 11i Customer Workflow

Following is the Account - Create or Update Oracle 11i Customer workflow.



As the workflow diagram indicates, the process involves the following steps:

1. Start.
2. Read Siebel Account—Executes the Business Service EAI Siebel Adapter which reads the data from the Siebel Business Service components and populates the Siebel integration object Account - Put Oracle11i Customer (Siebel Applications).
3. Map Siebel Account to Oracle Customer—Executes the Business Service Account Maps - Siebel Outbound (Oracle Applications 11i) that maps data from the Siebel integration object Account - Put Oracle 11i Customer (Siebel Applications) into the Oracle integration object Account - Put Oracle 11i Customer (Oracle Applications).
4. Write Oracle Customer to Buffer Tables—Executes the Business Service Database Adapter, which reads data from the Oracle integration object
Account - Put Oracle 11i Customer (Oracle Applications) and inserts it into Oracle buffer tables.
5. Validate and Copy to Interface—Executes the Business Service Oracle Procedure Adapter, which runs the prevalidation program SIE_CUSTOMER_PREVAL_PR11i and inserts records from the buffer table into the Oracle interface tables.
6. Loop Back to Siebel Applications—Creates a Siebel message by setting the Integration Id for Account, Contact, and Addresses. It also sets the corresponding Status Codes, Sync Date, and Error Text (if any).
7. Write Siebel Account—Executes the Business Service EAI Siebel Adapter which updates the account in Siebel Applications with data in the Siebel message created in the previous Loop Back to Siebel process.

8 End.

NOTE: Once data is loaded into the Oracle interface tables, the Customer import is run in Oracle Applications. If the import fails, the interface_status column in the interface table is updated with the error message. The Account - Receive Oracle Customer Import Status workflow is used to bring these error messages into Siebel Applications.

Account Inbound Workflow

During Account Inbound integration, the Oracle Application R11i integration object Account - Get Oracle11i Customer (Oracle Applications) is transformed into the Siebel integration object Account - Get Oracle11i Customer (Siebel Applications).

This integration involves the following processes:

1 An insert or change to an Oracle Customer, Customer Address, Customer Contact, Profile, or Phone number in the Oracle application fires an Oracle alert.

The alerts are:

- Siebel Customer Account Insert
- Siebel Customer Contact Insert
- Siebel Customer Interface Header Update
- Siebel Customer Site Insert

2 The alert creates an entry in the Siebel Notification Table with the following attributes:

- For Accounts:
  - OBJECT_NAME = Account - Get Oracle11i Customer (Oracle Applications)
  - TBL_NAME = HZ_CUST_ACCOUNTS
- For Contacts:
  - OBJECT_NAME = Account - Get Oracle11i Customer (Oracle Applications)
  - TBL_NAME = SIE_CONTACTS_V
- For Customer Site:
  - OBJECT_NAME = Account - Get Oracle11i Customer (Oracle Applications)
  - TBL_NAME = SIE_ADDRESSES_V

3 The Oracle Receiver:

a Polls the Notification Table and gets the entry.

b Creates a query specification integration object instance that locates the particular Customer.

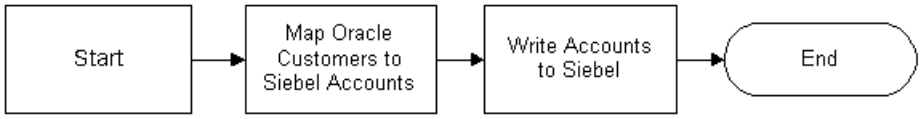
c Calls the Database Adapter to extract details of the customer into an instance of the integration object Account - Get Oracle11i Customer (Oracle Applications).

d Executes the Oracle Receiver Dispatch workflow, passing the instance of the integration object as a parameter.

4 The Oracle Receiver Dispatch workflow calls the Account - Receive Oracle11i Customer workflow and loads the Account into the Siebel application.

Account - Receive Oracle11i Customer Workflow

The predefined workflow for this integration is described as follows.



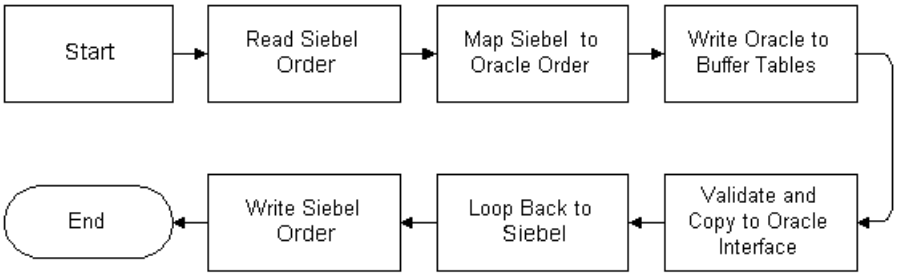
The Account - Receive Oracle11i Customer workflow:

1 Maps Oracle Customer to the Siebel Accounts—The workflow calls Business Service Account Maps - Siebel Inbound to convert data from the Oracle integration object to the Siebel integration object for Account.

2 Write Account to Siebel Applications—The EAI Siebel Adapter is used to update the Siebel database with an appropriate Siebel integration object instance.

Order Outbound Workflow

This predefined workflow propagates Order data from Siebel Applications to Oracle Applications.



This workflow process is summarized as follows.

1 Start—The workflow starts when an order outbound is submitted.

2 Reads Siebel Order—Executes the Business Service EAI Siebel Adapter, which reads data from the Siebel business components and populates the Siebel integration object Order - Put Oracle11i Order (Siebel Applications).

3 Maps Siebel Order to Oracle Order—Executes the Business Service Order Maps - Siebel Outbound (Oracle 11i), which maps data from the Siebel integration object Order - Put Oracle11i Order (Siebel Applications) into the Oracle integration object Order - Put Oracle11i Order (Oracle Applications).

4 Writes Order to Buffer Tables—Executes the Business Service EAI DB Adapter, which reads data from the Oracle integration object Order - Put Oracle11i Order (Oracle Applications) and inserts it into the Oracle buffer tables.

5 Validate and Copy to Oracle Interface—Executes the Business Service EAI ODBC Service, which runs the prevalidation program SieOrderValidate11i. The prevalidation program validates data in the buffer table and calls the Process Order API to create the order in Oracle Applications.

6 Loops Back to Siebel Applications—Creates a Siebel message by setting the Integration Id for Order and Order Line and setting the corresponding Status Codes, Sync Date, and Error Text (if any).

7 Writes to the Siebel Order—Executes the Business Service EAI Siebel Adapter which updates the Order in Siebel with data in the Siebel message created in the previous Loop Back to Siebel process.

8 End—The workflow is complete.

Order Inbound Workflow

Order Inbound integration associates Siebel integration object Order - Put Oracle11i Order (Siebel Applications) with the Oracle integration object Orders - Get Oracle11i Orders (Oracle Applications). This integration involves the following processes:

1 An insert or change to an Oracle Order or Order Line in the Oracle application fires an Oracle alert.

The Order alerts are:

- Siebel Order Header Insert
- Siebel Order Header Update
- Siebel Order Hold Insert
- Siebel Order Hold Update
- Siebel Order Lines Insert
- Siebel Order Lines Update
- Siebel Order

The Connector allows you to delete Order Lines and Orders if the status is ENTERED. The triggers for delete are:

- SIE_ORDER_HEADER_DEL_TR
- SIE_ORDER_LINE_DEL_TR

2 The alert creates an entry in the Siebel Notification Table with the following attributes:

- For Orders:
  - OBJECT_NAME = Order - Get Oracle11i Order (Oracle Applications)
  - TBL_NAME = SIE_HEADERS_ALL_V
- For Order Lines:
  - OBJECT_NAME = Order - Get Oracle11i Order (Oracle Applications)
  - TBL_NAME = SIE_LINES_ALL_V

3 The Oracle Receiver:

   a Polls the Notification Table and gets the entry.

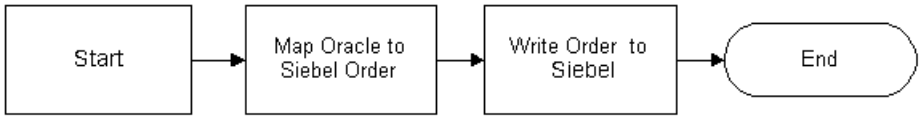
   b Creates a query specification integration object instance that locates the particular Order.

   c Calls the Database Adapter to extract details of the Order into an instance of the integration object Orders - Get Oracle11i Orders (Oracle Applications).

   d Executes the Oracle Receiver Dispatch Workflow, passing the instance of the integration object as a parameter.

4 The Oracle Receiver Dispatch workflow calls the Order - Receive Oracle11i Order workflow and loads the Order into the Siebel application.

Order - Receive Oracle11i Order Workflow



As the workflow indicates, this integration involves the following steps:

1 Start—The Oracle Receiver starts this workflow. For more information, see "Understanding the Oracle Receiver" on page 74.

2 Maps the Oracle to Siebel Order—Calls the Siebel Business Service Order Maps - Siebel Inbound to convert data from the Oracle integration object to the Siebel integration object for Orders.

3 Writes Order to Siebel Applications—The EAI Siebel Adapter is used to update the Siebel database with an appropriate Siebel integration object instance.

4 End—The workflow is complete.

Unicode to Codepage Deployments

The following section provides special procedures and guidelines for Siebel Connector for Oracle Applications deployments in which one application is operating in a Unicode environment while another application is operating in a codepage environment. One of the main issues to consider is that complete character conversion may not be possible if you are using applications with different character set encodings (such as Unicode UTF-16 and a codepage).

CAUTION: Failure to properly address potential character conversion shortcomings may result in data corruption.

Consider the following example:

Software Application	Character Set Encoding Name
Siebel Business Applications 8.0	Unicode UTF-16
Oracle Application	Western European (1252) codepage

In this situation, you may not be able to convert non-Western European language data from Siebel Business Applications 8.0 (UTF-16) to the Oracle application (1252). This limitation occurs because:

- Non-Western European languages such as Japanese, Korean, and Chinese (Simplified and Traditional) require multibyte data representation and the Western European (1252) codepage can only represent data in single-bytes.
- Non-Western European language data in Unicode UTF-16 may not have equivalent characters in the Western European (1252) codepage.

NOTE: The reverse case of this example is also true. For example, you have similar conversion issues when Siebel Business Applications 8.0 is operating in a codepage environment and the Oracle Application operates in a Unicode environment.

Deployment Scenarios

The following scenarios describe integration possibilities between Siebel Business Applications 8.0 and Oracle Applications. Some guidelines that apply to both scenarios are:

- While the Siebel Object Manager *always* operates in a Unicode environment, the Siebel database can operate in either a Unicode (for example, UTF-16) or non-Unicode (for example, codepage) environment.
- Character conversion occurs in two places:
  - Where data moves from the Siebel Database to the Siebel Connector for Oracle Applications on the Siebel Object Manager.
  - Where data moves from the Siebel Connector for Oracle Applications on the Siebel Object Manager to the Oracle Application.

Integrating with Oracle Applications

Table 11 describes integrating Siebel Business Applications 8.0 with Oracle Applications.

Table 11. Integrating with Oracle Applications

Siebel Database	Siebel Object Manager	Oracle Application
Codepage	Unicode	Codepage
Unicode	Unicode	Codepage
Codepage	Unicode	Unicode
Unicode	Unicode	Unicode

Character Conversion and Error Detection

Character conversion occurs when data is exchanged between Siebel Business Applications 8.0 and Oracle Applications that are using different character set encodings.

Outbound Data

For outbound data moving from Siebel Business Applications to Oracle Applications, the data is first converted from the Siebel database (if the Siebel database is operating in a codepage environment) to the Siebel object manager. The first conversion is handled automatically by the Siebel data manager layer. This conversion should always be successful since Unicode character sets can represent any characters used by non-Unicode character sets, such as codepages.

The second conversion is performed by the Open Database Connectivity (ODBC) mechanism interfaced through the Siebel Database Adapter. This conversion can fail if data moves from a Unicode character set to a non-Unicode character set, such as a codepage. The current ODBC driver for the Oracle Relational Database Management System (RDBMS) does not report character conversion errors and may commit errant data to the Oracle Application's interface tables and Application Programming Interfaces (APIs), resulting in data corruption.

To prevent this type of data corruption, implement character conversion error checking in your integration workflow to trap errors before sending data to the Oracle Applications.

Inbound Data

For inbound data moving from Oracle Applications to Siebel Business Applications, the ODBC driver first converts data from the Oracle Application (if the Oracle Application is operating in a codepage environment) to the Siebel object manager (which is always running in a Unicode environment). This conversion should always be successful since Unicode character sets can represent any characters used by non-Unicode character sets, such as codepages.

A second conversion may take place when data moves from the Siebel object manager to the Siebel database if the Siebel database is operating in a codepage environment. This second conversion is performed by the Siebel data manager layer and can fail if data moves from a Unicode character set to a non-Unicode character set, such as a codepage. Unlike the ODBC driver for the Oracle RDBMS, the Siebel data manager avoids data corruption by trapping data conversion errors, reporting them, and not committing the errant data to the Siebel database. However, you may still want to include exception-handling steps to handle the reported error accordingly.

Detecting Errors With the Transcode Business Service

You can use the Transcode business service by adding it as a step in your integration workflow. This business service uses a property set as input, validates and converts it from one character set encoding to a different character set encoding, and outputs the result in another property set. Figure 4 shows how this mechanism can be incorporated into an existing workflow. For more information on the Transcode business service, see *Siebel Global Deployment Guide*.

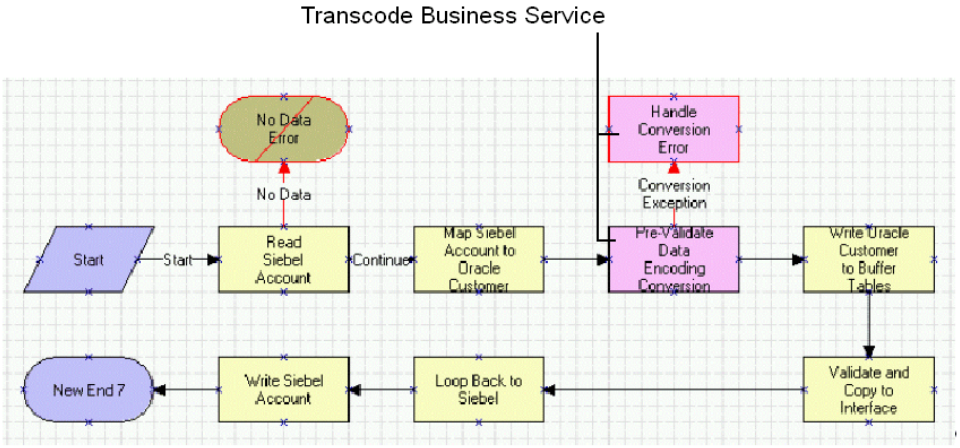


Figure 4. Transcode Business Service for Connector for Oracle Applications

Understanding the Database Adapter

The Database (DB) Adapter is a Siebel business service that can be used to construct and execute SQL statements on an external database. As a SQL execution module, the Adapter is driven by a SQL integration object.

Like other Siebel business services, the DB Adapter simplifies moving and converting data between Siebel Business Applications and external applications.

Similar to the Siebel Integration Manager, the DB Adapter is used to move data to or from an external database. The DB Adapter dynamically creates a SQL statement (or set of SQL statements), based upon data in an integration object and the invoked method, which indicates whether the SQL statement is an INSERT, UPDATE, DELETE, or SELECT. The DB Adapter executes the SQL statements against the external Oracle data store. For INSERT, UPDATE, DELETE, a count or error is returned. In the case of SELECT, the retrieved data values are returned and inserted into a second integration object.

NOTE: The Database Adapter is also known as the EAI SQL Adapter.

Methods and Arguments

The DB Adapter business service, also known as the EAI SQL Adapter, incorporates methods that are used to perform various tasks (queries, data inserts, updates, and so on). Each method can have one or more arguments that are used to further define the method's action. An argument typically consists of data or an object that the method processes.

Five methods may be used with the DB Adapter:

- Query
- Delete
- Upsert
- Synchronize
- Execute

Query

The DB Adapter uses the Query method to query data from an external database based on a SQL integration object and returns the corresponding integration object instances.

Query takes a QBE (Query By Example) instance as input and returns one or more output objects. If the component field values are set in the input, the values will be used in the where clause of the generated SQL. A blank search specification at the root level will query for all the rows from the table corresponding to the root component.

Parent-child relationships are determined by foreign key definitions in the integration object. The Adapter expects the foreign key of a child component to refer to a target key in the parent component.

Arguments

Query arguments are listed in Table 12.

Table 12. Query Arguments

Parameter Name	Required?	Input/ Output	Data Type	Description
NumOutputObjects	No	Output	Number	Number of Output Integration Objects
OutputIntObjectFormat	No	Input	String	Output integration object format. You can set this parameter to one of these formats: ■ Siebel Hierarchical ■ XML Hierarchical ■ XML Flat
SiebelMessage	Yes	Input/ Output	Hierarchy	Input message containing the QBE object instances.

NOTE: This table format will be used for all method arguments described in this book.

Delete

The Delete method is used to delete a hierarchy on the external database that is based upon an integration object. Delete takes a QBE instance as input and deletes the entire hierarchy rooted at the specified root component instance. The search specification is allowed only at the root level. If no search specification is provided, Delete removes all rows from the table corresponding to the root component. The *CascadeDelete* property is specified in the integration object definition at each component level. If this property is set, that component is also deleted when its parent is deleted. Here too, parent-child relationships are determined by the foreign key definition in each component.

Arguments

Delete takes only one argument, defined in Table 13.

Table 13. Delete Argument

Parameter	Required?	Input/Output	Data Type	Description
SiebelMessage	Yes	Input	Hierarchy	QBE instances are contained within the Siebel message.

Upsert

The Upsert method is used to insert and/or update data into the external database based on the input integration object instances. Upsert performs an UPDATE or an INSERT at each component level, depending upon whether the row already exists or not in the database. The input to the upsert method is the actual integration object instance data.

The DB Adapter uses a combination of two algorithms to upsert data, depending upon the ratio of the number of database rows to the number of component instances in the input instance. This optimization is turned on by default. If the number of database rows is small, it is efficient to query for all rows (of a given parent) and try to match them in memory. If there are a large number of database rows, it is more efficient to query the database for each input component instance, to determine whether the corresponding rowset exists in the database.

Upsert supports multiple user key specification to find the matching row in the database. Each user key is tried in sequence to determine whether or not the rowset exists in the database. If none of the specified user key fields have their values set, an error is returned. A null value for any of the user key fields is valid.

Arguments

Upsert takes a single argument, defined in Table 14.

Table 14. Upsert Argument

Parameter	Required?	Input/Output	Data Type	Description
SiebelMessage	Yes	Input	Hierarchy	Input object instances are contained within the Siebel message.

Synchronize

Synchronize makes the values of an external database match the values of an integration object instance by performing an Update, Insert, or Delete on the external tables. The Synchronize method is similar to Upsert except that deletes are performed on database rows where corresponding component instances are not present in the input integration objects.

Arguments

Synchronize takes a single argument, defined in Table 15.

Table 15. Synchronize Argument

Parameter	Required?	Input/Output	Data Type	Description
SiebelMessage	Yes	Input/Output	Hierarchy	The data hierarchy is to be synchronized with the external database.

Execute

The DB Adapter Execute method is used to perform the operations listed below on an integration object. Any operation can be specified at the component level by using the Op Code.

- Delete
- Upsert
- Synchronize

When Execute performs a Delete or Synchronize on a component, all operations below that component are invalid and are ignored.

Arguments

Execute takes only one argument, defined in Table 16.

Table 16. Execute Argument

Parameter	Required?	Input/Output	Data Type	Description
SiebelMessage	Yes	Input/Output	Hierarchy	Inputs Siebel messages.

Op Codes

Operation codes indicate the type of operation to be performed on an integration component. These codes are specified in the component instance; otherwise, they are inherited from the parent's component instance. Processing an integration component, the DB Adapter detects the operation code and performs the action indicated by the code. For example, when the upsert code is detected, the DB Adapter performs an UPSERT operation, starting at that component level in the hierarchy.

- upsert
- delete
- sync
- none

Operation codes are used with the Execute method to specify the operation at the component level.

Additional Information

This section provides additional information about the DB Adapter, also known as the EAI SQL Adapter.

Generating SQL

The DB Adapter generates SQL statements based upon the integration object definitions in tools and data in the input object. The process generates multiple SQL statements, executes them, and joins the result set.

Starting at the root of the tree, the Adapter generates SQL for all children of a component type. For better performance, the Adapter may use a SQL OR clause to group these children together. (Because the maximum length of a SQL statement is limited by ODBC, the length of the SQL statements can be controlled by changing a parameter.)

The DB Adapter business service uses the ODBC API for all database access. With the algorithm outlined above, DB Adapter needs to be processing only one component at a time. All the joins will occur in memory.

The DB Adapter relies upon the ODBC API to cache the underlying ODBC cursors. The ODBC API will not perform connection pooling because it can handle only one open ODBC connection at a time. DB Adapter will cache ODBC connections, if necessary.

Handling Transactions

The DB Adapter does not perform any BeginTransaction or EndTransaction on the Siebel side of data flow transactions. It does perform BEGIN, COMMIT, and ROLLBACK transactions on the Oracle database through the ODBC API. The ODBC API provides interfaces to support manual transaction control.

Translating Data Types

The ODBC API, used by the DB Adapter, converts the generic data types specified in the tools to the corresponding ODBC data types. The SQL integration objects are expected to encode these ODBC data types as their External Data Type. The SQL wizard or Oracle Applications wizard (that generates these integration objects) is used to set this external data type. The wizard may query Oracle Applications metadata to obtain database data types such as NUMBER, VARCHAR2, and so on. Then the ODBC API translates these application-specific data types into ODBC data types.

Passing Connection Parameters

The DB Adapter passes the following connection parameters to the ODBC API for its Connect method:

- ODBC DSN
- Database username
- Database password
- Table owner

The DB Adapter looks for these parameters in the following locations:

- Method arguments
- Server parameters
- Service user properties

The Adapter assumes that these parameters are passed to it by the caller. It will not explicitly retrieve these parameters.

Error Handling

The DB Adapter handles errors like other EAI external adapters. If an error occurs in the Oracle application database system, the error is reported by the underlying low-level ODBC layer. Regardless of the error type (database connectivity problem, invalid table/view/column names, and so on), the error from the ODBC layer is passed up the stack to the caller of the DB Adapter. The DB Adapter will not interpret the error or perform any recover procedure.

Passing the Oracle Operations Field to the DTE

Because Oracle Applications must be able to specify operations that will be carried out on Siebel business objects, each Siebel integration object must have a meta-operator field that can be filled in to specify the operation. The Field Type for the field is set to System (External). The DB Adapter does not use this field; it just passes the field to the Data Transformation Engine (DTE).

Understanding the Oracle Receiver

The Oracle Receiver, a continuously running program, is used when Oracle Applications have updated information that has to be transferred to Siebel Applications. This section describes the Receiver, focusing upon operations, associated tables, methods, and arguments. It also explains how to run the Receiver as a server component or business service.

Running the Oracle Receiver

The Oracle Receiver runs continuously, regularly polling the Notification Table for primary keys that identify changed rows in Oracle database base tables. When an Oracle table row is updated, an Oracle alert inserts a row with the primary keys into the Notification Tables. The next time the Oracle Receiver polls the Notification Tables, it sends this information to the DB Adapter.

The DB Adapter uses this information to extract the changed row information from the base tables and sends it to the Siebel application. For example, if there is a product information update to the Oracle MTL_SYSTEMS_ITEMS table, the Oracle alert Siebel Product Updates inserts the primary key of the changed row into Notification Table SIE_NOTIFY_TBL.

If the Oracle Receiver is shut down by a system or power failure, it is restarted as part of the general restarting process.

Operational Summary

When an Oracle database base table row is updated, a standard Oracle Events Alert or a database trigger copies the just-updated row to the Notification Table (SIE_NOTIFY_TBL) with the Status column value set to Ready for Transfer. At scheduled intervals or when user requests are made, the Oracle Receiver polls the Notification Table. When the Receiver finds updates, it extracts the rows into integration objects.

NOTE: The Notification Table can include records of several types of objects. All rows of the same type are extracted into a single Siebel message containing many integration object instances of that type.

The Receiver starts a separate workflow for each integration object. The Receiver calls the Oracle Receiver Dispatch Workflow. This workflow passes data through the Data Transformation Engine (DTE) into a Siebel integration object, and then passes it into a Siebel Business Object to update the Siebel database tables.

When the workflow is completed successfully, the Receiver deletes the corresponding rows from the Notification Table. If the workflow fails, the status of the rows is set to Error in the Notification Table.

The overall pattern of Oracle Receiver activity is represented in Figure 5.

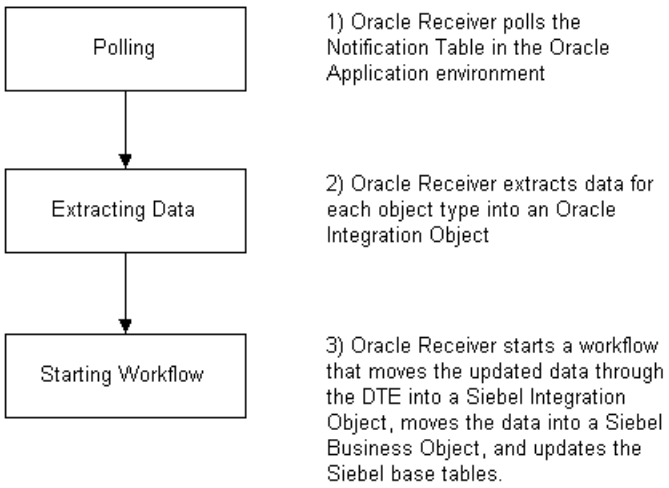


Figure 5. Oracle Receiver Operations

Oracle Receiver Dispatching

Activated by the ReceiverDispatchMethod, the Oracle Receiver Dispatch Workflow controls the Oracle Receiver's data dispatching operations. When it obtains a new integration object (based upon Oracle Base Table changes reflected in the Notification Table), the Receiver Dispatcher workflow branches the flow of execution in the way shown in Figure 6 on page 76.

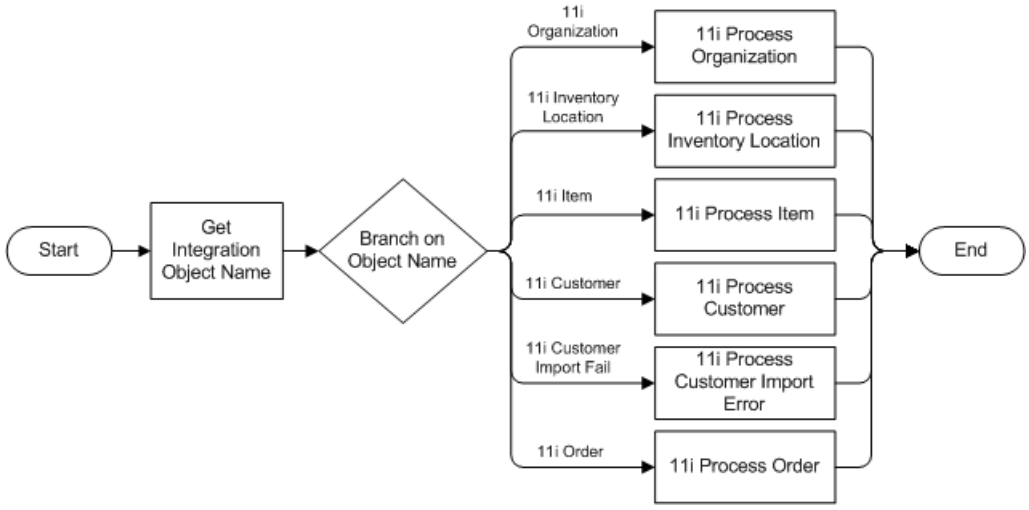


Figure 6. Oracle Receiver Dispatch Workflow

As Figure 6 on page 76 shows, the Oracle Receiver Dispatch workflow accommodates the following integration object types:

- Organization
- Inventory Location
- Item
- Customer
- Customer Input Error
- Order
- Order Input Errors

Notification Table

The Notification Table (SIE_NOTIFY_TBL) contains information about the Oracle base table and row that were updated. It identifies the primary keys and the type of update activity (Insert, Update, or Delete).

Table Structure

Notification Table columns are listed in Table 17.

Table 17. Notification Table Columns

Column	Description	Not Null	Column Type and Length
Seq_Num	Primary key.	Yes	Number
Object_Name	Name of the integration object.	Yes	Varchar2 (75)
Tbl_Name	Name of the Oracle base table containing the updated row.	Yes	Varchar2 (30)
Pri_Key1	Primary key 1. In combination with Pri_Key2, identifies the rowid.	No	Varchar2 (240)
Pri_Key2	Primary key 2. In combination with Pri_Key1, identifies the rowid.		Varchar2 (240)
Root_Tbl_Pri_Key1			Varchar2 (240)
Root_Tbl_Pri_Key2			Varchar2 (240)
Original_System_Reference			Varchar2 (240)
Operation	Type of operation (Insert, Update, Delete, and so on).	Yes	Varchar2 (1)
Status	Status of the row (Ready for Transfer, In Process, and so on).	Yes	Varchar2 (10)
Type	Data or error (usually, data).	Yes	Varchar2 (10)
Error_Message	Error message text (for example, Error).		Varchar2 (2000)
Last_Update_Date	Last date that the row was updated.	Yes	Date
Last_Updated_By	Number identifying the last person who updated the row.	Yes	Number
Creation_Date	Date that the row was created.	Yes	Date
Created_By	Number identifying the person who created the row.	Yes	Number
Last_Update_Login	Number identifying the last login.		Number

Table 17. Notification Table Columns

Column	Description	Not Null	Column Type and Length
Adpt_Mode	System use only.	Yes	Varchar2 (10)
Queue_Name	Needed if the ORCL QUEUE NAME parameter is set to find matching rows.	Yes	Varachr2 (30)

Methods and Arguments

The Oracle Receiver program incorporates methods that are used to perform various tasks. Each method can have one or more arguments that are used to further define the method's action. An argument typically consists of data or an object that the method processes.

The Oracle Receiver can be run as a server component or business service. The following list identifies the parameter, methods, and arguments used with each type of installation.

Running as a Server Component---As a server component, the Oracle Receiver can be run with the fixed parameter and parameters listed as follows.

Fixed Parameters

- ReceiverServiceName
- ReceiverMethodName
- ReceiverDispatchService
- ReceiverDispatchMethod

Parameters

- ExtDBODBCDataSource
- ExtDBPassword
- ExtDBUserName
- ExtDBTableOwner
- ORCLQueueName
- ORCLBatchSize

The Oracle Receiver's fixed parameters are described in Table 18.

Fixed Parameters

Table 18. Oracle Receiver Fixed Parameters

Parameter Name / Displayed Name	Required?	Description
ReceiverServiceName = EAI Oracle Receiver	Yes	Specifies the service (Oracle Receiver).
ReceiverMethodName = RunData	Yes	Specifies the method.
ReceiverDispatchService = EAI Oracle Receiver Dispatch	Yes	Specifies the Dispatch Business Service that the Receiver would invoke.
ReceiverDispatchMethod = Execute	Yes	Specifies the method to call on the service above.

Parameters

The Oracle Receiver's variable parameters are described in Table 19.

Table 19. Oracle Receiver Variable Parameters

Parameter Name / Displayed Name	Required?	Description
ExtDBODBCDataSource = ORCL ODBC DataSource	Yes	ODBC data source name for the Oracle database.
ExtDBPassword = ORCL Password	Yes	Password for the datasource.
ExtDBUserName = ORCL User Name	Yes	Username for the datasource.
ExtDBTableOwner = ORCLE Table Owner	Yes	Table owner for the data source.
ORCL BatchSize = ORCL Batch size		If this parameter is specified, the Oracle Receiver will extract, in a single pass, the maximum number of records from the Notification Table, up to the ExtDBBatchSize.
ORCLQueueName = ORCL Queue Name		System parameter.

Delete Processing

When data is deleted in Oracle Applications, the Oracle Receiver passes the deletion instructions on to Siebel Applications. This process is outlined below.

1. Rowsets are deleted in the Oracle Applications table.
2. The Opcode for the deleted rows is set to D in the Notification Table.
3. The Oracle Receiver passes the integration Id of the deleted row to the DTE.
4. The DTE deletes data in Siebel Applications by using business component interfaces, or calling the Delete method in Siebel Adapter.

Understanding the Oracle Applications Wizard

The Oracle Applications wizard, also known as the Integration Object wizard, is used when data flows are created to accurately define Oracle Applications Integration Objects.

Operational Summary

The Oracle Applications wizard is used to adapt and extend predefined integration points provided with Siebel Connector for Oracle Applications. Integration objects define the structure of messages that are exchanged between Oracle's Siebel Business Applications and the Oracle Application.

The wizard allows you to create and modify these objects. It directly queries the Oracle data model definition to retrieve the data structure that must be represented by new or modified integration objects. This process automates the transfer of customization that typically has to be made in the object definitions of Oracle Applications.

These operations are represented in Figure 7.

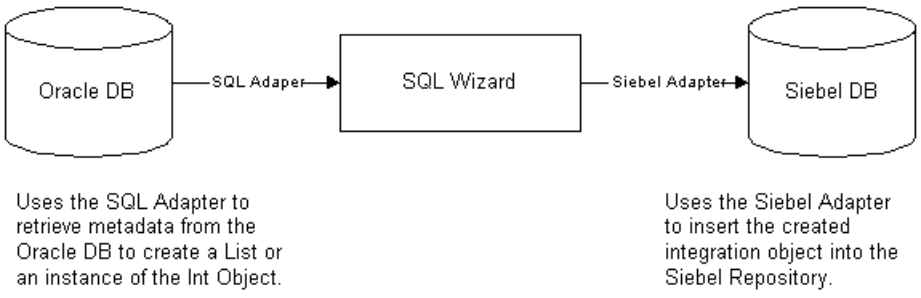


Figure 7. Oracle Applications Wizard Operations

Working with Integration Objects

SQL integration objects are metadata that define the schema of a table hierarchy in the external Oracle database. These objects contain integration components that map directly to Oracle tables and views. They also incorporate the primary key and foreign key relationships that exist between the tables in the external database.

Using Metadata to Create Integration Objects

As it creates integration objects, the Oracle Applications wizard retrieves metadata information from FND tables in the external Oracle database. Figure 8 shows the relationship between the Oracle Applications integration objects and the Oracle Applications database tables.

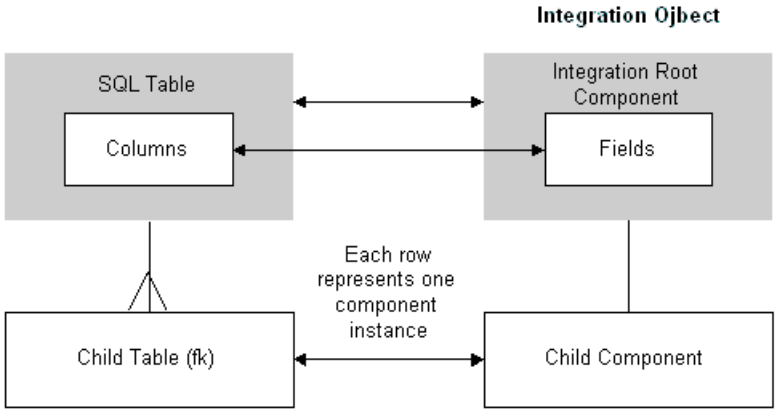


Figure 8. Relationship Between SQL Integration Objects and SQL Database Tables

NOTE: An Oracle Applications integration object may have SQL as the base object, enabling the Adapter to validate the objects.

Obtaining Oracle Application Metadata

The Oracle Wizard uses the DB Adapter to obtain information about Oracle Applications metadata. In this process, the Wizard passes integration objects to the DB Adapter that indicate where and how the Adapter should fetch the information. The Adapter obtains the following Oracle Applications database information for the wizard:

- Names of the tables that will be used to create the integration objects
- Names and properties of the table columns
- Referential constraints of the tables
- Foreign key columns of given foreign key relationships
- Primary key columns of the tables

Oracle Tables Containing Metadata

The Wizard uses metadata contained in the following Oracle Applications tables:

- FND_APPLICATION
- FND_TABLES
- FND_COLUMNS
- FND_FOREIGN_KEYS
- FND_FOREIGN_KEY_COLUMNS
- FND_PRIMARY_KEYS
- FND_PRIMARY_KEY_COLUMNS

See the Oracle Application Technical Reference Manuals for descriptions of these tables.

Metadata Integration Objects Used to Query the SQL Adapter

The Oracle Applications wizard uses the following metadata integration objects to query the DB Adapter for needed information:

- Oracle Wizard Get Tables—Used to query for all tables of an application. Cached locally to facilitate mapping of names to IDs and IDs to names.
- Oracle Wizard Get Children—Used to query for children of a table. The Primary Key Application ID and Primary Key Table ID are set as search specifications.
- Oracle Wizard Get Table All—Used to query for the columns, primary key fields, and foreign key fields of a table. A query is made with respect to a specific foreign key that is set as a search specification on the FND_FOREIGN_KEYS component.

Running the Wizard

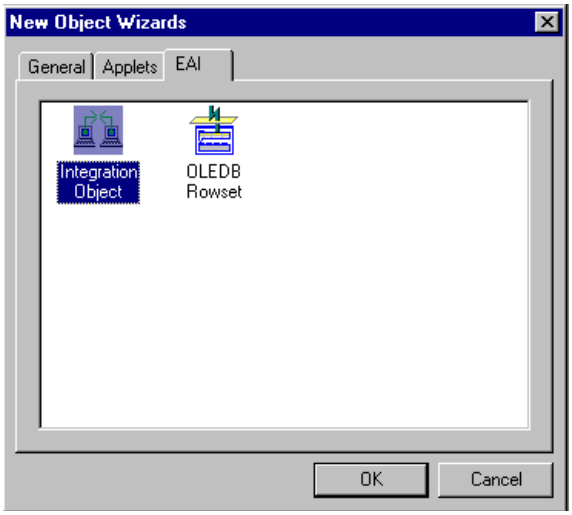
This section explains how to run the Oracle Applications wizard.

To run the Oracle Application wizard

1. Check the Siebel Tools configuration file (tools.cfg) to be certain that it includes the correct Oracle parameters, including ODBC Datasource, Username, Password, and Table Owner.

2 In Siebel Tools, click on the New Object button on the toolbar.

The New Object Wizards screen appears:



3 Click on Integration Object type and click OK.

The Integration Object Builder screen appears.

4 On the drop-down menu, select the Project for the integration object.

5 Select the Wizard name (for example, EAI Oracle Application Wizard) and then click Next.

6 When the refreshed Integration Object Builder screen appears, select the source object for the integration object from the drop-down menu. It lists table names with the application names within parentheses. Enter an appropriate name for the integration object in the field below. Then click Next.

7 When the Integration Object Builder - Choose Integration Components screen appears, select the components to be included in the final integration object. Then click Next.

8 The integration object is saved in the database and a confirmation screen appears. Click Finish to close the wizard.

9 You can view the new integration object hierarchy created in Oracle's Siebel Tools.

Understanding the Oracle Procedure Adapter

The Oracle Procedure Adapter invokes a stored procedure in an Oracle Applications database.

NOTE: The Oracle Procedure Adapter is also known as EAI ODBC Service.

Operational Summary

The Oracle Procedure Adapter (EAI ODBC Service) is used to invoke any stored procedure on the external database. In turn, this is used to move data from interface (buffer) tables to Oracle base tables. It includes error-handling logic to promote reliable data exchange by appropriately managing exceptional conditions.

The Adapter runs a stored procedure in the Oracle Applications database through an ODBC call. Oracle supplies an API, FndAPI, which is a stored procedure. This procedure can start another stored procedure. When the called procedure completes, FndAPI returns control- and completion-error codes to the Oracle Procedure Adapter.

Methods and Arguments

The Oracle Procedure Adapter (EAI ODBC Service) uses a single method, ExecuteProcedure.

ExecuteProcedure

The ExecuteProcedure method runs a stored procedure on the Oracle Database through ODBC. The stored procedure takes two arguments, an input (first argument) and an output (second argument). The arguments are listed in Table 20.

Table 20. ExecuteProcedure Argument

Parameter Name	Description
ProcedureName	Name of the stored procedure being invoked.
ProcedureArgument	Input argument parameter value.
Status	Output parameter returned from the stored procedure.

A Detailed Data Mappings

This chapter gives details about data mappings.

Entity Attribute Mappings for Oracle Applications

This section provides detailed information about Siebel Applications and Oracle Applications integrations.

- Organization
- Inventory Location
- Product
- Order

Organization Integration EAM Mappings

This mapping, shown in Table 21 and Table 22, defines Oracle Applications organization data to the Siebel Internal Division business component.

Entity Mapping

Table 21. Organization Integration EAM Mapping

Siebel Business Component	Oracle Base Table	Oracle to Siebel Integration
Internal Division	HR_ORGANIZATION_ UNITS, HR_ORGANIZATION_ INFORMATION	SIE_ORG_V

Attribute Mapping

Table 22. Organization Integration Attribute Mapping

Siebel Field (Internal Division)	Oracle Base Column (HR_ORGANIZATION_UNITS, HR_ORGANIZATION_ INFORMATION)	Oracle to Siebel Integration (SIE_ORG_V)
Integration Id	HR_ORGANIZATION_UNITS. ORGANIZATION_ID	OPERATING_UNIT_ID
Name	HR_ORGANIZATION_UNITS. NAME	NAME
Currency Code	HR_ORGANIZATION_ INFORMATION.ORG_ INFORMATION10	CURRENCY_CODE
Organization Flag		Set to Y in Order Maps - Siebel Outbound (Oracle R11i) DTE business service)
Parent Organization Id		Set to Y in Order Maps - Siebel Outbound (Oracle R11i) DTE business service.

Inventory Location Integration EAM Mapping

This mapping, shown in Table 23 through Table 25, defines the propagation of Oracle Organizations data to the Siebel FS Inventory Location business component.

Entity Mapping

Table 23. Inventory Location Integration EAM Mapping

Siebel Business Component	Oracle Base Table	Oracle to Siebel Integration
FS Inventory Location	hr_organization_units	SIE_INVLOC_V
FS Inventory Locations	hr_organization_ information	SIE_INVLOC_V

Attribute Mapping

Table 24. Inventory Location Integration Attribute Mapping (part 1)

Siebel Field (FS Inventory Location)	Oracle Base Column (HR_ORGANIZATION_ UNITS)	Oracle to Siebel Integration (SIE_INVLOC_V)
Integration Id	ORGANIZATION_ID	ORGANIZATION_ID
Inventory Name	NAME	NAME
Inventory Type	TYPE	TYPE

Table 25. Inventory Location Integration Attribute Mapping (part 2)

Siebel Field (FS Inventory Location)	Oracle Base Column (HR_ORGANIZATION_ INFORMATION)	Oracle to Siebel Integration (SIE_INVLOC_V)
Organization	ORG_INFORMATION3	OPERATING_UNIT

Product Integration EAM Mappings

This integration associates orderable Oracle global master material items with the Siebel Internal Product business object. The EAM mapping for this integration is shown Table 26 to Table 29.

Entity Mapping

Table 26. Product Integration EAM Mapping

Siebel Business Component	Oracle Base Table	Oracle to Siebel Integration
Internal Product	MTL_SYSTEM_ITEMS	SIE_PRODUCT_V
Internal Division	ORG_ORGANIZATION_ DEFINITIONS	SIE_PRODUCT_V
FS Inventory Location	ORG_ORGANIZATION_ DEFINITIONS	SIE_PRODUCT_V

Attribute Mapping

Table 27. Product Integration Attribute Mapping (part 1)

Siebel Field (Internal Product)	Oracle Base Column (MTL_System_Items)	Oracle to Siebel Integration (SIE_Product_V)
Integration Id	INVENTORY_ITEM_ ID	INVENTORY_ITEM_ID
Description	DESCRIPTION	DESCRIPTION
Orderable	CUSTOMER_ORDER_ ENABLED_FLAG	CUSTOMER_ORDER_ ENABLED_FLAG
Sales Product Flag	CUSTOMER_ORDER_ FLAG	CUSTOMER_ORDER_ FLAG
Sales Service Flag	SERVICE_ITEM_ FLAG	SERVICE_ITEM_FLAG
Unit of Measure	PRIMARY_UOM_ CODE	PRIMARY_UOM_CODE (through EAI value map Oracle Unit of Measure)

Table 28. Product Integration Attribute Mapping (part 2)

Siebel Field (Internal Division)	Oracle Base Column (ORG_ORGANIZATION _DEFINITIONS)	Oracle to Siebel Integration (SIE_Product_V)
Integration Id	OPERATING_UNIT	OPERATING_UNIT
Default Shipping Org Int Id	MTL_SYSTEM_ITEMS .DEFAULT_SHIPPING _ORG	DEFAULT_SHIPPING_ ORG
Status	MTL_SYSTEM_ITEMS .INVENTORY_ITEM_ STATUS_CODE	INVENTORY_ITEM_ STATUS_CODE

Table 29. Product Integration Attribute Mapping (part 3)

Siebel Field (FS Inventory Location)	Oracle Base Column (ORG_ORGANIZATION _DEFINITIONS)	Oracle to Siebel Integration (SIE_Product_V)
Integration Id	MTL_SYSTEM_ITEMS .ORGANIZATION_ID	ORGANIZATION_ID

Order Integration EAM Mapping

The Siebel Order business object is integrated with Oracle Order. The EAM mapping associated with this integration is shown in Table 30 and Table 31.

Entity Mapping

Table 30. Order Entity Mapping

Siebel Business Component	Oracle Base Table	Siebel to Oracle Integration	Oracle to Siebel Integration
Order Entry - Orders	OE_ORDER_HEADERS_ALL (OE_TRANSACTION_TYPES_ALL,HZ_CUST_SITE_USE_ALL)	SIE_ORDER_HEADERS_ALL	SIE_HEADERS_ALL_VSIE_HEADERS_HOLDS_V
Order Entry - Line Items	OE_ORDER_LINES_ALL (HZ_CUST_SITE_USE_ALL)	SIE_ORDER_LINES_ALL	SIE_LINES_ALL_VSIE_ORDER_LINE_HOLDS_V

Attribute Mapping

Table 31. Order Attribute Mapping

Siebel Field	Oracle Base Column	Siebel to Oracle Integration	Oracle to Siebel Integration
Siebel Field (Order Entry - Orders)	**Oracle Base Column (OE_ORDER_HEADERS_ALL)**	**Siebel to Oracle Integration (SIE_ORDER_HEADERS_ALL)**	**Oracle to Siebel Integration (SIE_HEADERS_ALL_V)**
Created	CREATION_DATE	CREATION_DATE	
Created By	CREATED_BY	CREATED_BY	
Updated	LAST_UPDATE_DATE	LAST_UPDATE_DATE	
Updated By	LAST_UPDATED_BY	LAST_UPDATED_BY	
Id	ORIG_SYS_DOCUMENT_REF	ORIG_SYS_DOCUMENT_REF	ORIG_SYS_DOCUMENT_REF
Order Type	TRANSACTION_TYPE_CODE(OE_TRANSACTION_TYPES_ALL)	ORDER TYPE	
Order Date	ORDERED_DATE	ORDERED_DATE	DATE_ORDERED

Table 31. Order Attribute Mapping

Siebel Field	Oracle Base Column	Siebel to Oracle Integration	Oracle to Siebel Integration
Siebel Field (Order Entry - Orders) (continued)	**Oracle Base Column (OE_ORDER_HEADERS_ ALL) (continued)**	**Siebel to Oracle Integration (SIE_ORDER_HEADERS _ALL) (continued)**	**Oracle to Siebel Integration (SIE_HEADERS_ALL_V) (continued)**
	BOOKED_DATE	BOOKED_DATE	
Currency Code	TRANSACTIONAL_ CURR_CODE	TRANSACTIONAL_ URR_CODE	CURRENCY_CODE
Pricing Date	PRICING_DATE	PRICING_DATE	
Status	FLOW_STATUS_CODE	ENTERED_STATE_NAME (through EAI Value Map Oracle R11i Order Status)	STATUS (through EAI Value Map Oracle R11i Order Status)
No Siebel field available	BOOKED_FLAG	BOOKED_FLAG (set to Y when order status is BOOKED)	
Account Integration Id	SOLD_TO_ORG_ID	SOLD_TO_ORG_ID	CUSTOMER_ID
Primary Position Contact Id	SALESREP_ID	SALESREP_ID	
Requested Ship Date	REQUEST_DATE	REQUEST_DATE	DATE_REQUESTED_ CURRENT
Tax Exempt	TAX_EXEMPT_FLAG	TAX_EXEMPT_FLAG (through EAI Value Map Oracle R11i Tax Exempt Flag)	TAX_EXEMPT_FLAG (through EAI Value Map Oracle R11i Tax Exempt Flag)
Tax Exempt Number	TAX_EXEMPT_NUMBER	TAX_EXEMPT_NUMBER	TAX_EXEMPT_NUMBER
Tax Exempt Reason	TAX_EXEMPT_REASON_ CODE	TAX_EXEMPT_REASON_ CODE (through EAI Value Map Oracle R11i Tax Exempt Reason)	TAX_EXEMPT_REASON_ CODE (through EAI Value Map Oracle R11i Tax Exempt Reason)
Siebel Field (Order Entry - Orders) (continued)	**Oracle Base Column (OE_ORDER_HEADERS_ ALL) (continued)**	**Siebel to Oracle Integration (SIE_ORDER_HEADERS _ALL) (continued)**	**Oracle to Siebel Integration (SIE_HEADERS_ALL_V) (continued)**

Table 31. Order Attribute Mapping

Siebel Field	Oracle Base Column	Siebel to Oracle Integration	Oracle to Siebel Integration
Price List Integration Id (Order Entry - Orders)	PRICE_LIST_ID	PRICE_LIST_ID	PRICE_LIST_ID
Order Priority	SHIPMENT_PRIORITY_ CODE	SHIPMENT_PRIORITY_ CODE (through EAI Value Map Oracle R11i Shipment Priority)	SHIPMENT_PRIORITY_ CODE
Carrier Type	SHIPPING_PRIORITY _CODE	SHIPPING_METHOD_ CODE (through EAI Value Map Oracle R11i Ship Method Code)	SHIPPING_PRIORITY CODE (through EAI Value Map Oracle R11i Ship Method Code)
Freight Terms	FREIGHT_TERMS_CODE	FREIGHT_TERMS_CODE (through EAI Value Map Oracle R11i Freight)	FREIGHT_TERMS_CODE (through EAI Value Map Oracle R11i Freight)
Freight Terms Info	FOB_POINT_CODE	FOB_POINT_CODE (through EAI Value Map Oracle 1i Fob Code)	FOB_CODE (through EAI Value Map Oracle11i Fob Code)
Ship Instruction	SHIPPING_ INSTRUCTIONS	SHIPPING_ INSTRUCTIONS	SHIPPING_INSTRUCTIONS
Organization Integration Id	ORG_ID	ORG_ID	ORG_ID
	-	SOLD_FROM_ORG_ID	
No Siebel Field Available		VERSION_NUMBER (hard coded to 1 in DTE)	
Bill To Address Site Use Integration Id	CUST_ACCT_SITE_ID(HZ _CUST_SITE_USE_ALL)	INVOICE_TO_ORG_ID	INVOICE_TO_ADDRESS_I D (subject to change)
Siebel Field (Order Entry - Orders) (continued)	**Oracle Base Column (OE_ORDER_HEADERS _ ALL) (continued)**	**Siebel to Oracle Integration (SIE_ORDER_HEADERS _ALL) (continued)**	**Oracle to Siebel Integration (SIE_HEADERS_ALL_V) (continued)**

Table 31. Order Attribute Mapping

Siebel Field	Oracle Base Column	Siebel to Oracle Integration	Oracle to Siebel Integration
Ship To Address Site Use Integration Id	CUST_ACCT_SITE_ID(HZ_CUST_SITE_USE_ALL)	SHIP_TO_ORG_ID	SHIP_TO_ADDRESS_ID
Ship To Contact Integration Id	SHIP_TO_CONTACT_ID		SHIP_TO_CONTACT_ID
Payment Term Integration Id	PAYMENT_TERM_ID	PAYMENT_TERM_ID	TERM_ID
Payment Method	PAYMENT_TYPE_CODE	PAYMENT_TYPE_CODE (through EAI Value Map Oracle R11i Payment Type Code)	PAYMENT_TYPE_CODE (through EAI Value Map Oracle R11i Payment Type Code)
Credit Card Expiration Date	CREDIT_CARD_ EXPIRATION_DATE	CREDIT_CARD_ EXPIRATION_DATE	
Credit Card Holder	CREDIT_CARD_HOLDER_ NAME	CREDIT_CARD_HOLDER_ NAME	
Credit Card Number	CREDIT_CARD_NUMBER	CREDIT_CARD_NUMBER	
Integration Id	HEADER_ID		HEADER_ID
Back Office Order Number	ORDER_NUMBER		ORDER_NUMBER
Contact Integration Id	SOLD_TO_CONTACT_ID		ORDERED_BY_CONTACT_ ID
Hold Flag (hard coded in inbound DTE)			

Table 31. Order Attribute Mapping

Siebel Field	Oracle Base Column	Siebel to Oracle Integration	Oracle to Siebel Integration
Siebel Field (Order Entry - Line Items)	**Oracle Base Table (OE_ORDER_LINES_ ALL)**	**Siebel to Oracle Integration (SIE_ORDER_LINES_IN TERFACE)**	**Oracle to Siebel Integration (SIE_LINES_ALL_V)**
Created	CREATION_DATE	CREATION_DATE	
Created By	CREATED_BY	CREATED_BY	
Updated	LAST_UPDATE_DATE	LAST_UPDATED_DATE	
Updated By	LAST_UPDATED_BY	LAST_UPDATED_BY (through EAI Value Map Oracle R11i User ID)	
Id (Order Entry - Orders)	ORIG_SYS_DOCUMENT_ REF	ORIG_SYS_DOCUMENT_ REF	
Id	ORIG_SYS_LINE_REF	ORIG_SYS_LINE_REF	ORIG_SYS_LINE_REF
Line Number	LINE_NUMBER	LINE_NUMBER	LINE_NUMBER
Quantity Requested		ORDERED_QUANTITY	ORDERED_QUANTITY
No Siebel field available	ORDERED_QUANTITY	CALCULATE_PRICE_ FLAG (hard coded to Y in DTE)	
Due Date	REQUEST_DATE	REQUEST_DATE	DATE_REQUESTED
Base Price	UNIT_LIST_PRICE	UNIT_LIST_PRICE	UNIT_LIST_PRICE
Unit Price	UNIT_SELLING_PRICE	UNIT_SELLING_PRICE	UNIT_SELLING_PRICE
Product Integration Id	INVENTORY_ITEM_ID	INVENTORY_ITEM_ID	INVENTORY_ITEM_ID
Quantity Shipped	SHIPPED_QUANTITY	SHIPPED_QUANTITY	SHIPPED_QUANTITY
Order Priority (Order Entry - Orders)	SHIPMENT_PRIORITY_ CODE	SHIPMENT_PRIORITY_ CODE (through EAI Value Map Oracle R11i Shipment Priority)	
Siebel Field (Order Entry - Line Items) (continued)	**Oracle Base Table (OE_ORDER_LINES_ ALL) (continued)**	**Siebel to Oracle Integration (SIE_ORDER_LINES_IN TERFACE) (continued)**	**Oracle to Siebel Integration (SIE_LINES_ALL_V) (continued)**

Table 31. Order Attribute Mapping

Siebel Field	Oracle Base Column	Siebel to Oracle Integration	Oracle to Siebel Integration
Carrier Code	SHIPPING_METHOD_CODE	SHIPPING_METHOD_CODE (through EAI Value Map Oracle R11i Ship Method Code)	SHIPPING_METHOD_CODE (through EAI Value Map Oracle R11i Ship Method Code)
Price List Integration Id	PRICE_LIST_ID	PRICE_LIST_ID	
Source Inventory Loc Integration Id (Order Entry - Orders)	SOLD_FROM_ORG_ID	SOLD_FROM_ORG_ID	
	SHIP_FROM_ORG_ID	SHIP_FROM_ORG_ID	
Tax Exempt Flag	TAX_EXEMPT_FLAG	TAX_EXEMPT_FLAG (through EAI Value Map Oracle R11i Tax Exempt Flag)	
Tax Exempt Number	TAX_EXEMPT_NUMBER	TAX_EXEMPT_NUMBER	TAX_EXEMPT_NUMBER
Tax Exempt Reason	TAX_EXEMPT_REASON_CODE	TAX_EXEMPT_REASON_CODE (though EAI Value Map Oracle R11i Tax Exempt Reason)	TAX_EXEMPT_REASON_CODE (though EAI Value Map Oracle R11i Tax Exempt Reason)
Organization Integration Id (Order Entry - Orders)	ORG_ID	ORG_ID	
Bill To Address Site Use Integration Id (Order Entry - Orders)	INVOICE_TO_ORG_ID	INVOICE_TO_ORG_ID	

Table 31. Order Attribute Mapping

Siebel Field	Oracle Base Column	Siebel to Oracle Integration	Oracle to Siebel Integration
Siebel Field (Order Entry - Line Items) (continued)	**Oracle Base Table (OE_ORDER_LINES_ ALL) (continued)**	**Siebel to Oracle Integration (SIE_ORDER_LINES_IN TERFACE) (continued)**	**Oracle to Siebel Integration (SIE_LINES_ALL_V) (continued)**
Ship to Address Integration Id	CUST_ACCT_SITE_ID(HZ _CUST_SITE_USE_ALL)		SHIP_TO_ADDRESS_ID
Payment Term Integration Id (Order Entry - Orders)	PAYMENT_TERM_ID	PAYMENT_TERM_ID	
Integration Id	LINE_ID		LINE_ID
Cancelled Quantity	CANCELLED_QUANTITY		CANCELLED_QUANTITY
Price List Id (Siebel row_id of the associated price list is found in DTE function GetPriceListI d, and then set Price List Id to this found row_id)			

Table 31. Order Attribute Mapping

Siebel Field	Oracle Base Column	Siebel to Oracle Integration	Oracle to Siebel Integration
Source Inventory Loc Id (Siebel row_id of the associated inventory location is found in DTE function GetInvLocId, and then set it to Source Inventory Loc Id)			
Siebel Field (Order Entry - Line Items) (continued)	**Oracle Base Table (OE_ORDER_LINES_ ALL) (continued)**	**Siebel to Oracle Integration (SIE_ORDER_LINES_IN TERFACE) (continued)**	**Oracle to Siebel Integration (SIE_LINES_ALL_V) (continued)**
Invoiced Quantity	INVOICED_QUANTITY		INVOICED_QUANTITY
Status	FLOW_STATUS_CODE		STATUS (through EAI Value Map "Oracle R11i Order Line Status")
Shipment Number	SHIPMENT_NUMBER		SHIPMENT_NUMBER
Scheduled Ship Date	SCHEDULE_SHIP_DATE		SCHEDULE_DATE
Hold Flag (hard-coded in inbound DTE)			

EAI Value Mappings for Oracle Applications

This section provides EAI Value Mappings for Siebel Applications to Oracle Applications integrations. Table 32 shows EAI Value Mappings for the following data types:

Oracle11i Account Category	Maps Siebel Applications account type to Oracle Applications customer class.
Oracle11i Account Phone	Maps Siebel Applications account field name (containing the phone number) to the Oracle Applications phone type.
Oracle11i Account Status	Maps Siebel Applications account status to Oracle Applications customer status.
Oracle11i Address Phone	Maps Siebel Applications address field name (containing the phone number) to the Oracle Applications phone type.
Oracle11i Contact Phone	Maps Siebel Applications contact field name (containing phone number) to the Oracle Applications phone type.
Oracle11i Contact Title	Maps Siebel Applications to Oracle Applications Contact title (Mr., Mrs., and so on).
Oracle11i Country	Maps Siebel Applications to Oracle Applications country values.
Oracle11i FOB Code	Maps Siebel Applications Freight Terms Info to Oracle Applications FOB Point.
Oracle11i Freight	Maps Siebel Applications to Oracle Applications freight terms.
Oracle11i Order Line Status	Maps Siebel Applications to Oracle Applications order status
Oracle11i Order Status	Maps Siebel Applications to Oracle Applications order status.
Oracle11i Order Type	Maps Siebel Applications to Oracle Applications order type.
Oracle11i Payment Type Code	Maps Siebel Applications to Oracle Applications payment type.
Oracle11i Sales Rep ID	Maps Siebel Applications User ID to Oracle Applications Sales Rep ID.
Oracle11i Ship Method Code	Maps Siebel Applications to Oracle Applications Order Shipment Carrier.
Oracle11i Shipment Priority	Maps Siebel Applications to Oracle Applications Order priority.
Oracle11i Tax Exempt Flag	Maps Siebel Applications to Oracle Applications Tax Exempt Flag.
Oracle11i Tax Exempt Reason	Maps Siebel Applications to Oracle Applications Tax Exempt reasons.
Oracle11i Unit of Measure	Maps Siebel Applications to Oracle Applications Product unit of measure.
Oracle11i User ID	Maps Siebel Applications to Oracle Applications User ID.

For each mapped data type (for example, Oracle Applications R11i Account Phone), the table shows a description, the Siebel User Interface, Siebel Business Component Field, Siebel List of Value Type, Oracle User Interface, Oracle Table Column, and Oracle Lookup Type table. If available, the Direction, Siebel Value, Siebel Display Value, Oracle Value, and Oracle Meaning are provided for each data value.

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
OracleR11i Account Category					
Description	Siebel account type to Oracle Applications customer class mapping				
Siebel User Interface	Type (Account)				
Siebel Buscomp.Field	Account.Type				
Siebel LOV Type	ACCOUNT_ TYPE				
Oracle User Interface	Category (Customer)				
Oracle Table.Column	HZ_PARTIES. CATEGORY_ CODE				
Oracle Lookup Type (Table)	CUSTOMER_ CATEGORY (AR_ LOOKUPS)				
Oracle Screen Navigation	(Receivables Manager responsibility) Setup > System > Quickcodes > Receivables				
	Siebel Outbound	Customer	Customer	CUSTOMER	Customer

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
OracleR11i Account Phone					
Description	Siebel account field name containing phone number mapping to Oracle phone type				
Siebel User Interface	Main Fax #, Main Phone # (Account)				
Siebel Buscomp.Field	Account.*				
Siebel LOV Type	N/A				
Oracle User Interface	Telephone Type (Customer)				
Oracle Table.Column	HZ_CONTACT_POINTS. PHONE_LINE_TYPE				
Oracle Lookup Type (Table)	PHONE_LINE_TYPE (FND_LOOKUP_VALUES) and COMMUNICATION_TYPE (FND_LOOKUP_VALUES)				
Oracle Screen Navigation	(Receivables Manager responsibility) Setup > System > Quickcodes > Receivables				
	Siebel Outbound	Main Fax Number	N/A	FAX	Fax
	Siebel Outbound	Main Phone Number	N/A	PHONE	Phone
OracleR11i Account Status					
Description	Siebel account status to Oracle customer status mapping				
Siebel User Interface	Status (Account)				
Siebel Buscomp.Field	Account. Account Status				
Siebel LOV Type	ACCOUNT_ STATUS				
Oracle User Interface	Status (Customer)				
Oracle Table.Column	HZ_CUST_ ACCOUNTS.STATUS				

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
Oracle Lookup Type (Table)	CUSTOMER_ STATUS (AR_ LOOKUPS)				
Oracle Screen Navigation	(Receivables Manager responsibility) Setup > System > Quickcodes > Receivables				
	Siebel Outbound	Active	Active	A	Active Only
	Siebel Outbound	Inactive	Inactive	I	Inactive Only
OracleR11i Address Phone					
Description	Siebel address field name containing phone number mapping to Oracle phone type				
Siebel User Interface	Not exposed in standard Account Address user interface				
Siebel Buscomp.Field	Business Address.*				
Siebel LOV Type	N/A				
Oracle User Interface	Address Telephone Type (Customer)				
Oracle Table.Column	HZ_CONTACT_POINTS. PHONE_LINE_TYPE				
Oracle Lookup Type (Table)	PHONE_LINE_ TYPE (FND_LOOKUP_VALUES) and COMMUNICATION_TYPE (FND_LOOKUP_VALUES)				
Oracle Screen Navigation	(Receivables Manager responsibility) Setup > System > Quickcodes > Receivables				
	Siebel Outbound	Fax Number	N/A	PHONE	Phone
	Siebel Outbound	Phone Number	N/A	PHONE	Phone

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
OracleR11i Contact Phone					
Description	Siebel contact field name containing phone number mapping to Oracle phone type				
Siebel User Interface	Work Phone #, Work Fax #, Assistant Phone #, Home Phone # (Contact). Alternate Phone # and Cellular Phone # not exposed in standard UI.				
Siebel Buscomp.Field	Contact.*				
Siebel LOV Type	N/A				
Oracle User Interface	Telephone Type (Customer Contact)				
Oracle Table.Column	HZ_CONTACT_POINTS. PHONE_LINE_TYPE				
Oracle Lookup Type (Table)	PHONE_LINE_ TYPE (FND_LOOKUP_VALUES) and COMMUNICATION_TYPE (FND_LOOKUP_VALUES)				
Oracle Screen Navigation	(Receivables Manager responsibility) Setup > System > Quickcodes > Receivables				
	Siebel Outbound	Alternate Phone #	N/A	PHONE	Phone
	Siebel Outbound	Assistant Phone #	N/A	PHONE	Phone
	Siebel Outbound	Cellular Phone #	N/A	PHONE	Phone
	Siebel Outbound	Fax Phone #	N/A	PHONE	Phone
	Siebel Outbound	Home Phone #	N/A	PHONE	Phone
	Siebel Outbound	Work Phone #	N/A	PHONE	Phone

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
OracleR11i Contact Title					
Description	Siebel to Oracle Contact title (Mr., Mrs., and so on) mapping				
Siebel User Interface	Mr./Ms. (Contact)				
Siebel Buscomp.Field	Contact.M/M				
Siebel LOV Type	MR_MS				
Oracle User Interface	Title (Customer Contact)				
Oracle Table.Column	RA_ CONTACTS. TITLE				
Oracle Lookup Type (Table)	CONTACT_ TITLE (AR_ LOOKUPS)				
Oracle Screen Navigation	(Receivables Manager responsibility) Setup > System > Quickcodes > Receivables				
	Siebel Outbound	Mr.	Mr.	MR	Mr.
	Siebel Outbound	Mrs.	Mrs.	MRS	Mrs.
	Siebel Outbound	Ms.	Ms.	MS	Ms.
	Siebel Outbound	Dr.	Dr.	DR	Dr.
	Siebel Outbound	Miss	Miss	MISS	Miss

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
OracleR11i Country					
Description	Siebel to Oracle country mapping				
Siebel User Interface	Country (Account Address)				
Siebel Buscomp.Field	Business Address. Country				
Siebel LOV Type	COUNTRY				
Oracle User Interface	Country (Customer Address)				
Oracle Table.Column	HZ_LOCATIONS.COUNTRY				
Oracle Lookup Type (Table)	FND_ TERRITORIES.TERRITORY_ CODE				
Oracle Screen Navigation	(Receivables Manager responsibility) - Setup - System - Countries				
	Siebel Outbound	USA	USA	US	United States
	Siebel Outbound	United Kingdom	United Kingdom	GB	United Kingdom

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
OracleR11i Fob Code					
Description	Siebel Freight Terms Info to Oracle FOB Point				
Siebel User Interface	Shipping Terms Info				
Siebel Buscomp.Field	Order Entry - Orders.Freight Terms Info				
Siebel LOV Type	Not configured in standard product. A picklist should be configured on this field to take advantage of this mapping.				
Oracle User Interface	FOB (Order)				
Oracle Table.Column	OE_ORDER_ HEADERS_ ALL.FOB_ POINT_CODE				
Oracle Lookup Type (Table)	FOB (AR_ LOOKUPS) (exposed through OE_ FOBS_ACTIVE_V)				
Oracle Screen Navigation	(Receivables Manager responsibility) Setup > System > Quickcodes > Receivables				
	Siebel Outbound	Destination	N/A	Destination	Destination
	Siebel Outbound	Origin	N/A	Origin	Origin
	Siebel Outbound	CIF	N/A	CIF	CIF
	Siebel Inbound	Destination	N/A	Destination	Destination
	Siebel Inbound	Origin	N/A	Origin	Origin
	Siebel Inbound	CIF	N/A	CIF	CIF

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
OracleR11i Freight					
Description	Siebel to Oracle freight terms mapping				
Siebel User Interface	Shipping Terms (Order)				
Siebel Buscomp.Field	Order Entry - Orders.Freight Terms				
Siebel LOV Type	FREIGHT_TERMS				
Oracle User Interface	Freight Terms (Order)				
Oracle Table.Column	OE_ORDER_ HEADERS_ ALL.FREIGHT_TERMS_ CODE				
Oracle Lookup Type (Table)	FREIGHT_TERMS (OE_ LOOKUPS)				
Oracle Screen Navigation	(Order Management Superuser responsibility) Setup > System > Quickcodes - Order Management				
	Siebel Outbound	Due	Due	Due	Prepay and Add
	Siebel Outbound	No Charge	No Charge	Paid	Prepaid
	Siebel Outbound	FOB	FOB	FOB	Freight on Board
	Siebel Outbound	TBD	TBD	TBD	To Be Determined
	Siebel Inbound	Due	Due	Due	Prepay and Add
	Siebel Inbound	No Charge	No Charge	Paid	Prepaid
	Siebel Inbound	FOB	FOB	FOB	Freight on Board
	Siebel Inbound	TBD	TBD	TBD	To Be Determined

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
OracleR11i Order Line Status					
Description	Siebel to Oracle order status mapping				
Siebel User Interface	Status (Order)				
Siebel Buscomp.Field	Order Entry - Orders.Status				
Siebel LOV Type	FS_ORDER_ STATUS				
Oracle User Interface	Status (Order)				
Oracle Table.Column	OE_ORDER_ LINES_ALL. FLOW_ STATUS_CODE				
Oracle Lookup Type (Table)	LINE_FLOW_ STATUS (OE_ LOOKUPS)				
Oracle Screen Navigation	(Order Management Superuser responsibility) Setup >System > Quickcodes >Receivables				
	Siebel Inbound	Booked	Booked	BOOKED	Booked
	Siebel Inbound	Open	Open	ENTERED	Entered
	Siebel Inbound	Pending	Pending	AWAITING_ SHIPPING	Awaiting Shipping
	Siebel Inbound	Cancelled	Cancelled	CANCELLED	Cancelled
	Siebel Inbound	Closed	Closed	CLOSED	Closed
	Siebel Inbound	Shipped	Shipped	INVOICED	Invoiced
	Siebel Inbound	Shipped	Shipped	INVOICED_P ARTIAL	Invoiced Partial
	Siebel Inbound	Shipped	Shipped	SHIPPED	Shipped
	Siebel Inbound	Closed	Closed	FULFILLED	Fulfilled

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
Oracle Screen Navigation (continued)	Siebel Inbound	Picked	Picked	PICKED	Picked
	Siebel Inbound	Booked	Booked	SCHEDULED	Scheduled
	Siebel Inbound	Picked	Picked	PICKED_PARTIAL	Picked partial
OracleR11i Order Status					
Description	Siebel to Oracle order status mapping				
Siebel User Interface	Status (Order)				
Siebel Buscomp.Field	Order Entry - Order Lines.Status				
Siebel LOV Type	FS_ORDER_ STATUS				
Oracle User Interface	Status (Order Line)				
Oracle Table.Column	OE_ORDER_ HEADERS_ ALL.FLOW_ STATUS_CODE				
Oracle Lookup Type (Table)	FLOW_ STATUS (OE_ LOOKUPS)				
Oracle Screen Navigation	(Order Management Superuser responsibility) Setup > System > Quickcodes > Order Management				
	Siebel Outbound	Booked	Booked	BOOKED	Booked
	Siebel Outbound	Open	Open	ENTERED	Entered
	Siebel Inbound	Booked	Booked	BOOKED	Booked
	Siebel Inbound	Open	Open	ENTERED	Entered
	Siebel Inbound	Cancelled	Cancelled	CANCELLED	Cancelled
	Siebel Inbound	Closed	Closed	CLOSED	Closed

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
OracleR11i Order Type					
Description	Order type mapping				
Siebel User Interface	Type (Order)				
Siebel Buscomp.Field	Order Entry - Orders.Order Type --> Order Entry - Order Types.Order Type				
Siebel LOV Type	FS_ORDER_ TYPE				
Oracle User Interface	Type (Order)				
Oracle Table.Column	OE_ORDER_ HEADERS_ ALL.ORDER_ TYPE_ID > OE_ORDER_ TYPES_115_ ALL.ORDER_ TYPE_ID > OE_ORDER_ TYPES_115_ ALL.NAME				
Oracle Lookup Type (Table)	N/A				
	Siebel Outbound	eSales Order	eSales Order	Mixed	Order and Return Lines
	Siebel Outbound	Sales Order	Sales Order	Mixed	Order and Return Lines

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
OracleR11i Payment Type Code					
Description	Payment Type mapping				
Siebel User Interface	Payment Method (Order)				
Siebel Buscomp.Field	Order Entry - Orders.Payment Method				
Siebel LOV Type	PAYMENT_METHOD				
Oracle User Interface	Payment Type (Order)				
Oracle Table.Column	OE_ORDER_ HEADERS_ ALL.PAYMENT_TYPE_CODE				
Oracle Lookup Type (Table)	PAYMENT TYPE (OE_ LOOKUPS)				
Oracle Screen Navigation	(Order Management Superuser responsibility) Setup > System > Quickcodes >Order Management				
	Siebel Outbound	Cash	N/A	CASH	Cash
	Siebel Outbound	Check	N/A	CHECK	Check
	Siebel Outbound	Credit	N/A	CREDIT_CARD	Credit Card
	Siebel Inbound	Cash	N/A	CASH	Cash
	Siebel Inbound	Check	N/A	CHECK	Check
	Siebel Inbound	Credit	N/A	CREDIT_CARD	Credit Card

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
OracleR11i Sales Rep ID					
Description	Siebel user ID to Oracle Sales Rep ID mapping				
Siebel User Interface	Team (Order)				
Siebel Buscomp.Field	Order Entry - Orders.Primary Position Contact Id				
Siebel LOV Type	N/A				
Oracle User Interface	Salesperson (Order)				
Oracle Table.Column	OE_ORDER_ HEADERS_ ALL. SALESREP_ID > RA_ SALESREPS_ ALL. SALESREP_ID				
Oracle Lookup Type (Table)	N/A				
	Siebel Outbound	0 - 1	SADMIN	1449	
OracleR11i Ship Method Code					
Description	Order Shipment Carrier mapping				
Siebel User Interface	Ship Carrier (Order), Ship Carrier (Order Line)				
Siebel Buscomp.Field	Order Entry - Orders.Carrier Type, Order Entry - Line Items.Carrier Code				
Siebel LOV Type	FS_CARRIER				
Oracle User Interface	Freight Carrier (Order)				
Oracle Table.Column	OE_ORDER_ HEADERS_ ALL.SHIPPING_METHOD_ CODE, OE_ORDER_ LINES_ALL. SHIPPING_ METHOD_ CODE				
Oracle Lookup Type (Table)	ORG_FREIGHT.FREIGHT_ CODE				

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
Oracle Screen Navigation	(Order Management Superuser responsibility) Setup > Shipping > Freight > Define Freight Carriers				
	Siebel Outbound	Airborne	Airborne	AIRBORNE	Airborne
	Siebel Outbound	DHL	DHL	DHL	DHL
	Siebel Outbound	Emery	Emery	EMERY	Emery
	Siebel Outbound	Fedex	Federal Express	Federal Express	Federal Express
	Siebel Outbound	UPS	UPS	UPS	UPS
	Siebel Outbound	US Mail	US Mail	USMAIL	US Mail
	Siebel Inbound	Airborne	Airborne	AIRBORNE	Airborne
	Siebel Inbound	DHL	DHL	DHL	DHL
	Siebel Inbound	Emery	Emery	EMERY	Emery
	Siebel Inbound	Fedex	Federal Express	Federal Express	Federal Express
	Siebel Inbound	UPS	UPS	UPS	UPS
	Siebel Inbound	US Mail	US Mail	USMAIL	US Mail

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
OracleR11i Shipment Priority					
Description	Order priority mapping				
Siebel User Interface	Priority (Order)				
Siebel Buscomp.Field	Order Entry - Orders.Order Priority				
Siebel LOV Type	FS_INVLOC_ ORDPRI				
Oracle User Interface	Shipment Priority (Order)				
Oracle Table.Column	OE_ORDER_ HEADERS_ ALL. SHIPMENT_ PRIORITY_ CODE				
Oracle Lookup Type (Table)	SHIPMENT_ PRIORITY (OE_ LOOKUPS)				
Oracle Screen Navigation	(Order Management responsibility) Setup > System > Quickcodes > Receivables				
	Siebel Outbound	High	High	High	High Priority
	Siebel Outbound	Medium	Medium	Standard	Standard Priority
	Siebel Inbound	High	High	High	High Priority
	Siebel Inbound	Medium	Medium	Standard	Standard Priority

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
OracleR11i Tax Exempt Flag					
Description	Tax Exempt status mapping				
Siebel User Interface	Tax Exempt (Order)				
Siebel Buscomp.Field	Order Entry - Orders.Tax Exempt				
Siebel LOV Type	Boolean Y/N				
Oracle User Interface	Tax Exemption Tax (Order)				
Oracle Table.Column	OE_ORDER_ HEADERS_ ALL.TAX_ EXEMPT_FLAG				
Oracle Lookup Type (Table)	S (Standard), E (Exempt), R (Require)				
	Siebel Inbound	N	Unchecked	S	Standard
	Siebel Outbound	N	Unchecked	S	Standard

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
OracleR11i Tax Exempt Reason					
Description	Tax Exempt Reason mapping				
Siebel User Interface	Not exposed in standard User Interface				
Siebel Buscomp.Field	Order Entry - Orders.Tax Exempt Reason				
Siebel LOV Type	Boolean Y/N				
Oracle User Interface	Tax Exemption: Reason (Order)				
Oracle Table.Column	OE_ORDER_ HEADERS_ ALL.TAX_ EXEMPT_ REASON_ CODE				
Oracle Lookup Type (Table)	TAX_REASON (AR_LOOKUPS)				
Oracle Screen Navigation	(Receivables Manager responsibility) Setup > System > Quickcodes > Receivables				
	Siebel Outbound	Education	N/A	EDUCATION	education
	Siebel Outbound	Hospital	N/A	HOSPITAL	hospital
	Siebel Outbound	Manufacturer	N/A	MANUFACTURER	manufacturer
	Siebel Inbound	Education	N/A	EDUCATION	education
	Siebel Inbound	Hospital	N/A	HOSPITAL	hospital
	Siebel Inbound	Manufacturer	N/A	MANUFACTURER	manufacturer

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
OracleR11i Unit of Measure					
Description	Product unit of measure mapping				
Siebel User Interface	U/M (Product)				
Siebel Buscomp.Field	Order Entry - Line Items.Product Unit of Measure, Internal Product.Unit of Measure				
Siebel LOV Type	UNIT_OF_ MEASURE				
Oracle User Interface	Unit (Order Line), Primary Unit of Measure (Item)				
Oracle Table.Column	OE_ORDER_ LINES_ALL. ORDER_ QUANTITY_ UOM, MTL_ SYSTEM_ ITEMS. PRIMARY_ UOM_CODE				
Oracle Lookup Type (Table)	MTL_UNITS_ OF_MEASURE.UOM_CODE				

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
Oracle Screen Navigation	(Order Management responsibility) Setup - UOM - Units				
	Siebel Inbound	Dozen	Dozen	DZ	Dozen
	Siebel Inbound	Each	Each	EA	Each
	Siebel Inbound	Gram	Gram	G	Gram
	Siebel Inbound	Kilogram	Kilogram	KG	Kilogram
	Siebel Inbound	Litre	Litre	L	Liter
	Siebel Inbound	Ounce	Ounce	OZ	Ounce
	Siebel Inbound	Pound	Pound	LBS	Pound
	Siebel Inbound	Day	Day	DAY	Day
	Siebel Inbound	Foot	Foot	FT	Foot
	Siebel Inbound	Hour	Hour	HR	Hour
	Siebel Inbound	Dollar	Dollar	USD	US Dollar

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
Oracle Screen Navigation (continued)	Siebel Inbound	Year	Year	YR	Year
	Siebel Inbound	Bag	Bag	BAG	Bag
	Siebel Inbound	Case	Case	CSE	Case
	Siebel Inbound	Half Pallet	Half Pallet	HPL	Half Pallet
	Siebel Inbound	Pallet	Pallet	PL	Pallet
	Siebel Inbound	Thousand	Thousand	BTH	Box of 1000
	Siebel Inbound	Unit	Unit	UNT	Unit
	Siebel Inbound	Dozen	Dozen	DOZ	Dozen
	Siebel Inbound	Foot	Foot	FTP	Foot
	Siebel Inbound	Kilogram	Kilogram	KGM	Kilogram
	Siebel Inbound	Pound	Pound	LB	Pound
	Siebel Inbound	Ounce	Ounce	OUZ	Ounce
OracleR11i User ID					
Description	Siebel to Oracle User ID mapping				
Siebel User Interface	Help > About Record				
Siebel Buscomp.Field	Created By, Updated By, and so on > S_USER.ROW_ID				
Siebel LOV Type	N/A				
Oracle Table.Column	CREATED_BY, LAST_UPDATED_BY, and so on > FND_USER. USER_ID				

Table 32. Siebel Application to Oracle Integration EAI Value Maps

	Direction	Siebel Value	Siebel Display Value	Oracle Value	Oracle Meaning
Oracle Lookup Type (Table)	N/A				
Oracle Screen Navigation	(System Administrator responsibility) Security > User > Define				
	Siebel Outbound	0-1	SADMIN	1000112	N/A

B Oracle Scripts

This appendix describes Oracle scripts.

Oracle Applications Scripts

The Oracle Applications scripts listed here are supplied with the Siebel Connector for Oracle Applications.

- siebelaccount11ialert_insert_hz_cust_accounts.sql

  This script creates an insert alert, Siebel Customer Account Insert, which puts an entry into the Notification Table for every new record created in table HZ_CUST_ACCOUNTS.

- siebelaccount11ialert_insert_hz_cust_acct_sites_all.sql

  This script creates an insert alert, Siebel Customer Site Insert, which puts an entry into the Notification Table for every new record created in table HZ_CUST_ACCT_SITES_ALL.

- siebelaccount11ialert_insert_hz_org_contacts.sql

  This script creates an insert alert, Siebel Customer Contact Insert, which puts an entry into the Notification Table for every new record created in table HZ_ORG_CONTACTS.

- siebelaccount11ialert_update_ra_customers_interface_all.sql

  This script creates an update alert, Siebel Customer Interface Header Update, which puts an entry into the Notification Table for any update in table RA_CUSTOMERS_INTERFACE_ALL. This action puts interface_status into Siebel Applications if the records are not successfully created in Oracle Applications by the Customer Import.

- siebelaccount11iprocedure_sie_customer_preval_pr.sql

  This script creates the pre-validation procedure, SIE_CUSTOMER_PREVAL_PR11i, which validates the data inserted into the Oracle Customer Interface tables.

- siebelaccount11itable_sie_contact_phones_int_all.sql

  This script creates the table, SIE_CONTACT_PHONES_INT_ALL, which is used in the Accounts Outbound track of the Siebel Connector for Oracle Applications. The data is inserted into this table before the pre-validation program validates it and moves it into the Oracle interface table.

- siebelaccount11itable_sie_customer_profiles_int_all.sql

  This script creates the table, SIE_CUSTOMER_PROFILES_INT_ALL, which is used in the Accounts Outbound track of the Siebel Connector for Oracle Applications. The data is inserted into this table before the pre-validation program validates it and moves it into the Oracle interface table.

- siebelaccount11itable_sie_customers_interface_all.sql

  This script creates the table, SIE_CUSTOMER_INTERFACE_ALL, which is used in the Accounts Outbound track of theSiebel Connector for Oracle Applications. The data is inserted into this table before the pre-validation program validates it and moves it into the Oracle interface table.

- siebelaccount11iview_sie_addresses_v.sql

  This script creates the view, SIE_ADDRESSES_V, which is used in the Accounts Inbound track of the Siebel Connector for Oracle Applications.

- siebelaccount11iview_sie_contacts_v.sql

  This script creates the view, SIE_CONTACTS_V, which is used in the Accounts Inbound track of the Siebel Connector for Oracle Applications.

- siebelgeneral11iprocedure_sie_insert_or_update_11i.sql

  This script creates the procedure, SIE_INSERT_OR_UPDATE, which is executed by all the alerts to insert records into the Notification Table.

- siebelgeneral11iprocedure_sie_insert_or_update_11i2.sql

  This script creates the procedure, SIE_INSERT_OR_UPDATE2, which is executed by all the alerts on the Oracle interface tables to insert records into the Notification Table.

- siebelgeneral11iscript_notify_invloc.sql

  This script can be used to insert records into the Notification Table to pull all the Inventory Organizations from Oracle Applications into Siebel Applications as an initial data load of Inventory Locations.

- siebelgeneral11iscript_notify_org.sql

  This script can be used to insert records into the Notification Table to pull all the Operating Units from Oracle Applications into Siebel Applications as an initial data load of Organizations.

- siebelgeneral11iscript_notify_product.sql

  This script can be used to insert records into the Notification Table to pull all the Items from Oracle Applications into Siebel Applications as an initial data load of products.

- siebelgeneral11isequence_sie_notify_s.sql

  This script creates the sequence SIE_NOTIFY_S, which is used to generate unique SEQ_NUM in the Notification Table.

- siebelgeneral11isetup.sql

  This script is run at the SQL prompt after logging into the database as the super user (usually APPS). This script calls other the setup scripts listed below, creates all the Siebel Connector for Oracle Applications Objects, creates Synonyms, and grants the necessary privileges to the APPS user and the SIEBEL user.

- siebelgeneral11isetup1.sql

  This script grants the necessary privileges to SIEBEL from APPS on Oracle Applications base tables that will be accessed by the SIEBEL user.

- siebelgeneral11isetup2.sql

  This script creates necessary Synonyms and all the Siebel Connector for Oracle Applications objects under the SIEBEL user and also grants necessary privileges to the APPS user on the Siebel Connector for Oracle Applications objects.

- siebelgeneral11isetup3.sql

  This script creates necessary synonyms, alerts, and triggers for the Connector under the APPS user.

- siebelgeneral11itable_sie_notify_tbl.sql

  This script creates the Notification Table SIE_NOTIFY_TBL.

- siebelgeneral11itable_sie_obj_priority_tbl.sql

  This script creates the priority table SIE_OBJ_PRIORITY_TBL and also the records that indicate the priority in which the records are to be processed.

- siebelinventorylocation11iview_sie_invloc_v.sql

  This script creates the view SIE_INVLOC_V, which is used in the InvLoc Inbound track of the Oracle11i Applications Connector.

- siebelorder11ialert_insert_oe_order_header_all.sql

  This script creates an insert alert, Siebel Order Header Insert, which puts an entry into the Notification Table for every new record created in table OE_ORDER_HEADERS_ALL.

- siebelorder11ialert_insert_oe_order_hold_all.sql

  This script creates an insert alert, Siebel Order Hold Insert, which puts an entry into the Notification Table for every new record created in table OE_ORDER_HOLDS_ALL.

- siebelorder11ialert_insert_oe_order_lines_all.sql

  This script creates an insert alert, Siebel Order Lines Insert, which puts an entry in the Notification Table for every new record created in table OE_ORDER_LINES_ALL.

- siebelorder11ialert_update_oe_order_header_all.sql

  This script creates an insert alert, Siebel Order Header Update, which puts an entry in the Notification Table for any update in table OE_ORDER_HEADERS_ALL.

- siebelorder11ialert_update_oe_order_hold_all.sql

  This script creates an insert alert, Siebel Order Hold Update, which puts an entry in the Notification Table for any update in table OE_ORDER_HOLDS_ALL.

- siebelorder11ialert_update_oe_order_lines_all.sql

  This script creates an insert alert, Siebel Order Lines Update, which puts an entry in the Notification Table for every new record created in table OE_ORDER_LINES_ALL.

- siebelorder11iprocedure_order_process11i.sql

  This script creates the pre-validation procedure, SIEORDERVALIDATE11i, which validates the data inserted into the Oracle Order Interface tables.

- siebelorder11itable_sie_order_headers_all.sql

  This script creates the table SIE_ORDER_HEADERS_ALL, which is used in the Orders Outbound track of the Oracle11i Applications Connector. The data is inserted into this table before the pre-validation program validates it and moves it into the Oracle interface table.

- siebelorder11itable_sie_order_lines_all.sql

  This script creates the table, SIE_ORDER_LINES_ALL, which is used in the Orders Outbound track of the Oracle11i Applications Connector. The data is inserted into this table before the pre-validation program validates it and moves it into the Oracle interface table.

- siebelorder11itrigger_delete_oe_order_headers_all.sql

  This script creates an after delete trigger, SIE_ORDER_HEADER_DEL_TR, which puts an entry in the Notification Table for any records deleted in table OE_ORDER_HEADERS_ALL.

- siebelorder11itrigger_delete_oe_order_lines_all.sql

  This script creates an after delete trigger, SIE_ORDER_LINE_DEL_TR, which puts an entry in the Notification Table for any records deleted in table OE_ORDER_LINES_ALL.

- siebelorder11iview_sie_headers_all_v.sql

  This script creates the view, SIE_HEADERS_ALL_V, which is used in the Orders Inbound track of the Siebel Connector for Oracle Applications.

- siebelorder11iview_sie_lines_all_v.sql

  This script creates the view, SIE_LINES_ALL_V, which is used in the Orders Inbound track of the Siebel Connector for Oracle Applications.

- siebelorganization11iview_sie_org_v.sql

  This script creates the view, SIE_ORG_V, which is used in the Organizations Inbound track of the Siebel Connector for Oracle Applications.

- siebelproduct11ialert_insert_mtl_system_items.sql

  This script creates an insert alert, Siebel Product Insert, which puts an entry in the Notification Table for every new Customer Order Enabled – Standard or Model Item created in table MTL_SYSTEM_ITEMS_B.

- siebelproduct11ialert_update_mtl_system_items.sql

  This script creates an update alert, Siebel Product Update, which puts an entry into the Notification Table for any update to a Standard or Model Item in table MTL_SYSTEM_ITEMS_B.

- siebelproduct11itrigger_update_mtl_system_items.sql

  This script creates an after update trigger, SIE_UPDATE_ITEMS_TR, which puts an entry into the Notification Table for any update in table MTL_SYSTEM_ITEMS.

  There is also a trigger for updates on table, MTL_SYSTEM_ITEMS_B. By default, the alert, is created when Siebel Connector for Oracle Applications objects are created. The trigger improves efficiency by checking the old and new customer order enabled flag values without inserting entries into the Notification Table for these items. Optionally, you can delete the alert, and enable the trigger.

- siebelproduct11iview_sie_prod_invloc_v.sql

  This script creates the view, SIE_PROD_INVLOC_V, which is used in the Products Inbound track of the Siebel Connector for Oracle Applications.

- siebelproduct11iview_sie_prod_org_v.sql

  This script creates the view, SIE_PROD_ORG_V, which is used in the Products Inbound track of the Siebel Connector for Oracle Applications.

- siebelproduct11iview_sie_product_v.sql

  This script creates the view, SIE_PRODUCT_V, which is used in the Price Lists Inbound track of Oracle's Siebel Connector for Oracle Applications.

Index

A

Account Management processes 10
Account Outbound integration 22
accounts
creating 43
data integration 43
EAMs and 56
integration and 55
loading from Oracle 36
updating 44
working with 43
administrative tasks, listed 37
application integrations
See Siebel/Oracle integrations
architecture, Oracle Applications Connector 12
arguments
Database Adapter 69
Delete 70
Execute 72
Oracle Procedure Adapter 84
Oracle Receiver 78
Query 70
Synchronize 71
Upsert 71
attribute mappings 57
Automatic Site Numbering 22

B

BIM (Business Integration Manager) 15, 30
buffer tables 16, 63
Business Integration Manager (BIM) 15, 30
business processes
automating 10
data synchronization 11
Business Service Editor 14

C

CascadeDelete property 70
configuration activities 20
configuring
Business Integration Manager 30
Oracle Application 22
Oracle Receiver 29
Siebel Mobile Web Client 51
Siebel Tools 51
County column 22
Customer Service processes 10

D

data
inbound, Oracle to Siebel 18
loading 34
outbound, Siebel to Oracle 17
synchronizing 11
verifying load of 36
data integrations
See also Siebel/Oracle integrations; workflows
about 46
accounts 43, 55
between Siebel and Oracle applications 17
discounts 55
inbound 38
interdependencies 54
inventory locations 54
monitoring 37
orders 45, 56
organizations 54
outbound 37
overview 52
price lists 55
products 54
recommended sequence 54
Data Mapper 14
Data Transformation Engine (DTE)
Business Service Editor and 14
described 14, 52
passing Oracle operations field to 74
data transformation maps 14
data types 73
Database (DB) Adapter
about 68
arguments 69
described 16
error handling 74
generating SQL statements 72
methods 69
passing connection parameters 73
Database Wizard 14
databases
See also Oracle databases
local databases 12

Siebel databases 34
DB Adapter
See Database Adapter
Delete method 70
delete processing, Oracle Receiver 80
developer environment, installing 47
discount line items 55
discounts
integration and 55
DTE
See Data Transformation Engine

E

EAI ODBC Service
See Oracle Procedural Adapter
EAI Siebel Adapter 16
EAI SQL Adapter
See Database (DB) Adapter
EAI value maps
creating 33
described 33, 53
Oracle R11i applications 97
end users
working with accounts 43
working with orders 44
entity attribute mappings (EAMs)
Oracle R11i applications 85
Siebel/Oracle integrations 56
entity mappings 57
errors, Database Adapter 74
eScript 26, 27, 29, 49
Execute method 72
ExecuteProcedure method 84

F

FndAPI procedure 84
foreign keys 82

I

installing Oracle Applications Connector
configuration activities 20
configuring Oracle Application 22
installation activities 20
installing developer environment 47
installing Oracle Application 22
installing Oracle Client 25, 48
installing Siebel Mobile Web Client 48
installing Siebel Server 25
installing Siebel Tools 48
interdependencies 19
overview 19
preliminary tasks 21, 23
procedure for 24, 26, 48
verifying installation 26
verifying registration of ODBC driver 48
Integration ID column 36
Integration Object Editor 13
Integration Object Wizards 13
See also Oracle Applications Wizard
integration objects
described 13, 52
metadata 81
working with 81
integration points 13
integration workflows 15
interface tables 84
inventory locations
creating, Oracle 35
EAMs and 56
integration and 54
Invoicing processes 10

L

loading Oracle application data 34
local databases 12

M

mappings
EAI value, Oracle R11i 97
EAM, Oracle R11i 85
metadata
integration objects 81
Oracle applications 81
meta-operator field 74
methods
Database Adapter 69
Oracle Procedure Adapter 84
Oracle Receiver 78
mobile user support 11

N

network connectivity 23
notification tables
arguments 78
columns 77
described 16
methods 78
Oracle Receiver and 74, 76
structure 77

O

ODBC API 73
ODBC connections 73
ODBC cursors 73
ODBC data source

creating 27, 29, 49
testing connection 28, 50
ODBC data types 73
ODBC driver 48
operation codes 72
Oracle alerts
about 15
Oracle Application 32
SIE_UPDATE_ITEMS_TR trigger 59
Oracle Application
See also Oracle applications
inbound integrations 38
installing 22
integrating with Siebel 8 Application 11
loading accounts from 36
loading data 34, 35
loading orders from 36
loading products from 36
monitoring integrations 37
Oracle alerts setup 32
outbound integrations 37
setting up connector objects 31
Oracle applications
See also Oracle Application
activating workflows for 31
EAI value maps 33
integrating data with Siebel applications 17
loading data 34
metadata 81
verifying data load 36
Oracle Applications Connector
about 9
administrative tasks 37
architecture 12
features 10
mobile user support 11
predefined integration points 10
user ID for 24
using 9
Oracle Applications Connector Objects
setting up, Oracle 31
Oracle Applications Server 21, 22, 31
Oracle Applications Wizard
data types and 73
described 13
integration objects 81
operational summary 80
running 82
working with 80
Oracle Client, installing 25, 48
Oracle databases
administrators 37
creating logins 24
sending account updates to 44
sending new accounts to 43
sending order updates to 45
sending orders to 45
verifying order status 46
Oracle interface tables 62
Oracle Procedural Adapter 16, 83
Oracle R11i Application
EAI value mappings 97
EAM mappings 85
scripts 119
Oracle Receiver
about 17, 74
arguments 78
configuring 29
delete processing 80
dispatch workflow calls 76
methods 78
operational summary 75
parameters 79
running 74
starting 30
workflows 75, 76
Oracle scripts
loading inventory organizations 59
loading Oracle operating units 58
Oracle R11i Application 119
running 31
verifying installation of 25, 26
Oracle tables 82
Oracle Wizard Get Children object 82
Oracle Wizard Get Table All object 82
Oracle Wizard Get Tables object 82
orders
creating 45
data integration 45
EAMs and 56
integration and 56
loading from Oracle 36
updating 45
verifying 46
working with 44
organizations
creating, Oracle 35
EAMs and 56
integration and 54
Outbound Order integration 23

P

parameters, Oracle Receiver 79
parent-child relationships 69
passwords, defaults 24
price lists
integration and 55

Product Management processes 10
products
EAMs and 56
integration and 54
loading from Oracle 36

Q

QBE (Query By Example) instance 69, 70
queries, SQL Adapter 82
Query By Example (QBE) instance 69, 70
Query method 69

R

RA_TERMS table 36
run-time architecture 17

S

S_PAYMENT_TERM table 36
server requirements 21
setup scripts 32
SIE_NOTIFY_TBL 75, 76
SIE_UPDATE_ITEMS_TR 59
Siebel 8 Application
integrating with Oracle Application 11
Siebel Adapter 16
Siebel applications
creating accounts in 43
creating orders in 45
EAI value maps 33
integrating data with Oracle applications 17
updating accounts in 44
updating orders in 45
verifying orders in 46
Siebel database 34
Siebel Enterprise Server 21
Siebel Gateway Server 21
Siebel Integration Workflows 52
Siebel Mobile Web Client
editing configuration file for 51
installing 48
local databases 11
Siebel Server
creating ODBC data source on 27, 29, 49
installing 25
Siebel Tool Wizards 13
Siebel Tools
configuring 51
installing 48
Siebel Wizard 13
Siebel Workflow Designer 52
Siebel/Oracle integrations
See also data integrations
EAMs 56
workflows 57
site use changes 60
software requirements 21
SQL Adapter 82
SQL integration objects
See integration objects
SQL statements, generating 72
SQL Wizard 73
SQL*Net database connectivity 22, 23
stored procedures 83
Synchronize method 71
synchronizing data 11
system administrators 37

T

tools.cfg file 82
transactions 73

U

Upsert method 71
user IDs 24
user names 24

W

wizards
Database Wizard 14
Integration Object Wizards 13
Oracle Applications Wizard 13, 73, 80
Siebel Tool Wizards 13
Siebel Wizard 13
SQL Wizard 73
workflows
activating 30
dispatching calls 76
integration workflows 15
Oracle Receiver and 75, 76
Siebel Integration Workflows 52
Siebel Workflow Designer 52
Siebel/Oracle integrations 57