# Oracle® Configurator and SellingPoint

Administration Guide

Release 11*i* & 4.2.2

May 2000

Part No.  A73282-02

The administration tasks presented here are required to set up and support developers and users of Oracle Configurator windows and the Oracle SellingPoint application.

Oracle Configurator and SellingPoint Administration Guide, Release 11*i* & 4.2.2

Part No. A73282-02

Primary Authors: Denise Boyer

Contributing Authors: Tina Brand

Contributors: Ashot Khachatryan, Serge Kudravtsev, Sarit Manna, Helen Reznik, Mark Sawtelle, Michael Sheehy, Alexander Siaston, Andrew Wolfe, Lois Wortley

# Contents

## 2 Client/Server Administration

## 3 The Oracle Configurator Schema

## 4 Data Transfer to the CZ Schema

## 5 Data Transfer from the CZ Schema

## 6 Pricing in Oracle Configurator

## 7 Installing the Servlet

## 8 Oracle Configurator Deployment

## A Import Tables

## B Export Tables

## C OC SQL*Plus Scripts and Procedures

## Glossary of Terms

## List of Figures

# List of Tables

# Send Us Your Comments

**Oracle Configurator and SellingPoint Administration Guide, Release 11*i***

**Part No. A73282-02**

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the chapter, section, and page number (if available). You can send comments through a call to Oracle Support Services or by sending them to:

Oracle Configurator
Oracle Corporation
Documentation
21 North Avenue
Burlington, MA 01803
USA

If you would like a reply, please give your name, address, telephone number, and (optionally)electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

# Preface

This *Oracle Configurator and SellingPoint Administration Guide* provides explanations, descriptions, and instructions for the administration tasks required to set up and support development and deployment of an Oracle Configurator window or the Oracle SellingPoint application.

## Intended Audience

This manual is intended for system administrators and database administrators who are supporting Oracle Configurator developers and end users. Anyone responsible for supporting use of Oracle Configurator (OC) should read this book. That includes supporting the application development environment (Oracle Configurator Developer) as well as the application run-time environment that is created.

Ordinarily, the tasks presented in this book are performed by a Database Administrator (DBA) or an Oracle Configurator administrator with DBA experience.

## Structure

This manual contains:

- Chapter 1, "Introduction"
- Chapter 2, "Client/Server Administration"
- Chapter 3, "The Oracle Configurator Schema"
- Chapter 4, "Data Transfer to the CZ Schema"
- Chapter 5, "Data Transfer from the CZ Schema"

- Chapter 6, "Pricing in Oracle Configurator"
- Chapter 7, "Installing the Servlet"
- Chapter 8, "Oracle Configurator Deployment"
- Appendix A, "Import Tables"
- Appendix B, "Export Tables"
- Appendix C, "OC SQL*Plus Scripts and Procedures"
- "Glossary of Terms"
- "Glossary of Acronyms"

## Related Documents

For more information, see the documentation for your release of Oracle Applications, Release 8i Oracle RDBMS documentation, Oracle Configurator documentation, and the product-specific *Release Notes* for releases supported to work with Oracle Configurator.

## Conventions

In examples, an implied carriage return occurs at the end of each line, unless otherwise noted. You must press the Return key at the end of a line of input.

The following conventions are also used in this manual:

| Convention | Meaning |
|---|---|
| .<br>.<br>. | Vertical ellipsis points in an example mean that information not directly related to the example has been omitted. |
| . . . | Horizontal ellipsis points in statements or commands mean that parts of the statement or command not directly related to the example have been omitted. |
| **boldface text** | Boldface type in text indicates a term defined in the text, the glossary, or in both locations. |
| < > | Angle brackets enclose user-supplied parameters or arguments. See Section C.2.1, "Script Arguments", on page C-19 for a description of common arguments used throughout this document. |
| [ ] | Brackets enclose optional clauses from which you can choose one or none. |

# 1

# Introduction

Oracle Configurator (OC) consists of the Oracle Configurator window, the Oracle SellingPoint application, Oracle Configurator Developer, and the Oracle Configurator schema (CZ). The Oracle Configurator window and Oracle Configurator schema are installed with Oracle Applications Release 11*i*. Oracle Configurator Developer is installed from the Oracle Configurator Developer CD. The Oracle SellingPoint application is also installed from the Oracle Configurator Developer CD.

This book presents all the basic administrative tasks necessary for supporting use of an Oracle Configurator window in the following general environments:

- Oracle Applications Release 11*i* with an Oracle Configurator window add-on in Order Management, Telesales, iStore, Sales Online or Order Capture
- A custom web application using Oracle Configurator.
- Oracle SellingPoint application as a client/server deployment.
- Oracle SellingPoint application as a mobile deployment synchronizing with a Release 11*i* Oracle Applications database.
- Any Oracle runtime configurator using imported Oracle Applications Release 11.0 or 10.7, or non-Oracle or legacy data."
- Any Oracle runtime configurator exporting data to a Release 10.7 or 11.0 Oracle Applications database for use by Oracle Order Entry.
- Oracle Configurator Developer

# 1.1 Oracle Runtime Configurators

Throughout this book, references to an Oracle runtime configurator imply both Oracle Configurator windows embedded in other applications and a standalone Oracle SellingPoint application.

***Table 1–1 Availability of Oracle Runtime Configurators***

| | Oracle Configurator Release 11*i* | Oracle SellingPoint Release 4.2.x |
|---|---|---|
| Oracle Configurator Window (Java applet) | ▪ Test configurator in Oracle Configurator Developer<br>▪ Production | ▪ Test configurator in Oracle Configurator Developer |
| Oracle Configurator Window (DHTML) | ▪ Test configurator in Oracle Configurator Developer<br>▪ Production | ▪ Test configurator in Oracle Configurator Developer |
| Oracle SellingPoint Application (Client/Server) | ▪ Test configurator in Oracle Configurator Developer, only | ▪ Test configurator in Oracle Configurator Developer<br>▪ Production |
| Oracle SellingPoint Application (Mobile) | ▪ Production | ▪ Production |

An Oracle runtime configurator is an end-user environment for configuring products and services. Implementers or developers of an Oracle runtime configurator use Oracle Configurator Developer to develop configuration models and user interface customizations. You can test your configuration models in test instances of these runtime configurators from within Oracle Configurator Developer.

The Oracle Configurator Toolkit is available for custom internet deployments.

### Oracle Configurator Window

The Oracle Configurator window is deployed within Oracle Applications or a custom web application as either a Java applet or a DHTML window running in a browser.

### Oracle SellingPoint Application

The Oracle SellingPoint application is available for deploying as a standalone client/server or mobile configurator.

## 1.2 Oracle Configurator Schema

The Oracle Configurator schema is part of the Oracle Applications database that is used by Oracle Configurator. When using existing Oracle Applications data to define configuration models, the data must be transferred into the Oracle Configurator schema for use by Oracle Configurator Developer and the runtime configurator. Data transfers within the Release 11*i* Oracle Applications database are completed by using Oracle Applications concurrent programs. SQL*Plus scripts available on the Oracle Configurator Developer CD provide import and export capabilities for:

- data imports from Release 10.7 and 11.0 Oracle Applications databases or non-Oracle legacy databases.
- data exports from Release 11*i* Oracle Configurator to Release 10.7 and 11.0 Oracle Applications databases.

## 1.3 Overview of Oracle Configurator Administration

Using Oracle Configurator (OC) requires administrative support. This document presents specific instructions for performing many of the administrative tasks required to set up and support development and deployment of an Oracle runtime configurator, including the following categories of administrative tasks:

- Client/Server connectivity for Oracle Configurator Developer and a standalone Oracle SellingPoint application
- Preparing data for transfer or import into the Oracle Configurator schema
- Transferring data within Oracle Applications database, Release 11*i*
- Importing data from Oracle Applications Release 10.7 or 11.0
- Importing data from non-Oracle or legacy databases
- Exporting data from the Oracle Configurator schema to Oracle Order Entry or Oracle Receivables (Release 10.7 or 11.0)
- Enabling end-user access to an Oracle runtime configurator
- Oracle Applications setup for using an Oracle Configurator window
- Installing the UI servlet for using an Oracle Configurator window
- Deployment of an Oracle runtime configurator
- Mobile deployment of an Oracle SellingPoint application

This book does *not* present instructions for

- Establishing any privileges to any user, including DBA privileges
- Establishing a database instance such as <appssid>
- Installing Oracle Applications for using the Oracle Configurator window
- Installing and setting up Oracle8*i* Enterprise Edition RDBMS

See other relevant Oracle documentation for help with these other tasks.

The administrative tasks needed to support using OC require both the knowledge of an experienced Oracle Database Administrator (DBA) and some knowledge of application system administration. Although Oracle Configurator Projects vary in administrative requirements, this manual provides explicit explanations and directions for common tasks wherever possible.

Oracle Configurator administrative tasks should be handled by the Oracle Applications System Administrator and the Oracle Database Administrator (DBA). An Oracle Applications System Administrator administers the user interface or applications side of Oracle Applications. An Oracle Database Administrator (DBA) administers the data that users enter, update, and delete while using Oracle Applications.

Oracle Configurator provides concurrent programs and SQL*Plus scripts for many of the administrative tasks. The concurrent programs are available in Oracle Applications System Administrator and Bills of Material. The scripts are available on the Oracle Configurator Developer CD in the `DBAdmin` folder, and are explained in this book.

Oracle Configurator integrates with other Oracle Applications by transferring data to and from the Oracle Configurator schema (CZ). For example, you can transfer Bills Of Material (BOM) data to the CZ schema, create a configuration model to run in an Oracle Configurator window using the Oracle Configurator Developer, then access that configuration model as an item in Order Management, configure the item in the Oracle Configurator window within Order Management, and then submit the configured item as part of the order. In Order Management, the configuration model is based on an imported BOM. It is also possible to configure a BOM in the Oracle Configurator window of Order Management, Order Capture, and Telesales without every having imported it into the Configurator Developer to create a configuration model for it.

## 1.4 Terms

Oracle and Oracle Configurator use specific terminology to refer to the concepts and components of databases and applications. Oracle8*i* Enterprise Edition, the Oracle RDBMS required by Oracle Applications, is installed, set up, and started as an Oracle **Server**.

An Oracle Server consists of one or more Oracle **database instances**. A database instance consists of memory and processes that manage a single, self-contained collection of data. An instance provides controlled access to the user(s) of the database.

An **Oracle database** is a collection of data treated as a unit. It has logical and physical structures. The logical structures are the schema objects (tables, views, stored procedures, database links). The physical structures include the datafiles and log files. Datafiles contain all of a database's data, including physical data, that are used to build up the logical structures.

A **schema** in a database is a collection of database objects. There can be multiple schemas in a single database. A schema (all the objects that make up a schema) takes the name of its **owner** (also called a database owner or DBOwner). In Oracle Applications, there is only one product-specific schema per product, meaning only one CZ schema.

In some cases, a **data transfer** is required to explicitly move data from one database or schema to another schema. Concurrent programs may be available to transfer data from one schema to another within a given database or among various releases of the Oracle Applications database. SQL*Plus scripts may be available for data **transfer** between separate databases.

In the Oracle Configurator product generally, and this book in particular, users are distinguished from end users. **Users** are the implementers using Oracle Configurator Developer to create a configuration model that runs in an Oracle Configurator window. **End users** are the users of those Oracle Configurator windows.

## 1.5 Requirements

All requirements and prerequisites for installing and using Oracle Configurator Developer are presented in the *Oracle Configurator and SellingPoint ReadMe* on the Oracle Configurator Developer CD or the *Oracle Configurator and SellingPoint Release Notes* (Part No. A73283-01). Additional requirements and prerequisites for installing and using Oracle Configurator Toolkit are presented in *Oracle Configurator Toolkit Developer's Guide* (Part No. A77067-05).

Oracle Configurator users must be defined users in (registered with) Oracle8*i* Enterprise Edition. Oracle Configurator end users must be defined users in Oracle Applications.

## 1.6 Oracle Configurator Installations

The Oracle Configurator schema and Oracle Configurator window (end-user) are part of the Oracle Applications installation. In this release, Oracle Configurator Developer and Oracle SellingPoint are not part of the Oracle Applications installation but are installed from the Oracle Configurator Developer CD. The Oracle Configurator window additionally requires an application server installation. See Chapter 7, "Installing the Servlet" and the *Oracle Configurator Toolkit Developer's Guide* (Part No. A77067-05) for these installation instructions.

The administrator must be informed of the basic installation(s) of the Oracle Configurator needed for the site. Basic installations include:

- Development
- Test
- Deployment
- Maintenance

For each installation, there is likely to be a separate organization. In any installation, you will be running the Oracle Configurator in one of the following scenarios:

- Oracle Applications Release 11*i* with the Java applet Oracle Configurator window in Order Management, Telesales, or Order Capture to configure items that are based on a configuration model or configurable BOM.
- Oracle Applications Release 11*i* with the DHTML Oracle Configurator window in iStore, Sales Online, or Order Capture to configure items that are based on a configuration model.
- A custom web application with a DHTML Oracle Configurator window for configuring items based on a configuration model. This requires using Oracle Configurator Toolkit.
- Any Oracle runtime configurator using imported Oracle Applications 11.0 or 10.7 data, or non-Oracle or legacy data.
- Oracle SellingPoint application synchronizing with a Release 11*i* Oracle Applications database.

- Oracle SellingPoint application as a mobile deployment exporting quote data to Oracle Order Entry or Oracle Receivables (Release 10.7 or 11.0).
- A test instance of an Oracle Configurator window or Oracle SellingPoint application launched from Oracle Configurator Developer running on a client Windows machine networked by LAN to the Release 11*i* Oracle Applications database on a server machine.

Settings in the CZ_DB_SETTINGS table in the Oracle Configurator schema may vary depending on whether the installation is for development, test, deployment, or maintenance. There is one CZ_DB_SETTINGS table for each Oracle Configurator schema that is in effect for all users of that schema.

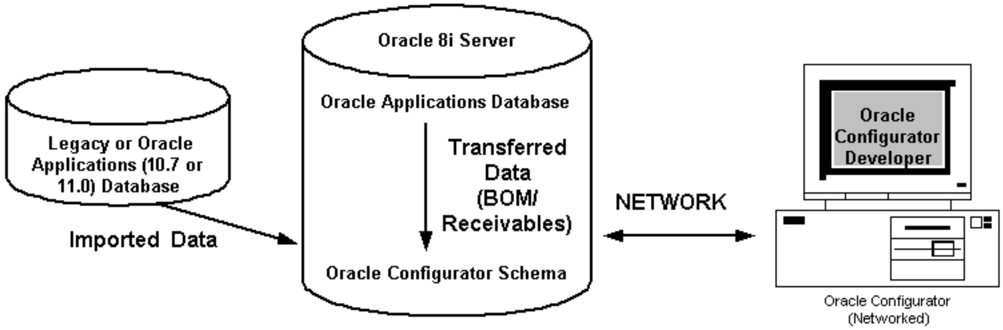
For details about what parameters in CZ_DB_SETTINGS apply in each of the scenarios, above, see Section 3.2, "Oracle Configurator Schema Settings" on page 3-4.

### 1.6.1 Development

A development environment is one in which you implement your Oracle Configurator project by constructing an end-user application referred to in this manual as an Oracle runtime configurator. Development is most commonly a networked installation with Oracle Configurator Developer on the client machine and the Oracle Configurator schema on a server. Running an Oracle Configurator window also requires installing an application server. See *Oracle Configurator and SellingPoint Release Notes* for additional warnings, requirements, and helpful hints before you begin development.

Data can be transferred from other schemas in the Oracle Applications database to the Oracle Configurator schema or it could be directly imported from Oracle Applications Release 10.7 or 11.0, or generically imported from any legacy database.

*Figure 1–1    Development Installation*


Oracle 8i Server
Oracle Applications Database
Transferred Data (BOM/ Receivables)
Oracle Configurator Schema
Legacy or Oracle Applications (10.7 or 11.0) Database
Imported Data
NETWORK
Oracle Configurator Developer
Oracle Configurator (Networked)

The source database containing legacy or non-Oracle configuration data, may or may not be on the same machine as the Oracle Configurator schema. If not, the integration tables in the Oracle Configurator schema must be set up with links to the source database.

After successfully importing any legacy data needed for modeling configurations at the start of your development cycle, Oracle recommends that you complete and test your configuration model before transferring new data. Configuration modeling data includes item master, item type, structure, and property data.

### 1.6.2 Test

A test environment is one in which you test your Oracle Configurator window in preparation for initial deployment, upgrades, and new releases, decoupled from continuing development.

Test the application functionality in a networked environment similar to the development environment making sure periodic data transfer and modifications work as required. For example, changed prices should propagate through to Order Management.

### 1.6.3 Deployment

A deployed application is one in which end users of the Oracle runtime configurator use the software in a production mode. To prepare for deployment, you should publish a production-ready version of the Oracle Configurator schema.

For more efficient use of machine resources, purge records flagged deleted before publishing the production-ready version.

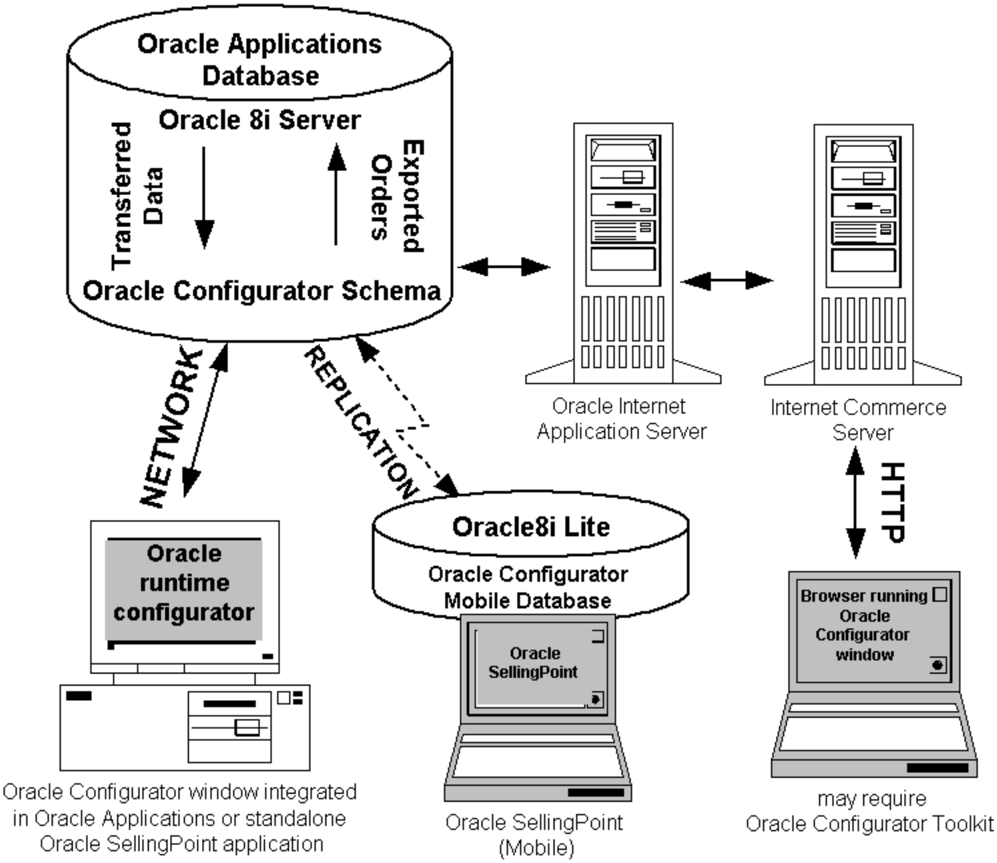
The possible deployment scenarios are:

- client/server (networked)
- web
- mobile

**Note:** Mobile deployment is supported for the Oracle SellingPoint application only and requires consulting services. For more information regarding mobile deployment, Chapter 8, "Oracle Configurator Deployment".

For information about how to publish or refresh a production Oracle Configurator schema, see Section 1.6.4, "Maintenance" on page 1-11.

***Figure 1–2 Deployment Installations***


Oracle Applications Database
Oracle 8i Server
Transferred Data
Exported Orders
Oracle Configurator Schema
NETWORK
REPLICATION
Oracle Internet Application Server
Internet Commerce Server
HTTP
Oracle runtime configurator
Oracle Configurator window integrated in Oracle Applications or standalone Oracle SellingPoint application
Oracle8i Lite
Oracle Configurator Mobile Database
Oracle SellingPoint
Oracle SellingPoint (Mobile)
Browser running Oracle Configurator window
may require Oracle Configurator Toolkit

In Figure 1–2, the networked Oracle Configurator window could be integrated with Oracle Order Management, TeleSales, or Sales Online. The Oracle Configurator Toolkit could be embedded in a custom web store application or Oracle iStore.

The Oracle Configurator window user interface runs in a browser. For the Toolkit, the application itself, and the Oracle Configurator schema, are running on the Oracle Configurator window server machine with the internet commerce server brokering the processes and http connection.

### 1.6.4 Maintenance

A maintenance environment is similar to a development environment since it is used to create upgrades.

At the time you deploy your Oracle runtime configurator and publish a production Oracle Configurator schema, you also create a maintenance Oracle Configurator schema. In the course of a deployed release of your Oracle runtime configurator, you may conduct periodic transfers from Oracle Applications or imports from your legacy data and redeploy refreshed Oracle runtime configurators.

Another aspect of maintenance involves fixing and improving the configuration model and publishing these in periodic upgrades. It is important to synchronize these changes in the maintenance Oracle Configurator schema with the Oracle Configurator schema under development for the next release of your Oracle runtime configurator. For information about refreshing from one version of your Oracle Configurator schema to another, see Section 3.3.1, "Refresh or Update the Production Schema" on page 3-17.

When you upgrade the release version of Oracle Configurator that your Oracle runtime configurator runs against, you start by upgrading your Oracle Configurator schema. For information on upgrading from one release to another of Oracle Configurator, see the *Oracle Configurator and SellingPoint ReadMe* on the Oracle Configurator Developer CD.

Once you have upgraded your Oracle Configurator schema for a new release, you must re-execute the Generate Active Model command in Oracle Configurator Developer.

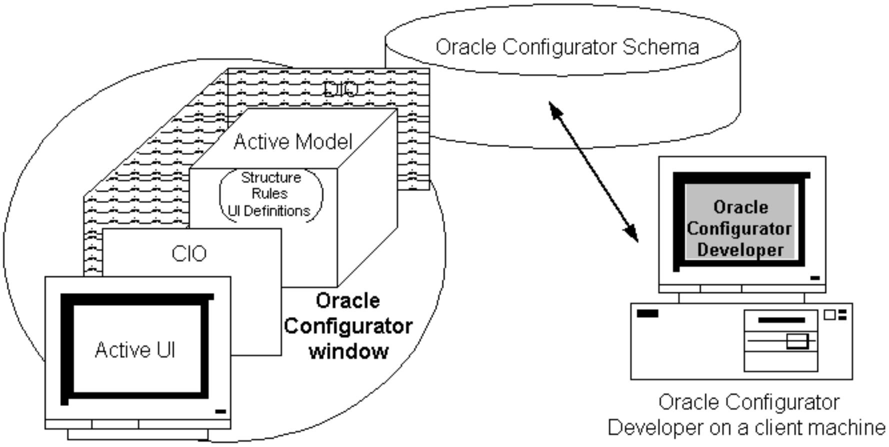
**Warning:** **Do not run Oracle Configurator Developer and the Generate Active Model command against a deployed production Oracle Configurator schema.**

## 1.7 OC Architecture Overview

No matter what the installation or environment, the Oracle Configurator architecture is essentially the same. An Oracle runtime configurator consists of a model that is developed in Oracle Configurator Developer. The model or configuration model is also sometimes called the **Active Model** and manages the model structure, configuration rules, and UI definitions of the Oracle runtime configurator.

The model-driven UI definitions in the Oracle runtime configurator are called the **Active UI**. The Active UI and Active Model interact with the data in the Oracle Configurator schema to present users with the data they need to develop and run the Oracle Configurator window.

*Figure 1–3 Overview of OC Architecture*


Oracle Configurator Schema
DIO
Active Model
Structure
Rules
UI Definitions
CIO
Active UI
Oracle Configurator window
Oracle Configurator Developer
Oracle Configurator Developer on a client machine

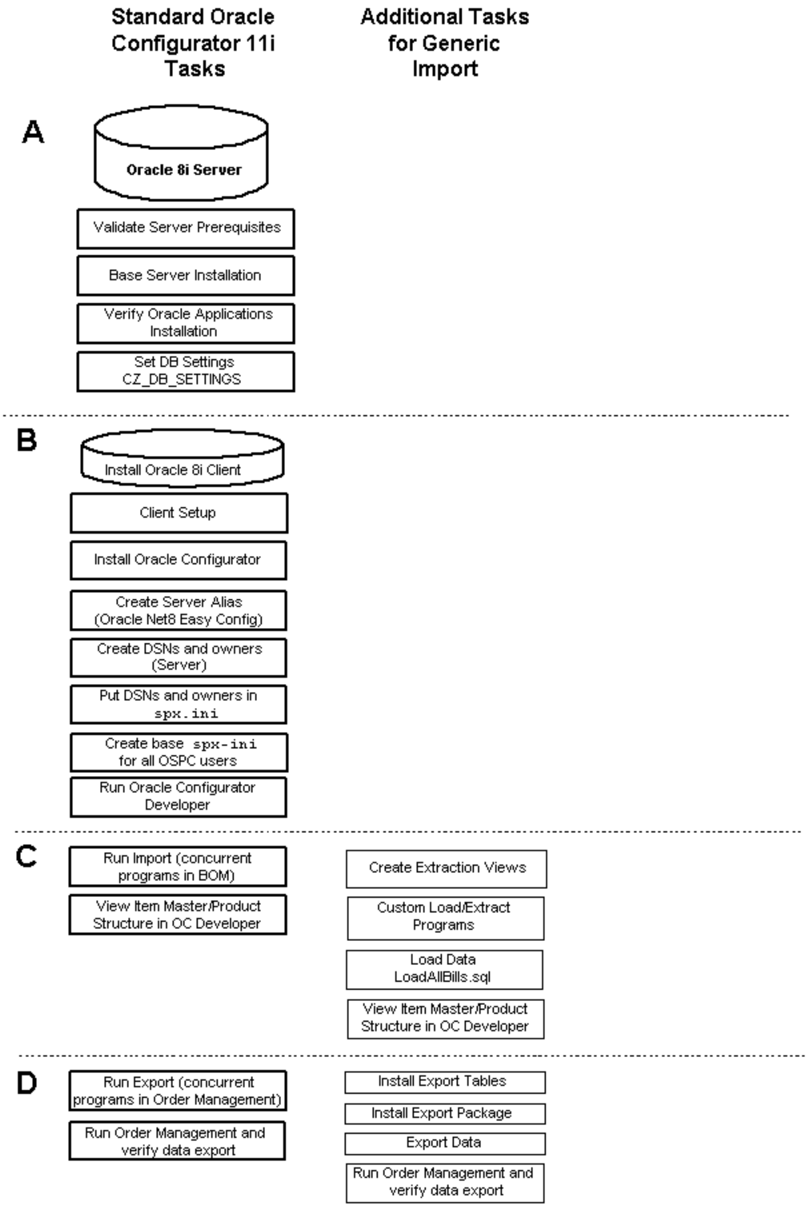
The CIO is the Configuration Interface Object, an API used for communication between the Active UI and other applications such as the Active Model.

The DIO is the Data Integration Object, responsible for communications between the Active UI and the Oracle Configurator schema, and between the Active Model and the Oracle Configurator schema.

## 1.8 Administration Tasks

In order to support any installation of Oracle Configurator, you must perform administrative tasks in the order presented below. The tasks in the right column are additional tasks required for supporting generic import of data to the Oracle Configurator schema. See Section 4.3, "Generic Import", on page 4-23 for more information about generic import.

***Figure 1–4 Administrative Task Flow***


Standard Oracle Configurator 11i Tasks
Additional Tasks for Generic Import
A
Oracle 8i Server
Validate Server Prerequisites
Base Server Installation
Verify Oracle Applications Installation
Set DB Settings CZ_DB_SETTINGS
B
Install Oracle 8i Client
Client Setup
Install Oracle Configurator
Create Server Alias (Oracle Net8 Easy Config)
Create DSNs and owners (Server)
Put DSNs and owners in spx.ini
Create base spx-ini for all OSPC users
Run Oracle Configurator Developer
C
Run Import (concurrent programs in BOM)
View Item Master/Product Structure in OC Developer
Create Extraction Views
Custom Load/Extract Programs
Load Data LoadAllBills.sql
View Item Master/Product Structure in OC Developer
D
Run Export (concurrent programs in Order Management)
Run Order Management and verify data export
Install Export Tables
Install Export Package
Export Data
Run Order Management and verify data export

## 1.8.1 Commonly Performed Tasks

This book presents specific instructions for performing many of the administrative tasks required to set up and support development and deployment of an Oracle runtime configurator. Instructions for some frequently performed tasks are included in the following sections:

- Connect to a Database Instance
- Verify Oracle Configurator Schema Version
- Run SQL*Plus in the <OC-scripts> Directory
- View Oracle Configurator Concurrent Program Requests

### 1.8.1.1 Connect to a Database Instance

Certain administrative tasks must be performed while connected to a database instance such as <appssid>.

To connect to a database, you must specify a user or schema and the instance in which it is defined. For example

1. Connect to your Oracle Configurator schema by connecting to the instance <ocsid> as a the user or schema you need to access.

   Example:

   ```
   SQL> connect <oc>/<ocpass>@<appssid>
   ```

   where <oc> is the owner (DBOwner) of the Oracle Configurator schema, and <appssid> is the name for the Oracle8*i* Enterprise Edition instance on which the Oracle Configurator schema is installed.

2. Or connect to the database instance as a user with DBA privileges:

   Example:

   ```
   SQL> connect <dba>/<dbapass>@<appssid>
   ```

3. Connect to the instance <appssid> as the integration user.

   Example:

   ```
   SQL> conn <imp> <imppass>@<appssid>
   ```

### 1.8.1.2 Verify Oracle Configurator Schema Version

The version of the Oracle Configurator schema is available in the CZ_DB_SETTINGS table.

1. To verify that the Oracle Configurator schema is correct, select the version settings from the CZ_DB_SETTINGS table.

   For example:

   ```
   SQL> select setting_id, value, desc_text from cz_db_settings where setting_
   id like '%_VERSION"
   ```

   The result for Release 11*i* should be MAJOR_VERSION = 14, MINOR_VERSION = c.

### 1.8.1.3 Run SQL*Plus in the <OC-scripts> Directory

OC provides scripts for importing legacy, non-Oracle, or Oracle Applications 10.7 or 11.0 data. These scripts are available on the Oracle Configurator Developer CD in the `DBAdmin` folder. Any OC script must be run from SQL*Plus running in the directory where all the scripts from the `DBAdmin` folder are located.

**Note:** Do not run any OC SQL*Plus scripts from SQL Worksheet.

Some scripts must be run while connected to a specific database instance (see Section 1.8.1.1, "Connect to a Database Instance").

To run SQL*Plus in the <OC-scripts> directory:

2. Start SQL*Plus
3. Go to File --> Open
4. Navigate to your <OC-scripts> directory
5. Click Cancel

You are now running SQL*Plus in the <OC-scripts> directory.

### 1.8.1.4 View Oracle Configurator Concurrent Program Requests

Since all reports, programs, and request sets are run as concurrent requests in Oracle Applications, you use the Requests window to view the status and output of your requests.

You can use the Requests window to view a list of all submitted concurrent requests, check whether your request has run, change aspects of a request's processing options, diagnose errors, or find the position of your request in the queues of available concurrent managers.

You can navigate to the Requests window using the Navigate window. Different Oracle Applications products use different menu paths in the Navigate window to access the Requests windows.To access this window:

---

**Note:** If your system administrator sets the profile option Concurrent: Report Access Level to "User", the Requests window displays the concurrent requests for the current user.

If your system administrator sets the profile option to "Responsibility", the Requests window displays the concurrent requests for the current responsibility in addition to requests for the current user.

---

# 2

# Client/Server Administration

Servers are hardware devices running processes shared by client machines, or servers are those processes shared by client machines. Server administration for Oracle Configurator involves both.

An Oracle runtime configurator and Oracle Configurator development tools, such as Oracle Configurator Developer, typically run on client machines connected to the Oracle Applications database server. An Oracle Configurator window embedded in other Oracle Applications Forms, such as Order Management, runs on the application server machine where the calling application runs. The Oracle Configurator Toolkit window runs embedded in other Oracle Applications such as iStore or in a custom webstore on the internet server machine serving the host web browser.

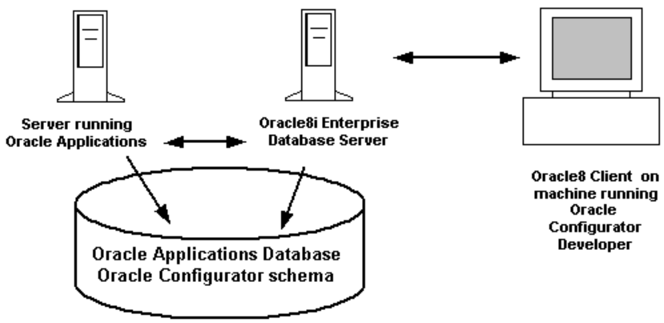
## 2.1 Overview of Server Administration

There are a number of servers involved in setting up and supporting Oracle Configurator:

- Database server where the Oracle Applications database, including the Oracle Configurator schema, is installed
- Application Server running Oracle Applications Forms
- Oracle Applications Internet Server for web deployments

In addition, Oracle Configurator Toolkit requires an application server to broker the processes and http connection (see Figure 1–2, "Deployment Installations", on page 1-10).

*Figure 2–1 Server Configuration for OC Server*



Oracle Applications Forms and the Oracle Applications database containing the Oracle Configurator schema can be located on separate machines.

## 2.2 Oracle Configurator Server Machine

In general, the Oracle Applications database with the Oracle Configurator schema is installed on a machine that acts as the server for networked clients running OC. This would be the case for any installment of OC. The server machine where the Oracle Configurator schema is installed must also have Oracle8*i* Enterprise Edition installed and the client machine(s) must have Oracle8 Client installed.

The server database setup includes the following tasks:

- Install the Oracle Applications database (<appssid> and <apps>)
- Set Oracle Configurator DB Settings

Once a server database is set up, additional tasks may include making copies of populated schemas and upgrading the Oracle Configurator schema to a new version of the schema in a new release of OC. For information about installing the Oracle Applications database, see your Oracle Applications documentation. For information about the Oracle Configurator schema, see Chapter 3, "The Oracle Configurator Schema". For specific information about Oracle Configurator schema DB settings, see Section 3.2, "Oracle Configurator Schema Settings", on page 3-4.

### 2.2.1 Validate Server Prerequisites

You must install Oracle8*i* Enterprise Edition first. If you intend to support a mobile deployment of the Oracle SellingPoint application, install Oracle8*i* Enterprise

Edition with the advanced replication option enabled before installing Oracle Applications. Do not try to turn the replication option on after having installed Oracle Applications unless you have advanced expertise in replication.

## 2.2.2 Create Users and Responsibilities

Any OC user must also be defined as a database user in the Oracle8*i* Enterprise Edition server database running the Oracle Configurator schema.

### 2.2.2.1 Accessing Oracle Runtime Configurators

End users for the Oracle Configurator window (DHTML or Java Applet) and their responsibilities are established through Oracle Applications administration and reside in the Oracle Applications database.

The Oracle Configurator schema DBOwner (<oc>) can log in to the Oracle Configurator Developer. A non-DBOwner user of Oracle Configurator Developer need only be an Oracle8*i* Enterprise Edition account with <SPX_USER> role privileges to log into Configurator Developer (<ocdev>). If <oc> is in the CZ_END_USERS table, then the Oracle Configurator schema DBOwner can log in to the Oracle SellingPoint application. To use the Oracle SellingPoint application, a non-DBOwner user (Oracle8*i* Enterprise Edition account) would also have to be inserted in the CZ_END_USERS table.

Running GRANT_TO_ROLE.sql grants access privileges to the SPX_USER role (or to the customized role stored in CZ_DB_SETTINGS). GRANT_TO_ROLE.sql goes through all of the Oracle Configurator schema objects (tables, views, sequences, packages, etc.) and grants access on each to <SPX_USER>. When complete, any database user that has been granted <SPX_USER> has sufficient database permissions to run Oracle Configurator Developer and the Oracle SellingPoint application.

### 2.2.2.2 Access and Role Requirements for Oracle SellingPoint Application

The Oracle Configurator schema (<oc>_schema) contains the following user tables:

CZ_END_USERS
CZ_USER_GROUPS
CZ_END_USER_GROUPS

This cluster of tables maps the association of users to user groups. The Oracle Configurator schema DBOwner (<oc>) must be inserted in the CZ_END_USERS table and granted the role <SPX_USER>. The <SPX_USER> role is stored in the CZ_DB_SETTINGS table and it is needed for using Oracle SellingPoint application.

One user group is implemented: END USER. The role granted to the END USER user group is <SPX_USER>.

The access privileges defined in <SPX_USER> are the following on all Oracle Configurator schema tables:

- Select
- Insert
- Update
- Delete

Any end user assigned the <SPX_USER> role and listed in the CZ_END_USERS table can log in to the Oracle SellingPoint application. Additionally, any user that is database enabled and granted <SPX_USER> privileges can create customers, i.e., Opportunities, Configurations, etc. End users cannot, however, view each other’s Opportunities in the Oracle SellingPoint application. Users can only view their own Opportunities.

#### 2.2.2.3 User Administration for Oracle SellingPoint Application

The <SPX_USER> role alone is not sufficient for adding or managing end users. The DBOwner of the Oracle Configurator schema <oc> must be assigned the RESOURCE role to create users. To allow addition of users in the Oracle SellingPoint application, the DBOwner of the Oracle Configurator schema must be granted DBA or preferably a more granular role such as 'create user'.

In terms of executing administrative tasks and using Oracle Configurator Developer, any user granted <SPX_USER> privileges can run the OC SQL*Plus scripts and packages provided in the Oracle Configurator Developer CD, use the SEQUENCEs, etc. Users granted <SPX_USER> privileges cannot restructure (DDL) tables and other objects in the Oracle Configurator schema.

The Oracle Configurator schema DBOwner (<oc>) also has privileges that enable them to set other end user permissions, such as Administrative permissions. With Administrative permissions, you can use the Tool > Administration > Users... menu of the Oracle SellingPoint application to set end users’ permissions to an Oracle SellingPoint application Project. See Section 8.2.1, "Oracle SellingPoint Application User Access" on page 8-3 for more information.

### 2.2.2.4 Access to Oracle Applications

Any OC user who will be submitting orders or new customer data to Oracle Applications must also be defined as an Oracle Applications user. It is usually easiest to define Oracle Applications sales representatives and import them. Additionally, end users are associated with an ORGANIZATION_ID for accessing Order Management. See Section 8.2.1, "Oracle SellingPoint Application User Access" on page 8-3 for more information.

During a data transfer from Oracle Applications, the CZ_END_USERS table in the Oracle Configurator schema is populated with all defined Oracle Applications users. All Oracle Applications users listed in the CZ_END_USERS table have <SPX_USER> role granted to them by being in the user group END USER.

Depending on the value of AUTOCREATE_IMPORTED_USERS in the CZ_DB_SETTINGS table, those end users are either database enabled or not. If not, run EndUsers.sql to selectively enable end users in CZ_END_USERS as database users. Setting AUTOCREATE_IMPORTED_USERS to "YES" in the CZ_DB_SETTINGS table designates all imported users as database users. See Section 3.2.4, "CZ_DB_Settings for ORAAPPS_INTEGRATE" on page 3-11 for more information about AUTOCREATE_IMPORTED_USERS.

Using the CZ_XFR_TABLE, you can enable or disable the CZ_END_USERS table to control whether or not users are imported from the Oracle Applications Database.

GRANT_SELECT_FOR.sql grants access on Oracle Applications tables to the Oracle Configurator schema DBOwner. See also Appendix C.2.6, "Using GRANT_SELECT_FOR.sql" on page C-27. When integrating Oracle SellingPoint application with Oracle Applications on a local instance, running GRANT_SELECT_FOR.sql as <apps> grants SELECT privileges on some Oracle Applications tables to the DBOwner of the Oracle Configurator schema involved in export to Order Management.

## 2.3 Overview of Client Administration

Client administration involves setting up a client machine with Oracle8 Client and installing Oracle Configurator. Installing Oracle Configurator for an implementation, test, or maintenance environment consists of installing Oracle Configurator Developer (which includes the Oracle runtime configurator).

After installing Oracle Configurator, you must establish data connectivity with the Oracle Configurator schema on the server machine and set parameters accordingly in the `spx.ini` file on the client machine.

## 2.4 Oracle Configurator Developer Client Requirements

Oracle Configurator Developer runs on a client machine for:

- Implementers developing and testing an Oracle runtime configurator.
- People who are maintaining, supporting, and upgrading a deployed Oracle runtime configurator.

The usual setup is Oracle Configurator Developer running on a client machine networked to a server where the the Oracle Applications database with the Oracle Configurator schema is installed. The requirements for a networked setup are:

- The Oracle Configurator schema is running on the server.
- Oracle8 Client is installed on the client machine that is running Oracle Configurator Developer. See Section 2.7, "Oracle8 Client Installation", on page 2-8.
- The client machine is configured to connect to the Oracle Configurator schema on the server machine (Oracle Net8 Easy Config). See Section 2.8.2, "Enable the Client for Database Connectivity", on page 2-10.
- A datasource name (DSN) for the Oracle Configurator schema on the server machine is defined in ODBC Administrator on the client machine. See Section 2.8.3, "Create DSNs and DBOwners", on page 2-11.
- The user logged into the client machine is a user defined in Oracle8*i* Enterprise Edition running the Oracle Configurator schema on the server.
- The user logged into the client machine is a user defined in the Oracle Applications database.
- The `spx.ini` file is edited to include the correct DSN and DBOwner for the Oracle Configurator schema on the server. See Section 2.8.5, "Parameters in spx.ini for Development and Test", on page 2-16.

## 2.5 Oracle SellingPoint Application Client Requirements

The Oracle SellingPoint application runs on a client machine for:

- Implementers developing and testing an Oracle SellingPoint application.
- People who are maintaining, supporting, and upgrading a deployed mobile Oracle SellingPoint application.

Setup for running the Oracle SellingPoint application on a networked client during development, testing, maintenance, and support is essentially the same as that for

Oracle Configurator Developer on a machine networked to a server. The requirements for a networked setup of the Oracle SellingPoint application are:

- The Oracle Configurator schema is running on the server.
- Oracle8 Client is installed on the client machine running the Oracle SellingPoint application. This is also necessary for running the Oracle Configurator Developer.
- The client machine is configured to connect to the Oracle Configurator schema on the server machine (Oracle Net8 Easy Config). See Section 2.8.2, "Enable the Client for Database Connectivity", on page 2-10.
- A datasource name (DSN) for the Oracle Configurator schema on the server machine is defined in ODBC Administrator on the client machine. See Section 2.8.3, "Create DSNs and DBOwners", on page 2-11.
- The user logged into the client machine is a user defined in the Oracle Applications database.
- The end user logged into the client machine is a user defined in (registered) Oracle8*i* Enterprise Edition running the Oracle Configurator schema on the server and a user defined in the Oracle Applications database.
- The end user logged into the client machine is an end user included in the CZ_END_USERS table of the Oracle Configurator schema.
- In order to use the Oracle SellingPoint application for testing or a mobile deployment, the end user must be assigned to the end_user_group, the end_user_group must be assigned to projects, and customers must be assigned to end users.
- The `spx.ini` file is edited to include the correct DSN and DBOwner for the Oracle Configurator schema on the server. See Section 2.8.5, "Parameters in spx.ini for Development and Test", on page 2-16.

## 2.6 Oracle Configurator Window Client Requirements

When running a test instance of the Oracle Configurator window (JAVA applet or DHTML) from within Oracle Configurator Developer, the requirements for running Configurator Developer must be satisfied. In addition, the URL for the DHTML and Java applet Servlet UI must be specified in the `spx.ini` file (see Section 2.8.5.10, "[Test]" on page 2-21).

All client requirements for running an Oracle Configurator window within Oracle Applications are satisfied by your Oracle Applications setup.

## 2.7 Oracle8 Client Installation

Oracle8 Client allows the OC user to access an Oracle8*i* Enterprise Edition database from a client machine not running Oracle8*i* Enterprise Edition. Networked client machines must have Oracle8 Client installed if they access the Oracle8*i* Enterprise Edition Server database.

Oracle8 Client is used to access the Oracle Configurator schema from a client machine.

## 2.8 Set Up Oracle Configurator

Once your site has Oracle8*i* Enterprise Edition Server and Oracle8 Client installed and you have installed Oracle Applications and the Oracle Configurator schema, you can install and set up Oracle Configurator for your users.

The tasks required to set up Oracle Configurator for your users are:

- Install Oracle Configurator
- Enable the Client for Database Connectivity
- Create DSNs and DBOwners
- Parameters in spx.ini for Development and Test
- Run Oracle Configurator Developer and the Test Configurator

Figure 2–2, below, illustrates an overview of the architecture of client/server data communication you set up with these tasks.

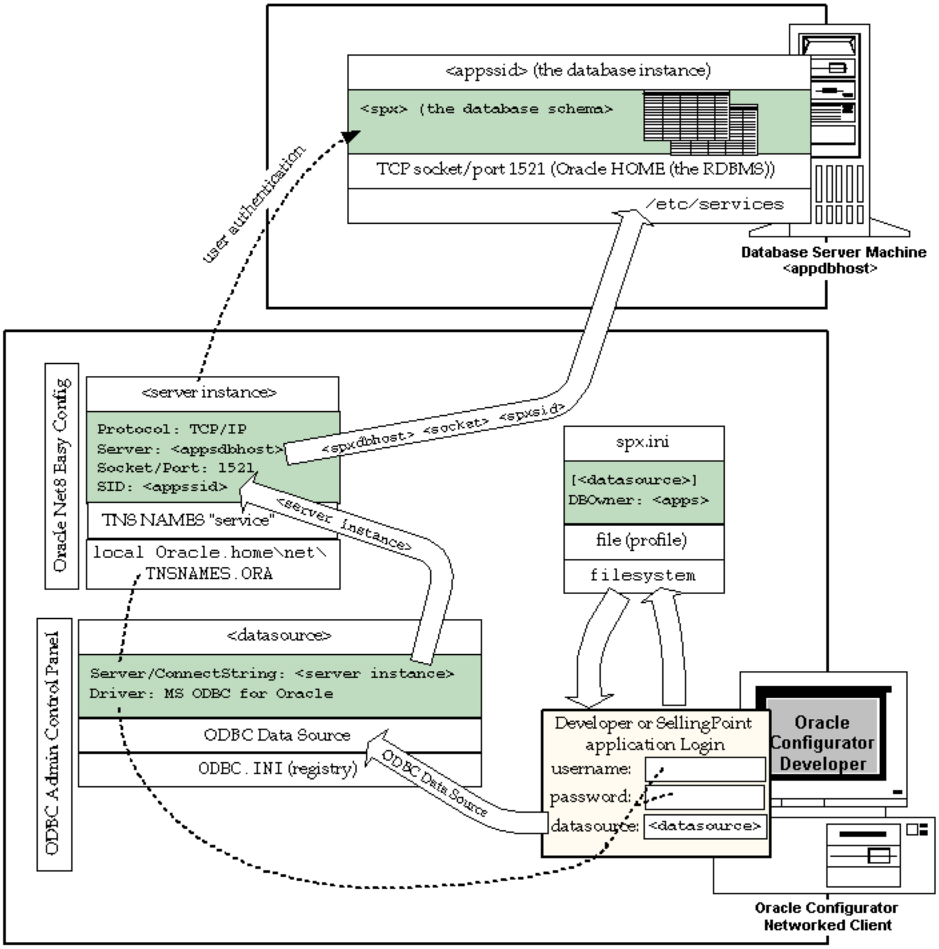


*Figure 2–2 Client/Server Data Communication Architecture*

## 2.8.1 Install Oracle Configurator

See the *Oracle Configurator and SellingPoint ReadMe* on the Oracle Configurator Developer CD for platform requirements, prerequisites, and installation instructions.

When establishing an instance of the Oracle Configurator schema, the Oracle Applications System Administrator customizes the DBOwner, access roles, and privileges.

You must insert a default <SPX_USER> record in the CZ_END_USERS table of the Oracle Configurator schema. Generic import populates the Oracle Configurator schema CZ_END_USERS table with the Oracle Applications user names and passwords, provided the DB_SETTING AUTOCREATE_IMPORTED_USERS is non-Null.

You then log into the Oracle SellingPoint application as the DBOwner to:

- create more users (optional)
- assign customers to users (required)
- assign projects to users (required)

These tasks are done in the Tools -> Administration menu. See Section 8.2.1, "Oracle SellingPoint Application User Access" on page 8-3 for details.

## 2.8.2 Enable the Client for Database Connectivity

Each machine running Oracle Configurator Developer must be configured to connect to an Oracle8*i* Enterprise Edition server instance through Oracle Net8 Easy Config. The service name is used to create a TNS alias.

When running Oracle Configurator Developer on a client machine, the client machine needs data connectivity to the Oracle Configurator schema on the server machine. The client machine must be running Oracle8 Client. To establish data connectivity on the client machine, you need to know the following parameters:

- the name of the server database instance or system identifier (SID) where the Oracle Configurator schema is located (<ocsid>)
- the name of the physical server machine where <ocsid> is located (<ocdbhost>)
- the port of the physical server machine through which the client connects to the Oracle Configurator schema (<port>)
- the username / password of the DBOwner of the Oracle Configurator schema schema to which the client is connecting (<oc> / <ocpass> or <ocdev> / <ocdevpass>).

You must set these parameters using Oracle Net8 Easy Config to establish data connectivity. To establish data connectivity, follow these instructions:

1. In Windows 95/98 or Windows NT 4.0, select Start > Programs > Oracle for Windows > Oracle Net8 Easy Config. (If you do not have this option, you have an outdated version of Oracle, or no client software). Install Oracle Net8 Easy Config using the Oracle Installer.
2. Select Add New Service and enter the new service name**.** This is the name of the server database instance containing the Oracle Configurator schema you will be connecting to (<ocsid>). You ***must*** use this same name as the parameter in the Server field of your ODBC Configuration. Click Next.
3. Select the networking protocol TCP/IP (Internet Protocol). Click Next.
4. Enter the Host Name (<ocdbhost>) and the Port Number (<port>). The host name is the name of the physical server machine where the Oracle Configurator schema is located. Make a note of the Port Number (default is 1521) in case it is needed for future reference. Click Next.
5. Enter the Database SID (<ocsid>). This is the name of the server database instance containing the Oracle Configurator schema you will be connecting to, the same name as the New Service you just added. Click Next.
6. Test the connection by clicking on the Test Service button.
7. Enter the Username and Password for the owner of the Oracle Configurator schema you've been setting up data connectivity to. Click Test.
8. After clicking on the Test button, the results display. Click Done when the result shows the test has completed successfully.
9. Click Finish to save your service configuration and exit Oracle Net8 Easy Config.

### 2.8.3 Create DSNs and DBOwners

Create ODBC datasource names (DSNs) for each Oracle8*i* Enterprise Edition server that you need for a development, test, or maintenance installation. If your production installation is run in client/server mode (not internet), the owner for that installation must also be defined in Oracle8*i* Enterprise Edition server.

Each machine running Oracle Configurator Developer runs against a version of the Oracle Configurator schema. The Data Source Name for that Database must be registered in the Microsoft ODBC Administrator control panel.

For an Oracle8 Client database, use the ODBC driver Microsoft ODBC Driver for Oracle.

Mobile production installations require a DSN and an owner in Oracle8*i* Lite for the production Oracle Configurator Mobile Database. For an Oracle8*i* Lite database, use the ODBC driver Oracle8*i* Lite ODBC Driver.

To set up the Data Source Name for your Oracle Configurator schema, follow these instructions:

1. In Windows 95/98 or Windows NT 4.0, select Start > Settings > Control Panel and open ODBC Data Sources (32bit). This opens the ODBC Data Source Administrator.
2. Select the System DSN tab.
3. Click Add... This opens the Create New Data Source dialog.
4. Select "Microsoft ODBC for Oracle" (version 2.573.xxxx.xx) for setting a server DSN. The driver you would select for an Oracle8*i* Lite DSN is Microsoft ODBC for Oracle8*i* Lite (client). If you have created a replica Oracle Configurator schema (Oracle Configurator Mobile Database) using rep_setup.sql, the DSN has already been set up for you. Click Finish.
5. The Microsoft ODBC for Oracle Setup dialog appears. Enter the name of the database you want to access (including the extension, i.e., .db or .odb) in the Data Source Name field.
6. Optionally, enter a description of the database driver that the data source connects to in the Description field.
7. Optionally, enter your database user ID in the User Name field.
8. Enter the Service Name for the Oracle Server engine in the Server field. The Service Name identifies the Oracle Database instance (<ocsid>) that you want to access.

> **Note:** This Service Name ***must*** be the same name you entered as the New Service Name when establishing data connectivity using Net8 Easy Config.

9. You can click Options to make more specifications about the Oracle ODBC setup (usually not necessary).

   Option: Translation

   Click the Select button to choose a loaded data translator. The default is No Translator.

Option: Performance

Include REMARKS in Catalog Functions specifies whether the driver returns Remarks columns for the SQL Columns result set. The ODBC Driver provides faster access when this value is not set.

Include SYNONYMS in SQL Columns specifies whether the driver returns column information.

Option: Customization

Enforce ODBC DayOfWeek Standard specifies whether the result set will conform to the ODBC specified day-of-week format (Sunday=1; Saturday=7).

10. Click OK to add the data source.

    This brings you back to the ODBC Database Administrator top level. Notice your DSN has been created and the ODBC Administrator updates the Windows registry information. The User Name and Service Name that you enter become the default data source connection values for this data source.

11. Click Add to add another data source or click OK to exit.

### 2.8.4 Example `spx.ini` File

The next two sections explain the parameters in the `spx.ini` file that determine connectivity and product behavior for development, test, and deployment. Parameters affecting Oracle Configurator Developer must be set for development and test. Parameters for the Oracle SellingPoint application must be set in the `spx.ini` file on the machine running the Oracle SellingPoint application if the Oracle SellingPoint application is:

- launched as the test environment by Oracle Configurator Developer (Test/Debug button)
- used to create a replica Oracle Configurator schema for mobile deployments
- deployed in a mobile environment synchronizing with the Oracle Configurator schema on a server.

No `spx.ini` file is required for deployments of an Oracle Configurator window.

***Example 2–1 Example spx.ini File***

```
[Merlin]
DBOwner=apps
```

```
ShowWarnings=false

[DSN]
Apps11iTest
DHTMLTest
JavaTest

[MDA]
DBOwner=apps
DSN = Apps11iTest
LCUST=
LOPP=
LACT=
HLOGO=
SLOGO=
LOGFILE=Oracle SellingPoint.log
RuntimeCache=1
EngineStartup=Early

[MDAPLUGINS]

[MDADSNS]
Apps11iTest
DHTMLTest
JavaTest

[Apps11iTest]
DBOwner=apps

[DHTMLTest]
DBOwner=apps

[JavaTest]
DBOwner=apps
gwyuid=APPLSYSPUB
gwypass=PUB

[vis11_replica]
DBOwner=SPX
master_name=vis11
master_schema_name=SPX
connection_type=0
connection_info=
ReplicationRefreshMode=OPTIMUM
timeout=600
```

```
olite_version_enum=3
last_replication=Friday, September 10, 1999 at 3:31:53 PM

[CURRENCY]
USD
FRF
DEM

[USD]
Name=U.S. Dollar
DecimalSeparator=.
NumericFormat=###\,###\,###\,###\,##0.00
NumericNegFormat=(###\,###\,###\,###\,##0.00)
CurrencyFormat=$###\,###\,###\,###\,##0.00
CurrencyNegFormat=$(###\,###\,###\,###\,##0.00)

[FRF]
Name=French Franc
DecimalSeparator=,
NumericFormat=###\ ###\ ###\ ###\ ##0.00
NumericNegFormat=-###\ ###\ ###\ ###\ ##0.00
CurrencyFormat=###\ ###\ ###\ ###\ ##0.00 FF
CurrencyNegFormat= -###\ ###\ ###\ ###\ ##0.00 FF

[DEM]
Name=German Mark
DecimalSeparator=,
NumericFormat=###\ ###\ ###\ ###\ ##0.00
NumericNegFormat=-###\ ###\ ###\ ###\ ##0.00
CurrencyFormat=###\ ###\ ###\ ###\ ##0.00 DM
CurrencyNegFormat= -###\ ###\ ###\ ###\ ##0.00 DM

[Design Chart]
DEF=M
SEC=X

[Test]
Launch=1
InitServletURL=
```

During installation of Oracle Configurator, the `spx.ini` file is copied to the `winnt` directory (for Windows NT machines) or the `Windows` directory (for Windows 95/98 machines). If the installation procedure encounters an existing spx.ini file, it renames that file spx_ini.bak, so that you do not lose edits you have made when you upgrade or reinstall Oracle Configurator.

The file contains [DSN] entries (used by Oracle Configurator Developer) and [MDADSNS] entries (used by the Oracle SellingPoint application). The file contains entries used by the Oracle SellingPoint application replica to synchronize with the master database.

## 2.8.5 Parameters in spx.ini for Development and Test

The spx.ini file sets the DBOwner and other parameters for running:

- Oracle Configurator Developer
- test instances of an Oracle Configurator window from within Oracle Configurator Developer (Test / Debug)
- test instance of the Oracle SellingPoint application from within Oracle Configurator Developer (Test / Debug)
- any instance of the Oracle SellingPoint application for creating a replica Oracle Configurator schema for mobile deployments

Throughout this section, references to the test configurator mean test instances of either an Oracle Configurator window or the Oracle SellingPoint application launched from within Oracle Configurator Developer (Test / Debug). See Section 2.8.6, "Parameters in spx.ini for Deployment" on page 2-21 for more information about parameters for testing a mobile deployment of Oracle SellingPoint application.

Oracle Configurator Developer and the test configurator require that the DSNs defined in the spx.ini file point to an installed Oracle Configurator schema. The DSNs set in the spx.ini file must also be registered in the ODBC Administrator for each machine running Oracle Configurator Developer and the test configurator.

You must edit the spx.ini file and update the [DSN] entries by adding the ODBC DSN(s) you created for your Oracle Configurator schema(s). The entries then appear in the Oracle Configurator Developer list of available data sources when you log in to Oracle Configurator Developer. You must create the Oracle Configurator schema DSN yourself, following the instructions in Section 2.8.3, "Create DSNs and DBOwners" on page 2-11; the spx.ini entries will not work until you create the DSN.

You must also edit the spx.ini file and add entries for the Oracle Configurator schema DSNs available to a test instance of the Oracle SellingPoint application [MDADSNS] and DBOwner. The DSNs listed in the default spx.ini file are those installed as part of the OC installation. Note, only dump files for server databases are on the Oracle Configurator Developer CD for demonstrations (See Section 2.8.5.2, "[DSN]").

Additional parameters may be defined specifically for manipulating the behavior of the test instance of the Oracle SellingPoint application (e.g., currency).

### 2.8.5.1 [Merlin]

The section [Merlin] lists parameters for Oracle Configurator Developer.

#### DBOwner in [Merlin]

The parameter DBOwner in the section [Merlin] specifies the default username of the owner of the Oracle Configurator schema that the spx.ini file accesses when starting up Oracle Configurator Developer. Users log into Oracle Configurator Developer with the schema name <oc> and the password <ocpass>.

This parameter must be updated to specify the actual DBOwner of the Oracle Configurator schema containing your Oracle Configurator Developer Project(s). Every database specified by a Data Source Name (see Section 2.8.5.2, "[DSN]", below) is associated with this DBOwner, unless another DBOwner is specified explicitly for both Oracle Configurator Developer and the test configurator with the setting:

```
[<DSN>]
DBOwner=<DBOwner>
```

See Section 2.8.5.6, "[<DSN>]", below.

#### ShowWarnings in [Merlin]

The parameter ShowWarnings in the section [Merlin] indicates whether or not generated logic warnings should be displayed in a warning dialog. By default this parameter is set to True, and warnings display in a warning dialog until you dismiss the dialog. If False, all warnings are entered into a log file and a popup dialog informs you of any generation logic warnings and remind you that they are in the log file. Rules that cause warnings do not have logic generated for them.

### 2.8.5.2 [DSN]

The section [DSN] lists the Data Source Names for the Oracle Configurator schemas available for use by Oracle Configurator Developer.

The DSN of the Oracle Configurator schema used with Oracle Configurator Developer must be listed here. Furthermore, a section must be added for the DSN of the server DBOwner by which users will access the server Oracle Configurator schema (see Section 2.8.5.6, "[<DSN>]").

In Example 2–1, "Example spx.ini File", there are several DSNs listed as examples. Apps11iTest is an Oracle Applications database available for use by Oracle Configurator Developer to test user interfaces launching the Oracle SellingPoint application. DHTMLTest is an Oracle Applications database available for use by Oracle Configurator Developer to test user interfaces launching a DHTML window.

JavaTest is an Oracle Applications database available for use by Oracle Configurator Developer to test user interfaces launching a Configurator Java Applet. **Gwyuid** in this DSN is the public Oracle gateway username and **gwypas** in this DSN is the public Oracle gateway password that grants access to the Oracle Applications signon form. Gwyuid and gwypas should be set the same as the default username and password in your Oracle Applications environment file. In order to use the Oracle Applications login functionality, the value for DBowner here should be the same as the FNDNAM parameter value in the Oracle Applications environment file.

### 2.8.5.3 [MDA]

The section `[MDA]` lists the parameters for the Oracle SellingPoint application.

#### DBOwner in [MDA]

The parameter `DBOwner` in the section `[MDA]` specifies the default username of the owner of the Oracle Configurator schema that this `spx.ini` file accesses when starting the Oracle SellingPoint application.

The DBOwner username FNDNAM is automatically inserted in the Oracle Configurator schema CZ_END_USERS table when the Oracle Configurator schema is installed with Oracle Applications. Users or end-users log into the test configurator with the name of a user listed in the CZ_END_USERS table. During development, test, and maintenance, this may be the schema name (DBOwner).

Every database specified by a Data Source Name (see "DSN in [MDA]") is associated with this DBOwner, unless another DBOwner is specified explicitly for both Oracle Configurator Developer and the Oracle SellingPoint application with the setting:

```
[<DSN>]
DBOwner=<DBOwner>
```

See Section 2.8.5.6, "[<DSN>]".

### DSN in [MDA]

The parameter DSN in the section [MDA] stores the last Oracle Configurator schema logged into by user of the test Oracle SellingPoint application and displays that Data Source Name by default in the Datasource field the next time the user or end-user with this spx.ini file logs into the Oracle SellingPoint application.

### LCUST

The parameter LCUST is populated with the last customer specified in the Oracle SellingPoint application. When the user or end-user logs back into the application again, the Resume button accesses this parameter and resets the application accordingly.

### LOPP

The parameter LOPP is populated with the last opportunity specified in the Oracle SellingPoint application. Using the Resume button accesses this parameter and resets the application accordingly.

### LACT

The parameter LACT is populated with the last activity specified in the Oracle SellingPoint application. A configuration is an activity, for instance. Using the Resume button accesses this parameter and resets the application accordingly.

### HLOGO

The parameter HLOGO specifies the path to the default logo displayed on the home screen in the Oracle SellingPoint application. No value assumes the default location: OSP/Shared/ActiveMedia/ in orawin95/ or orant/.

### SLOGO

The parameter SLOGO specifies the path to the default logo displayed on the summary screens in the Oracle SellingPoint application. No value assumes the default location: OSP/Shared/ActiveMedia/ in orawin95/ or orant/.

### LOGFILE

The parameter LOGFILE specifies the local filename where you want synchronization failures and other Oracle SellingPoint application errors to be logged. If you start the Oracle SellingPoint application from the START menu, by default, these errors are logged in the %ORACLE_HOME%\OSP\OSP\Oracle

SellingPoint.log file. If you start the Oracle SellingPoint application using the Test button in Oracle Configurator Developer, by default, these errors are logged in the %ORACLE_HOME%\OSP\Developer\Oracle SellingPoint.log file.

#### RuntimeCache

The parameter RuntimeCache specifies whether or not all UI information is loaded from the Oracle Configurator schema, and all screens are created at startup of the Oracle SellingPoint application. A value other than ‘1’ (one) causes screen information to load on demand as the end-user moves through the Oracle SellingPoint application UI. Any other value causes all screen information to load when first starting up the Oracle SellingPoint application.

#### EngineStartup

The parameter EngineStartup specifies whether or not the configuration engine starts up when the Oracle SellingPoint application starts up. A value of 'Early' causes the configuration engine to start up when the test configurator does. Any other value causes the configuration engine to start on demand when first creating a configuration.

### 2.8.5.4 [MDAPLUGINS]

The section [MDAPLUGINS] lists programs that can be launched from within the Oracle SellingPoint application. This Feature Is Not Currently Implemented.

### 2.8.5.5 [MDADSNS]

The section [MDADSNS] lists the DSNs for the Oracle Configurator schemas to which you want your Oracle SellingPoint application to have access.

### 2.8.5.6 [<DSN>]

If a DSN is not listed for every Oracle Configurator schema to which you need access from Oracle Configurator Developer or the Oracle SellingPoint application, it must be listed as a discrete section. A [<DSN>] section specifies the DBOwner by which the Oracle Configurator schema associated with the DSN will be accessed. Since the default install of OC includes sample Oracle8*i* Enterprise Edition server databases, these are each included as [<DSN>] sections (e.g., [Tutorial]) with the DBOwner apps.

```
[<DSN>]
DBOwner=<DBOwner>
```

---

**Note:** The DBOwner setting here overrides the DBOwner setting in the section `[Merlin]` and `[MDA]`. The DBOwner settings under `[Merlin]` and `[MDA]` are defaults.

---

#### 2.8.5.7 [<DSN>_replica]

See Section 2.8.6.5, "[<DSN>_replica]" on page 2-24 for information about this section of the `spx.ini` file.

#### 2.8.5.8 [CURRENCY]

The section `[CURRENCY]` lists the currency available in the Oracle SellingPoint application. The default `spx.ini` file includes formatting information for displaying prices in USD, FRF, and DEM in the Oracle SellingPoint application.

#### 2.8.5.9 [Design Chart]

The section `[Design Chart]` sets the alphanumeric symbols used to indicate defining and secondary optional features in the Design Chart configuration rule in Oracle Configurator Developer.

#### 2.8.5.10 [Test]

The section `[Test]` sets the type of environment to launch when using the Test/Debug button in Oracle Configurator Developer. Launch=1 (default) specifies the Oracle SellingPoint application as the test environment. Launch=2 specifies the Dynamic HTML in a browser. When Launch=2 is specified, the parameter InitServletURL must also be set to specify the URL of the servlet generating the Dynamic HTML in a browser.

Launch=3 specifies the Configurator Java Applet. When Launch=3 is specified, the parameter InitServletURL must also be set to specify the URL of the servlet generating the Configurator Java Applet. See Section 2.8.5.2, "[DSN]" for additional parameters that must be set for the test Configurator Java Applet.

These test options may also be selected in the Tools --> Options --> Test dialog in Oracle Configurator Developer.

### 2.8.6 Parameters in `spx.ini` for Deployment

The `spx.ini` file sets the DBOwner and other parameters for running:

- the Oracle SellingPoint application in a mobile deployment, including testing a mobile deployment.

The spx.ini file is not needed for running an Oracle Configurator window in a host application such as Order Management, iStore, or a custom webstore.

For information about settings needed by Oracle Configurator Developer, see Section 2.8.5, "Parameters in spx.ini for Development and Test" on page 16.

For more information about creating and using a replica Oracle Configurator schema for a mobile deployment, see Section 8.3, "Oracle SellingPoint Mobile Deployment" on page 8-9.

Oracle SellingPoint application requires that the DSNs defined in the spx.ini file map to an installed Oracle Configurator schema. The DSNs set in the spx.ini file must also be registered in the ODBC Administrator for each machine running the Oracle SellingPoint application.

You must edit the spx.ini file and update the [DSN] entries by adding the ODBC DSN(s) you created for your Oracle Configurator schema(s). The entries then appear in the Oracle SellingPoint application list of available data sources at log in when you create a replica for mobile deployment. You must create the Oracle Configurator schema DSN yourself, following the instructions in Section 2.8.3, "Create DSNs and DBOwners" on page 2-11; the spx.ini entries will not work until you create the DSN. The DSN for the replica is created automatically by the InitializeRemoteDB command.

Additional parameters may be defined for manipulating the behavior of the Oracle SellingPoint application (e.g., currency).

Below is a sample spx.ini file followed by sections explaining each of the parameters in the file:

### 2.8.6.1 [MDA]

The section [MDA] lists the parameters for the Oracle SellingPoint application.

#### DBOwner in [MDA]

The parameter DBOwner in the section [MDA] specifies the default username of the owner of the Oracle Configurator schema that this spx.ini file accesses when starting the Oracle SellingPoint application. This must be the DBOwner of the Oracle Configurator schema containing the Oracle Configurator Developer Project(s) containing the configuration model(s) you replicate and deploy in your Oracle SellingPoint application.

The DBOwner username is automatically inserted in the Oracle Configurator schema CZ_END_USERS table when the Oracle Configurator schema is installed with Oracle Applications. End users log into the Oracle SellingPoint application with the name of a user listed in the CZ_END_USERS table, which is the name of an Oracle Applicationsuser .

Every database specified by a Data Source Name (see Section , "DSN in [MDA]") is associated with this DBOwner, unless another DBOwner is specified explicitly for both Oracle Configurator Developer and the Oracle SellingPoint application with the setting:

```
[<DSN>]
DBOwner=<DBOwner>
```

See Section 2.8.5.6, "[<DSN>]".

### DSN in [MDA]

The parameter `DSN` in the section `[MDA]` stores the last Oracle Configurator schema logged into by the Oracle SellingPoint application end user and displays that Data Source Name by default in the Datasource field the next time the end user with this `spx.ini` file logs into the Oracle SellingPoint application.

### LCUST

See Section 2.8.5, "Parameters in spx.ini for Development and Test" on page 2-16.

### LOPP

See Section 2.8.5, "Parameters in spx.ini for Development and Test" on page 2-16.

### LACT

See Section 2.8.5, "Parameters in spx.ini for Development and Test" on page 2-16.

### HLOGO

See Section 2.8.5, "Parameters in spx.ini for Development and Test" on page 2-16.

### SLOGO

See Section 2.8.5, "Parameters in spx.ini for Development and Test" on page 2-16.

### LOGFILE

See Section 2.8.5, "Parameters in spx.ini for Development and Test" on page 2-16.

#### RuntimeCache

See Section 2.8.5, "Parameters in spx.ini for Development and Test" on page 2-16.

#### EngineStartup

See Section 2.8.5, "Parameters in spx.ini for Development and Test" on page 2-16.

### 2.8.6.2 [MDAPLUGINS]

The section `[MDAPLUGINS]` lists programs that can be launched from within the Oracle SellingPoint application. This Feature Is Not Currently Implemented.

### 2.8.6.3 [MDADSNS]

The section `[MDADSNS]` lists the DSNs for the Oracle Configurator schemas to which you want your Oracle SellingPoint application to have access. This parameter must be updated to specify the actual DSNs the end-users need in order to access the Oracle Configurator schema containing their Oracle SellingPoint application.

### 2.8.6.4 [<DSN>]

If a DSN is not listed for every Oracle Configurator schema to which you need access from the Oracle SellingPoint application, it must be listed as a discrete section. A `[<DSN>]` section specifies the DBOwner by which the Oracle Configurator schema associated with the DSN will be accessed.

```
[<DSN>]
DBOwner=<DBOwner>
```

> **Note:** The DBOwner setting here overrides the DBOwner setting in the section `[MDA]`. The DBOwner settings under `[MDA]` are defaults.

### 2.8.6.5 [<DSN>_replica]

In Figure 2–1, "Example spx.ini File", `[vis11]` is the DSN section for an Oracle Configurator schema that has been replicated to a client, and `[vis11_replica]` is the DSN section for the replica. The DSN for a replica database always consists of the server database DSN followed by `_replica`.

A replica’s DSN section has several additional parameters

### DBOwner

The parameter `DBOwner` here specifies the replication user, meaning the user who ran InitializeRemoteDB in the Oracle SellingPoint application.

### Master_name

The parameter `master_name` specifies the DSN of the Oracle Configurator schema from which this replica was created.

### Master_schema_name

The parameter `master_schema_name` specifies the schema name of the Oracle Configurator schema from which this replica was created.

### Connection_type

The parameter `connection_type` specifies the type of connection. The only valid value is 0, which indicates SQL*Net should be used for replication.

### Connection_info

The parameter `connection_info` provides additional connection information. There is no valid value for this parameter because no connection information is needed for SQL*Net.

### ReplicationRefreshMode

The parameter `ReplicationRefreshMode` specifies the type of refresh that should take place when synchronizing the replica client with the server database. The default value is OPTIMUM, meaning refresh finds where the replica client and the server database are out of sync, and updates just that data completely. The value FAST pushes only changes in the replica client to the server database. The value COMPLETE executes a complete synchronization, not just of changes. This setting may result be very slow for large databases.

### Timeout

The parameter `timeout` specifies the number of seconds to elapse before replication times out and returns an error. The default value is 600.

### Olite_version_enum

The parameter olite_version_enum specifies the replication control for Oracle8*i* Lite compatibility. The only valid value is ‘3’, which indicates compatibility with version 4.0.

#### Last_replication

The parameter `last_replication` specifies the date and time of the last replication.

### 2.8.6.6 [CURRENCY]

The section `[CURRENCY]` lists the currency available in the Oracle SellingPoint application. The default `spx.ini` file includes formatting information for displaying prices in USD, FRF, and DEM in the Oracle SellingPoint application.

## 2.8.7 Run Oracle Configurator Developer and the Test Configurator

Using the base `spx.ini` file for development, start Oracle Configurator Developer. Log in as the DBOwner (<oc>/<ocpass>) or <ocdev>/<ocdevpass> or an imported/added user listed in the CZ_END_USERS table.

To run the test configurator from Oracle Configurator Developer, execute the Generate Active Model and Generate Active UI commands, then click Test and log in as a user listed in the CZ_END_USERS table.

Using the base `spx.ini` file for production, start the Oracle Configurator Developer. Oracle Configurator Developer and Oracle SellingPoint application both look for the `spx.ini` file in the `/Windows/ or /Winnt` directory on the local hard drive.

3

# The Oracle Configurator Schema

Oracle runtime configurators use a standard schema for configuration data referred to as the Oracle Configurator schema (CZ tables) in the Oracle Applications database. The Oracle Configurator schema is used to store customer information and all information relative to a configuration model — product data, project structure, configuration rules, and user interface layouts. Customer and product-related data, such as price lists, are generally transferred into the Oracle Configurator schema from data sources external to Oracle Configurator.

## 3.1 Characteristics of the Oracle Configurator Schema

The instance name for the Oracle Configurator schema is identified in the `spx.ini` file. See Section 2.8.5, "Parameters in spx.ini for Development and Test", on page 2-16 for more information about the `spx.ini` file.

The Oracle Configurator schema does not use public synonyms.

The Oracle Configurator schema consists of the following subschemas:

IM - Item-Master
PS - Project Structure
UI - User Interface
LC - Logic for Configuration (Active Model)
PR - Pricing
OM - Opportunity Management
QC - Quotes and Configurations
XF - Transfer specifications and control
GN - General Use tables

The following sections list the tables in each subschema.

### 3.1.1 IM Item-Master

CZ_ITEM_MASTERS
CZ_ITEM_TYPES
CZ_PROPERTIES
CZ_ITEM_TYPE_PROPERTIES
CZ_ITEM_PROPERTY_VALUES
CZ_REL_TYPES
CZ_ITEM_PARENTS

### 3.1.2 PS Project Structure

CZ_DEVL_PROJECTS
CZ_PS_NODES
CZ_DEVL_PRJ_USER_GROUPS
CZ_FUNC_COMP_SPECS
CZ_FUNC_COMP_REFS
CZ_PS_PROP_VALS
CZ_PS_PROPCOMPAT_GEN
CZ_INTL_TEXTS
CZ_LOCALIZED_TEXTS
CZ_LOCALES
CZ_RULES
CZ_RULES_FOLDERS
CZ_DES_CHART_CELLS
CZ_DES_CHART_FEATURES
CZ_POPULATORS
CZ_POPULATOR_MAPS
CZ_FILTER_SETS
CZ_EXPRESSIONS
CZ_EXPRESSION_NODES
CZ_COMBO_FEATURES
CZ_GRID_DEFS
CZ_GRID_COLS
CZ_GRID_CELLS
CZ_SUB_CON_SETS
CZ_POPULATOR_MAPS

### 3.1.3 UI User Interface (Active UI)

CZ_UI_DEFS
CZ_UI_NODES

CZ_UI_PROPERTIES
CZ_UI_NODE_PROPS

### 3.1.4 LC Logic for Configuration (Active Model)

CZ_LCE_HEADERS
CZ_LCE_LINES
CZ_LCE_OPERANDS

### 3.1.5 PR Pricing

CZ_PRICE_GROUPS
CZ_PRICES

### 3.1.6 OM Opportunity Management

CZ_OPPORTUNITY_HDRS
CZ_OPPORTUNITY_HDR_CONTACTS
CZ_CONTACTS
CZ_CUSTOMERS
CZ_ADDRESSES
CZ_ADDRESS_USES
CZ_CUSTOMER_END_USERS
CZ_END_USERS
CZ_END_USER_GROUPS
CZ_USER_GROUPS

### 3.1.7 QC Quotes and Configurations

CZ_PROPOSAL_HDRS
CZ_PROP_QUOTE_HDRS
CZ_QUOTE_HDRS
CZ_QUOTE_ORDERS
CZ_QUOTE_MAIN_ITEMS
CZ_QUOTE_SPARES
CZ_QUOTE_SPECIAL_ITEMS
CZ_DRILL_DOWN_ITEMS
CZ_CONFIG_HDRS
CZ_CONFIG_INPUTS
CZ_CONFIG_INPUT_STRS
CZ_CONFIG_ITEMS

### 3.1.8 XF Transfer Specifications and Control

CZ_XFR_PROJECT_BILLS
CZ_XFR_PRICE_LISTS
CZ_XFR_TABLES
CZ_XFR_FIELDS
CZ_XFR_RUN_INFOS
CZ_XFR_RUN_RESULTS
CZ_XFR_STATUS_CODES
CZ_XFR_FIELD_REQUIRES

### 3.1.9 GN General Use Tables

CZ_DB_LOGS
CZ_DB_SETTINGS

## 3.2 Oracle Configurator Schema Settings

The Oracle Configurator schema provides installation-wide customizable settings that describe the structure and the content of the Oracle Configurator schema, and also give parameters for certain application functions. These settings are stored in the CZ_DB_SETTINGS table.

A CZ_DB_SETTINGS table exists in every Oracle Configurator schema.

You must enter the values your installation requires by modifying the value fields in the CZ_DB_SETTINGS table.

The section names (SECTION_NAME) in the CZ_DB_SETTINGS table are:

DB_USER_ROLES
DATABASE_OWNERS
SCHEMA
ORAPPS_INTEGRATE
IMPORT

Table 3–1, "CZ_DB_SETTINGS" briefly describes these settings. The settings for each table section are described in more detail in the sections following the table.

*Table 3–1 CZ_DB_SETTINGS*

| Setting_ID | Section_name | Value | Default Value | Relevance and Contribution |
|---|---|---|---|---|
| 0 | DB_USER_ROLES | <any_string> | SPX_USER | Role assigned to all users defined in CZ_END_USERS table, including Oracle Applications users transferred from the Oracle Applications database. This is used by the Oracle SellingPoint application only. |
| 1 | DB_USER_ROLES | <any_string> | SPX_DEVELOPER | Role assigned to Oracle Configurator Developer users defined in CZ_END_USERS table. |
| 17 | DATABASE_OWNERS | <any_string> | Null | Name of the Oracle Applications owner <apps>. |
| 18 | DATABASE_OWNERS | <any_string> | Null | Name of the Oracle ApplicationsOrder Management <om> or Order Entry <oe> owner. |
| APPSLINK | DATABASE_OWNERS | <any_string> | Null | Link used for connecting to a remote database server for Oracle Applications <appssid>. |
| AUTOCREATE_IMPORTED_USERS | ORAPPS_INTEGRATE | YES/NO/<br>Named only | NO | Indicates whether or not to enable generically imported users as database users or to enable only the end users with valid login names as database users. This is used for generic import only. |
| BadDefaultPropertyValue | IMPORT | CHAR (1) | F | Indicates the action to be taken when the DEF_VALUE does not match the DATA_TYPE in the CZ_PROPERTIES online table. See Section 3.2.5, "CZ_DB_Settings for IMPORT" for more information about this setting. |
| BadItemPropertyValue | IMPORT | CHAR (1) | F | Indicates the action to be taken when an item’s PROPERTY _VALUE in the CZ_ITEM_PROPERTY_VALUES online table does not match the DATA_TYPE in the CZ_PROPERTIES online table. See Section 3.2.5, "CZ_DB_Settings for IMPORT" for more information about this setting. |
| BOM_REVISION | ORAPPS_INTEGRATE | <any_string> | Null | Indicates the BOM revision in the Oracle Applications database from which data is being imported.Valid values are "5.0.628 for Release 10.7, "11.0.28" for Release 11.0, and "11.5.0" for Release 11*i*. |

*Table 3–1 CZ_DB_SETTINGS*

| Setting_ID | Section_name | Value | Default Value | Relevance and Contribution |
|---|---|---|---|---|
| CommitSize | IMPORT | <integer> | 500 | Indicates the number of transfer or import records to be operated on at a time, between commits. It is recommended that you set this much larger than the expected number of records. |
| CustomerExportEnabled | ORAPPS_ INTEGRATE | YES/NO | YES | Indicates whether Oracle Configurator schema customer information can be transferred to Oracle Applications. |
| CustomerProfileClassName | ORAPPS_ INTEGRATE | ‘Default’ | Oracle Receivables default profile | Indicates the customer profile for Oracle Receivables. |
| DEFAULT_ITEM_TYPE | ORAPPS_ INTEGRATE | <integer> | 0 | The item type id assigned to import items by default. |
| DefaultPriceGroupID | ORAPPS_ INTEGRATE | Price_Group_ ID | Null | Price_Group in Oracle Configurator schema is applied. This is used if using the Oracle SellingPoint application 4.2.x for a mobile deployment where OraclePricing=N. |
| DefaultSOPriceID | ORAPPS_ INTEGRATE | Orig_Sys_Ref | Null | SO_Price_List is applied. DefaultSOPriceID, in SO_Price_List table, is the ID of an EMPTY price list in Oracle Applications. This value is system allocated when the Oracle Applications price list is created. |
| DiscountID | ORAPPS_ INTEGRATE | Discount_ID | Null | Discount in Oracle Configurator schema is applied. DiscountID=<discount_id>, where <discount_id> is associated with the price list identified by DefaultSOPriceID in Oracle Applications. This value is allocated when the discount is created. |
| MAJOR_VERSION | SCHEMA | System setting | System setting | Indicates the major version label for the Oracle Configurator schema. |
| MaximumErrors | IMPORT | <integer> | 10000 | Default error limit for concurrent program transfers and generic import runs before aborting. |
| MINOR_VERSION | SCHEMA | System setting | System setting | Indicates the minor version label for the Oracle Configurator schema. |

*Table 3–1 CZ_DB_SETTINGS*

| Setting_ID | Section_name | Value | Default Value | Relevance and Contribution |
|---|---|---|---|---|
| MULTISESSION | IMPORT | <integer> | 0 | A positive value indicates the number of seconds to wait, checking for current state every second and waiting while another import runs. After this number of seconds has elapsed, control goes to the waiting import session if no other session is active, or an exception is raised if another import session is still running. A value of 0 means do not wait, and raise an exception immediately if another import session is already running. Any negative value means ignore other sessions and run immediately. |
| OraclePricing | ORAPPS_ INTEGRATE | YES/NO | NO | Indicates whether or not Oracle Pricing is being used for configurations or orders.<br>If this is set to "NO", you must also set:<br>DefaultSOPriceID in SO_Price_Lists table, is the ID of an EMPTY price list in Oracle Applications. This value is system allocated when the Oracle Applications' price list is created.<br>DiscountID=<discount_id>, where <discount_id> is associated with the price list identified by DefaultSOPriceID in Oracle Applications. This value is allocated when the discount is created.<br>This is used if using the Oracle SellingPoint application 4.2.x for a mobile deployment where OraclePricing=N. |
| OracleSequenceIncr | SCHEMA | <integer> | 20 | An integer (default=20) that indicates the number of primary-key values allocated by each use of a sequence. The default means that keys are assigned in increments of 20. |
| OrderEntry | ORAPPS_ INTEGRATE | YES/NO | NO | Indicates whether or not transfer of quote/order information to Order Entry is enabled. |
| OrderImportSourceID | ORAPPS_ INTEGRATE | Order_Import_ Source_ID | Null | The import source ID in Oracle Applications for the Oracle Configurator schema from which orders will be transferred back into Oracle Applications. |

***Table 3–1 CZ_DB_SETTINGS***

| Setting_ID | Section_name | Value | Default Value | Relevance and Contribution |
|---|---|---|---|---|
| OrderTypeID | ORAPPS_ INTEGRATE | Order_Type_ ID | Null | Specifies the type of order in Oracle Applications that is being transferred from the Oracle Configurator schema. |
| PsNodeName | ORAPPS_ INTEGRATE | ‘Segment1’ (or Description) | Null | Indicates the source field to be loaded into the Name field in the CZ_PS_ NODES table. |
| RefPartNbr | ORAPPS_ INTEGRATE | ‘Segment1’ (or Description) | Segment1 | Identifies the field to be used for the name of a transferred or imported item in the Oracle Configurator schema Item Master and Project Structure tables. |
| RepConType | SCHEMA | 0/1 | 0 | Indicates the method to be used for replication. ’0’ indicates use of SQL*Net. ’1’ indicates file-based replication. |
| RepConInfo | SCHEMA | no value/ ’oma-network’ | no value | No connection information is needed for SQL*Net. Enter ’oma-network’ for file-based replication. |
| Replication | SCHEMA | YES/NO | NO | Indicates whether or not replication is active on this database. The `rep_ setup.sql` script sets the value to ’YES’. |
| RepOliteDriver | SCHEMA | Oracle Lite 40 ODBC Driver | Oracle Lite 40 ODBC Driver | Specifies which driver to use when the Initialize Remote DB command in the Oracle SellingPoint application creates an ODBC DSN for the replica database. The only valid value is Oracle Lite 40 ODBC Driver. |
| RepOliteVersion | SCHEMA | 3 | 3 | Specifies the replication control for Oracle8*i* Lite compatibility. The only valid value is ‘3’, which indicates compatibility with version 4.0. |
| RepTimeout | SCHEMA | <integer> | 600 | The number of seconds to elapse before replication times out and returns an error. |
| RUN_BILL_EXPLODER | ORAPPS_ INTEGRATE | YES/NO | YES | Indicates whether or not to run the BOM_EXPLODER procedure. |

*Table 3–1 CZ_DB_SETTINGS*

| Setting_ID | Section_name | Value | Default Value | Relevance and Contribution |
|---|---|---|---|---|
| SpxDefaultTablespace | SCHEMA | <any_string> | Null | Indicates the default permanent tablespace to be used by <spx>, <spxdev>, and end users for permanent tables. |
| SpxIndxTablespace | SCHEMA | <any_string> | Null | Indicates the tablespace used for table indexes. Use this tablespace to store indexes in an area different than your default tablespace. If this tablespace is not specified, table indexes are stored in the SpxDefaultTablespace. |
| SpxTemporaryTablespace | SCHEMA | <any_string> | Null | Indicates the default temporary tablespace to be used by <spx>, <spxdev>, and end users for temporary tables. |

## 3.2.1 CZ_DB_Settings for DB_USER_ROLES

The settings in the DB_USER_ROLES section list which roles are granted to users accessing the Oracle Configurator schema and the tablespace to which <oc> and <ocdev> are assigned.

There are two Setting_IDs in the DB_USER_ROLES section of the CZ_DB_SETTINGS table. The value of the Setting_ID **0** is used for the default end user role. If no value is given, the value, SPX_USER is used. The value can be any string (data type VARCHAR2). The value of the Setting_ID **1** is used for the default Oracle Configurator Developer user role. If no value is given, the value SPX_DEVELOPER is used. The value can be any string (data type VARCHAR2).

The role SPX_USER is assigned by default to all users defined in the CZ_END_USERS table, including Oracle Applications users who use the Oracle SellingPoint application for testing.

When granted to a user, this role gives the user access to the Oracle Configurator schema with RESOURCE privileges.

## 3.2.2 CZ_DB_Settings for DATABASE_OWNERS

The settings in the DATABASE_OWNERS section list names of the Oracle Applications <apps> and the database link to <appsid> if it is remote. These CZ_DB_SETTINGS are used by OC concurrent programs for integration with Oracle Applications Release 10.7 or 11.0.

### 3.2.3 CZ_DB_Settings for SCHEMA

The settings in the SCHEMA section control general parameters of the whole Oracle Configurator schema.

**MAJOR_VERSION** is the major version label for the Oracle Configurator schema .

**MINOR_VERSION** is the minor version label for the Oracle Configurator schema .

**OracleSequenceIncr** is an integer (default=20) that indicates the number of primary-key values allocated by each use of a sequence. The default means that keys are assigned in increments of 20.

See Section 3.3.3, "Redo Sequences" on page 3-21 for more information about the use of this setting.

**Replication** is a YES/NO flag (default=NO) that determines whether replication is enabled or not on the Oracle Configurator schema. This should only be set to "Yes" for a mobile deployment. See Section 8.3, "Oracle SellingPoint Mobile Deployment", on page 8-9 for more information about mobile deployment.

**RepConType** is an integer (default=0) that determines whether the replication method is using SQL*Net (0) or file-based (1).

**RepConInfo** provides the replication connection information. No connection information is needed for SQL*Net, therefore, no value (default) is entered if using SQL*Net. Enter 'oma-network' if you are using file-based replication.

**RepOliteDriver** specifies which driver to use when the Initialize Remote DB command in the Oracle SellingPoint application creates an ODBC DSN for the replica database. The only valid value is 'Oracle Lite 40 ODBC Driver'.

**RepOliteVersion,** if using Oracle 8i Lite, is the integer value for this setting specifies the replication control for Oracle8*i* Lite compatibility. The only valid value is '3', which indicates compatibility with version 4.0.

**RepTimeout** is an integer value which indicates the number of seconds you want to elapse before replication times out and returns an error. Default is 600 seconds.

**SpxDefaultTablespace** holds the default permanent tablespace used by <oc>, <ocdev>, and end users for permanent tables. The value of the SCHEMA SpxDefaultTablespace is <defaultspace>.

**SpxTemporaryTablespace** holds the default temporary tablespace used by <oc>, <ocdev>, and end users for temporary tables. The value of the Setting_ID SpxTemporaryTablespace is <tempspace>.

**SpxIndxTablespace** holds the tablespace used for table indexes. Use this tablespace to store indexes in an area different than your default tablespace. If this tablespace is not specified, table indexes are stored in the SpxDefaultTablespace. The value of the Setting_ID SpxTemporaryTablespace is <indxspace>

### 3.2.4 CZ_DB_Settings for ORAAPPS_INTEGRATE

The settings in the ORAPPS_INTEGRATE section control how and what gets transferred or imported to and from the Oracle Configurator schema.

**AUTOCREATE_IMPORTED_USERS** is a YES / NO / NAMED_ONLY flag (default=NO) used for generic import only, that indicates whether, at the end of a generic import run, end users listed in the CZ_END_USERS table should also be enabled as database users. If the setting of AUTOCREATE_IMPORTED_USERS is ’NO’, all transferred or imported users are only imported into the CZ_END_USERS table, but not enabled as database users. To enable end users as database users, use the OC SQL*Plus script EndUsers.sql.

If the setting of AUTOCREATE_IMPORTED_USERS is ’YES’, all transferred or imported users are enabled as database users. Users with no login names in the Oracle Applications database (LOGIN_NAME=null), are assigned names (e.g., <spx_*n*>). To transfer or import only users with existing login names, use the value 'NAMED_ONLY'. Specifying ’NAMED_ONLY’ means all transferred or imported users whose LOGIN_NAME is not null are added to the CZ_END_USERS table in the Oracle Configurator schema and enabled as database users.

The CZ_END_USERS.LOGIN_NAME is populated from RA_SALESREPS.EMAIL_ADDRESS (to see value query RA_SALESREPS_ALL table). If the email address contains an '@', everything up to that symbol is extracted and placed in the login_name field. If the email address does not contain an '@' and it is not null, then the entire email address is placed into the login_name field. If the email address is null then the login_name field is automatically generated using the schema name and a unique sequential number (i.e., <spx>_<unique number>).

End users enabled as database users will have a login of CZ_END_USERS.LOGIN_NAME and a password of CZ_END_USERS.LOGIN_NAME.

Table 3–2, "Example AUTOCREATE_IMPORTED_USERS DB Settings Results" below shows examples of the possible DB User Name results with each AUTOCRATE_IMPORTED_USERS setting scenario.

*Table 3–2 Example AUTOCREATE_IMPORTED_USERS DB Settings Results*

| AUTOCREATE_ IMPORTED_ USERS DB Setting | RA_ SALESREPS.EMAIL_ ADDRESS | CZ_END_ USERS.LOGIN _NAME | Resulting Database User Name/ Password |
|---|---|---|---|
| YES | sam@comp.com | sam | sam / sam |
| YES | sam | sam | sam / sam |
| YES | sam_sales@comp.com | sam_sales | sam_sales / sam_ sales |
| YES | null | spx_1000 | spx_1000 / spx_1000 |
| NO | sam@comp.com | sam | none |
| NO | sam | sam | none |
| NO | sam_sales@comp.com | sam_sales | none |
| NO | null | none | none |
| NAMED_ONLY | sam@comp.com | sam | sam / sam |
| NAMED_ONLY | sam | sam | sam / sam |
| NAMED_ONLY | sam_sales@comp.com | sam_sales | sam_sales / sam_ sales |
| NAMED_ONLY | null | spx_1000 | none |
| NULL | | none | none |

**BOM_REVISION** indicates the version in the Oracle Applications database from which BOM data is being imported. The date format used for Oracle Applications Releases 10.7 or 11.0 is "DD / MON / RR" and Release 11i uses a "YYYY-MM-DD" format. This setting is checked to ensure that the correct date format is used in the call to the BOM explosion procedure. Valid values are "5.0.628" for Release 10.7, "11.0.28" for Release 11.0, and "11.5.0" for Release 11*i*. If null, "11.5.0" is used.

**CustomerExportEnabled** is a YES / NO flag (default=YES) that indicates whether Oracle Configurator schema customers can be transferred to Oracle Applications.

**CustomerProfileClassName** indicates the customer profile. Default is the profile for Oracle Receivables.

**DefaultItemType** indicates the item type id assigned to import items by default.

**DefaultPriceGroupID** is the <price_group_id> (default='-1' (null)) of a price group defined in the Oracle Configurator schema where this DB_SETTING is in effect. The value of <price_group_id> comes from CZ_PRICE_GROUPS.price_group_id in the Oracle Configurator schema.

DefaultPriceGroupID is the default selection on the Quote header screen in the Oracle SellingPoint application. The default is Price_group 1. When DB setting OraclePricing='NO', the prices from the selected Price_Group_ID in the Oracle Configurator schema are exported with the order into Oracle Applications Order Management (Release 11i) or Order Entry (Release 10.7 or 11.0).

**DefaultSOPriceID** is the <Orig_Sys_Ref> (default='-1' (null)) of an empty price list defined in the Oracle Applications database that will be transferred into the Oracle Configurator schema where this DB_SETTING is in effect. The value of <Orig_Sys_Ref> comes from SO_PRICE_LISTS.price_list_id in the Oracle Applications database, where price_list_id identifies an EMPTY price list.

DefaultSOPriceID is always used when the DB setting OraclePricing='NO'. When the DB setting OraclePricing='NO', the Orig_Sys_Ref in SO_Price_List that is specified to be applied by default should identify an EMPTY Oracle Applications price list

Orig_Sys_Ref is a compound value. The first part is the ORGANIZATION_ID, the second is the FND user ID (from Oracle Applications).

DefaultSOPriceID is not used when the DB setting OraclePricing='YES'. Only transferred or imported, non-EMPTY SOPriceIDs are used when OraclePricing='YES'.

**DiscountID** is the <discount_id> (default='-1' (null)) defined in the Oracle Applications database that will be transferred into the Oracle Configurator schema where this DB_SETTING is in effect. The value of <discount_id> comes from SO_DISCOUNTS.discount_id in the Oracle Applications database, where discount_id identifies that discounting can be applied to an order with non-Oracle Pricing in Oracle Applications.

DiscountID is only used when the DB setting OraclePricing='NO'. DiscountID allows a manually applied order-level discount to Oracle Applications. The discount identified by DiscountID must be overrideable and associated with the DefaultSOPriceID price list, which is always an EMPTY price list.

**OraclePricing** is a YES/NO flag (default=NO) that determines whether submitted orders from the Oracle Configurator schema have Oracle Pricing applied or not. If 'YES', Oracle Applications Order Management (Release 11i) or Order Entry (Release 10.7 or 11.0) will calculate and price the order itself. If 'NO', Oracle runtime configurator prices will override prices in the Order Management or Order Entry system.

**OrderEntry** is a YES / NO flag (default=NO) that determines whether integration with Oracle Applications Order Management (Release 11i) or Order Entry (Release 10.7 or 11.0) is enabled or not.

**OrderImportSourceID** is the <Order_Import_Source_ID> (default=‘**-**1’ (null)) in the Oracle Applications database for identifying where an order should be transferred from. The value comes from SO_ORDER_SOURCES.order_source_id in the Oracle Applications database. OrderImportSourceID is an import source ID in Oracle Applications for the Oracle Configurator schema from which an order will be transferred into Oracle Applications Order Management (Release 11i) or exported to Order Entry (Release 10.7 or 11.0). In the Oracle Configurator schema that is identified as the import or transfer source: use_ids=true.

OrderImportSourceID must be populated in order for Oracle Applications Order Management or Order Entry to work (OrderEntry='YES'). Set OrderImportSourceID in the CZ_DB_SETTINGS table before submitting an order.

**OrderTypeID** is the <Order_Type_ID> (default=‘**-**1’ (null)). The value comes from SO_ORDER_TYPES.order_type_id in the Oracle Applications database. OrderTypeID specifies the type of order in Oracle Applications that is being transferred from the Oracle Configurator schema. OrderTypeID must be populated for Oracle Applications Order Entry to work (OrderEntry='YES'). Set OrderTypeID in the CZ_DB_SETTINGS table before submitting an order.

**PsNodeName** is the <field name> (default=‘Segment1’) that indicates the source field to be loaded into the Name field in the CZ_PS_NODES (project or model structure) table in the Oracle Configurator schema. 'Segment1' is used by default so that the name loaded into the model structure in Oracle Configurator Developer will match the names in Item_Master.

**RefPartNbr** is the <field name> (default=‘Segment1’) that indicates the source field to be loaded from the BOM_EXPLOSIONS table into Ref_Part_Nbr in Item_Master in the Oracle Configurator schema. 'Segment1' is the usual field that contains the name for an item, so this is the default for retrieving the name to be displayed in the Item Master in Oracle Configurator Developer. To display the actual part numbers of items in the Item Master, set RefPartNbr to the name of the field populated with part numbers.

**RUN_BILL_EXPLODER** is a YES / NO flag (default=NO) that indicates whether the Oracle Applications Bills of Material exploder should be run on each bill that is marked for transfer or import in the CZ_XFR_PROJECT_BILLS table in the Oracle Configurator schema at the time of data transfer or import.

The OC concurrent programs load bills and items based on top bills listed in the CZ_XFR_PROJECT_BILLS table in the Oracle Configurator schema. Before extracting, if RUN_BILL_EXPLODER is set to YES, the procedure calls the BOM exploder to refresh data in BOM_EXPLOSIONS for each record in the CZ_XFR_PROJECT_BILLS table. If RUN_BILL_EXPLODER is set to NO, the concurrent programs or generic import scripts will transfer the BOMs that are flagged for import in the CZ_XFR_PROJECT_BILLS table without running the BOM exploder first.

### 3.2.5 CZ_DB_Settings for IMPORT

The settings in the IMPORT section are for controlling how the import executes.

**BadDefaultPropertyValue** is a string (default ’F’) that indicates the action to be taken when the DEF_VALUE in the CZ_IMP_PROPERTY table does not match the DATA_TYPE for transfer or import into the CZ_PROPERTIES online table. The valid values for this setting are:

| Value | Disposition |
|---|---|
| ’R’ | Reject the record in the import table and use the old DEF_VALUE |
| ’F’ | Force the record to be updated to include the DEF_VALUE from the import table |
| ’K’ | Update all information in the record except the DEF_VALUE |

**BadItemPropertyValue** is a string (default ’F’) that indicates the action to be taken when an item’s PROPERTY _VALUE in the CZ_IMP_ITEM_PROP_VALUES table does not match the DATA_TYPE in the CZ_PROPERTIES online table so it can be transferred or imported into the CZ_ITEM_PROPERTY_VALUES online table. The valid values for this setting are:

| Value | Disposition |
|---|---|
| ’R’ | Reject the record in the import table and use the old PROPERTY_VALUE |
| ’F’ | Force the record to be updated to include the PROPERTY_VALUE from the import table |
| ’K’ | Update all information in the record except the item PROPERTY_VALUE |

| Value | Disposition |
|---|---|
| ’X’ | Reject the record and logically delete any matching item property value record in the CZ_ITEM_PROPERTY_VALUES table. This makes the item property value default to the property default value in the CZ_ITEM_PROPERTY_VALUES table |

**CommitSize** is an <integer> (default=‘500') that indicates the number of records to be operated on at a time, between commits.

**MaximumErrors** is an <integer> (default=‘10000') that indicates the default error limit for transfer or import runs before aborting. If you have a large amount of data to transfer or import or you aren’t concerned with the process stopping once a certain number of errors is reached, set this to an extremely large number.

**MULTISESSION** is an <integer> (default=‘0') that indicates the number of seconds to wait for another import session to complete if another import is running. A positive value indicates the number of seconds to wait, checking for current state every second and waiting while another import runs. After this number of seconds has elapsed, control goes to the waiting import session if no other session is active, or an exception is raised if another import session is still running. A value of 0 means do not wait, and raise an exception immediately if another import session is already running. When MULTISESSION is missing from the CZ_DB_SETTINGS table, it is as if it were set to the default, 0.

Any negative value means ignore other sessions and run immediately. Setting this parameter to a negative number is equivalent to disabling it.

If an import session is aborted, the CZ_XFR_RUN_INFOS table may end up in an inconsistent state with the value of COMPLETED not ’1’. If then MULTISESSION is not disabled, a new import session cannot run.

See Section 4.3.1, "Setup for Generic Import", on page 4-24 for more detailed information about all CZ_DB_Settings that apply to generic import.

## 3.3 Oracle Configurator Schema Maintenance

Oracle Configurator provides SQL*Plus packages for maintaining the Oracle Configurator schema and its various subschemas.

*Table 3–3 Oracle Configurator Schema Maintenance Packages*

| Package | Invokes administrative packages for... |
|---|---|
| CZ_MANAGER | the Oracle Configurator schema |
| CZ_GN_MGR | general subschemas |
| CZ_IM_MGR | the item master subschema |
| CZ_LC_MGR | LCE compiled logic tables |
| CZ_OM_MGR | the opportunity management subschema |
| CZ_PR_MGR | the pricing subschema |
| CZ_PS_MGR | the product structure subschema |
| CZ_QC_MGR | quotes and configurations subschema |
| CZ_UI_MGR | the user interface subschema |
| CZ_XFR_MGR | transfer (import, export, integration) subschemas |

These packages report problems or information using the CZ_UTILS.REPORT function, which routes informational messages to the CZ_DB_LOGS table and to the SQL*Plus display. These maintenance packages may invoke additional administrative packages, such as PURGE, REDO_SEQUENCES, TRIGGERS_ENABLED, or CONSTRAINTS_ENABLED to perform specific functions.

## 3.3.1 Refresh or Update the Production Schema

Oracle Configurator provides a data refresh utility for updating configuration models in a production Oracle Configurator schema with changes made in a development Oracle Configurator schema. The utility ensures that existing production data, e.g. saved configuration data, is preserved.

The refresh utility refreshes the following tables when updating one Oracle Configurator schema to another, such as updating a production version of the Oracle Configurator schema whenever there is a change in the corresponding development or maintenance version of the schema.

### 3.3.1.1 Tables Requiring Refresh

CZ_COMBO_FEATURES
CZ_DES_CHART_CELLS

CZ_DEVL_PROJECTS
CZ_EXPRESSIONS
CZ_EXPRESSION_NODES
CZ_FILTER_SETS
CZ_FUNC_COMP_SPECS
CZ_GRID_CELLS
CZ_GRID_COLS
CZ_GRID_DEFS
CZ_INTL_TEXTS
CZ_ITEM_MASTERS
CZ_ITEM_PARENTS
CZ_ITEM_TYPE_PROPERTIES
CZ_ITEM_PROPERTY_VALUES
CZ_ITEM_TYPES
CZ_LCE_HEADERS
CZ_LCE_TEXTS
CZ_LOCALES
CZ_POPULATORS
CZ_POPULATOR_MAPS
CZ_PROPERTIES
CZ_PS_NODES
CZ_PS_PROP_VALS
CZ_REL_TYPES
CZ_RULES
CZ_RULES_FOLDERS
CZ_SUB_CON_SETS
CZ_UI_DEFS
CZ_UI_NODES
CZ_UI_PROPERTIES
CZ_UI_NODE_PROPS

**Note:** Many of these tables are not referenced by the Oracle runtime configurators.

### 3.3.1.2 Refresh Utility Prerequisites

- Utilities *exp80 and imp80* must be installed on the client machine.
- The table CZ_DB_SETTINGS of the production schema must be updated using the following statement prior to the execution of the refresh utility

```
Insert into cz_db_settings (setting_id, section_name, data_type, value,
desc_text) values ('ProductionStatus', 'SCHEMA',  4, 'YES', 'Schema is
production if value is YES, schema is considered to be development
otherwise');
```

- All scripts (`cz_refresh.sql`, `cz_exp.sql`, `cz_imp.sql`, `cz_temp.sql`, `cz_bexp.sql`, `cz_proc.sql`), which are available in the DBAdmin/ folder on the Oracle Configurator Developer CD, must be located in the same directory.

#### 3.3.1.3 Running the Refresh Utility

Use the following procedure to refresh all configuration models in a production Oracle Configurator schema with changes made in a development Oracle Configurator schema:

1. Connect to the Oracle Configurator schema (CZ) in the Release 11*i* Oracle Applications instance. See Section 1.8.1.1, "Connect to a Database Instance" on page 1-14 for details.
2. Run SQL*Plus in the <OC-scripts> directory connected to the instance <ocsid>. See Section 1.8.1.3, "Run SQL*Plus in the <OC-scripts> Directory" on page 1-15 for instructions on how to do this.
3. Run the `cz_refresh.sql` script.

   For example:

   ```
   SQL>  @cz_refresh
   ```

4. Enter the username (<oc>), password (<ocpass>), and TNS service name (<TNSalias>) of the source development schema.
5. Enter the username (<oc>), password (<ocpass>), and TNS service name (<TNSalias>) of the target production schema.

   The script exports the source development schema to a dump (.dmp) file.

6. Enter a filename for the source dump file.

   The script creates a temporary schema and imports the source dump file into it.

   Once the source dump file is imported into a temporary schema, the script creates a backup dump file of the target production tables before performing the refresh.

7. Enter a filename for the backup target dump file.

8. When the backup dump file is completed, the refresh procedure automatically begins.

   The refresh script creates a log file `cz_refresh.log` in the current directory.

As a precaution, include refreshing the sequences when updating one Oracle Configurator schema to another. See Section 3.3.3, "Redo Sequences" on page 3-21 for more information.

For additional information about refreshing data in your Oracle Configurator schema, see Section 4.5, "Refresh and Update Imported Data" on page 4-30.

### 3.3.2 Purge

PURGE is invoked by the packages CZ_MANAGER.sql and CZ_<subschema>_ MGR.sql (i.e., CZ_UI_MGR.sql).

PURGE physically deletes all logically-deleted records in the tables and subschemas. In some cases Purge will propagate deletions to additional records not marked as deleted, in the same or different tables. For example, PURGE will physically delete children of a logically-deleted PS_NODE record. PURGE will also delete all EXPRESSION_NODE records attached to a deleted EXPRESSION. In other cases, PURGE will not physically delete a logically-deleted record owing to a non-deleted reference to that record; e.g., a rule referring to a deleted PS_NODE will prevent PURGE from physically deleting the PS_NODE.

Each table has delete-propagation rules describing these characteristics.

When databases get large and performance slows down, use PURGE procedures to remove all logically-deleted items:

1. Locate the CZ_MANAGER.sql or CZ_<subschema>_MGR.sql package in the `DBAdmin` folder.
2. Create a dump (.dmp) file of your database using Oracle8 exp80.
3. Connect to your database as the DBOwner in SQL*Plus.
4. Issue the command `exec <sql_script>.purge;`, where <sql_script> is the name of a cz manager package in the database (i.e. CZ_PS_MGR.sql).
5. Inspect your data using Oracle Configurator Developer.

### 3.3.3 Redo Sequences

REDO_SEQUENCES is invoked by the packages CZ_MANAGER.sql and CZ_<subschema>_MGR.sql (i.e., CZ_PS_MGR.sql).

Depending on the argument given, REDO_SEQUENCES alters or recreates the sequence objects in the database that are used to allocate primary keys for tables in the subschema. The procedure checks the high primary-key value currently in the database and sets a new start value that is higher. It uses the default incremental value specified by ‘OracleSequenceIncr' in the CZ_DB_SETTINGS table unless you specify a new increment. When running the CZ_MANAGER package, the optional new increment argument does not change the default increment in CZ_DB_SETTINGS. To change the default increment, you must change the value of ’OracleSequenceIncr’ in the CZ_DB_SETTINGS table.

For example: CZ_MANAGER.REDO_SEQUENCES (’0’, ’5’) alters the existing sequence by the specified increment of 5. CZ_MANAGER.REDO_SEQUENCES (’1’,’15’) drops the existing sequence and creates a new sequence starting with the high primary-key value currently in the database and increments it by 15 for the new start value. If a new increment value was not specified in either case, the value of ’Oracle SequenceIncr’ in the CZ_DB_SETTINGS table would be used.

### 3.3.4 Enable Triggers

TRIGGERS_ENABLED is invoked by the packages CZ_MANAGER.sql and CZ_<subschema>_MGR.sql (i.e., CZ_PS_MGR.sql).

Depending on the argument it is given, TRIGGERS_ENABLED either enables or disables the triggers attached to the tables in a subschema. The default is ’1’ to enable triggers.

For example: CZ_MANAGER.TRIGGERS_ENABLED (’1’) enables triggers. CZ_MANAGER.TRIGGERS_ENABLED (’0’) disables triggers.

### 3.3.5 Enable Constraints

CONSTRAINTS_ENABLED is invoked by the packages CZ_MANAGER.sql and CZ_<subschema>_MGR.sql (i.e., CZ_PS_MGR.sql).

Depending on the argument it is given, CONSTRAINTS_ENABLED either enables or disables the constraints (e.g., foreign keys) attached to the tables in the subschema. The default is ’1’ to enable constraints.

For example: CZ_MANAGER.CONSTRAINTS_ENABLED (’1’) enables constraints. CZ_MANAGER.CONSTRAINTS_ENABLED (’0’) disables constraints.

## 3.4 Make a Copy of an Oracle Configurator Schema

In order to make a copy of an Oracle Configurator schema, create a dump (.dmp) file. Oracle recommends using the exp or exp80 command (logged in as the schema owner) to make a dump file of your Oracle Configurator schema before upgrading to a new release. A new release of OC may or may not require upgrading to a new MAJOR_VERSION and/or MINOR_VERSION of the schema. A copy (or dump file) of your Oracle Configurator schema allows you to revert back to the existing schema in the event of upgrade problems. See the *Oracle Configurator and SellingPoint ReadMe* for more information about upgrading.

The following sections explain how to import a dump file to create a copy of your Oracle Configurator schema. For information on how to create a dump file from an existing Oracle Configurator schema, see the Oracle documentation on the exp and exp80 commands.

### 3.4.1 Prerequisites for Importing a Dump File

The following must be complete for a successful import of a dump file:

1. There must be a DBOwner (schema owner) in your Oracle Configurator schema instance into which to import the dump file. In this document, this user and its password are referred to as <impdump>/<impdumppass>
2. The <impdump> schema must be empty of user tables and user objects. See Section 3.4.2, "Create an Empty Schema" on page 3-22.
3. The role SPX_USER must exist in your Oracle Configurator schema.
4. You must know the name of the schema from which the dump file was exported <expdump>.
5. The Oracle8*i* Enterprise Edition Import (imp or imp80) and Export (exp or exp80) utilities must be installed and in your command shell path.

### 3.4.2 Create an Empty Schema

In order to import (or re-import) a dump file into your Oracle Configurator schema, you must create a schema that is empty of user tables and user objects. The empty schema might be a previously used schema or one you create from scratch.

1. **Create a Schema from Scratch**

To create a schema from scratch, you must have DBA privileges and run the following SQL*Plus procedure:

```
SQL> create user <username> identified by <password>
default tablespace <user data file>
temporary tablespace <temporary data file>
quota <limit> on <user data file>
quota <limit> on <temporary data file>;

grant <grants> to <username>;
```

2. **Empty an Existing Schema**

   **CAUTION:** Performing the following procedure will remove all schema objects and data from the schema you are connected to.

   To remove user tables and user objects from an existing Oracle Configurator schema such as <oc>:

   a. Start SQL*Plus, and connect to the existing schema you want to empty.

      Example:

      ```
      SQL> conn <oc>/<ocpass>@<ocsid>
      ```

   b. Under Options / Environment, set pagesize to 999 or enter the command:

      ```
      SQL> SET PAGESIZE 999
      ```

3. Enter the following command, retaining all space characters:

   ```
   SQL> SELECT ’DROP TABLE ’ || TABLE_NAME || ’ CASCADE CONSTRAINTS;
   ’ AS DROPTABS_CMD FROM USER_TABLES;
   ```

   This command will cause SQL*Plus to print a large set of SQL commands.

4. Copy the set of SQL commands that resulted from the previous step.

5. Paste the set of commands at the SQL*Plus command prompt.

   Each command line is processed, dropping all the user tables.

6. Enter the following command, retaining all space characters:

   ```
   SQL> SELECT ’DROP ’ || OBJECT_TYPE || ’ ’ || OBJECT_NAME || ’;’
   AS DROPALL_CMD FROM USER_OBJECTS ORDER BY OBJECT_TYPE DESC,
   OBJECT_NAME ASC;
   ```

   This step will cause SQL*Plus to print a large set of SQL commands.

7. Copy and paste the set of generated SQL commands.
8. Confirm that you have deleted all user objects, by entering this command:

   ```
   SQL> SELECT OBJECT_ID, OBJECT_NAME FROM USER_OBJECTS;
   ```

   The response should be:

   ```
   no rows selected
   ```

   > **Note:** The set of command lines generated by steps 3 and 6 may exceed the ability of SQL*Plus to paste them all at once. The symptom of this problem is that the processing stops on an incomplete command line, or before the last command in the copied set is processed. If this happens, repeat the process of generating, copying, and pasting the commands.

9. Disconnect from the database. You can exit from SQL*Plus.

   ```
   SQL> DISC
   SQL> EXIT
   ```

10. Import or re-import the dump file following the instructions in Section 3.4.3, "Import a Dump File" on page 3-24.

### 3.4.3 Import a Dump File

To import a dump files:

1. Identify the directory where the dump file is located (<dumpfile>).
2. Open a system command shell (UNIX or DOS) and navigate to the directory where you placed the dump file.
3. Enter the import command as a single line.

   Example (for NT command shell):

   ```
   imp80 <impdump>/<impdumppass>@<spsxid> fromuser=<expdump> touser=<impdump>
   recordlength=4096 file=<dumpfile> log=<logfile>
   ```

   where <expdump> is the name of the schema from which the dump file was exported, <dumpfile> is the name of the dump file you are importing (e.g., TutorialServer.dmp), and <logfile> is the name you want to give the log file on the imp or imp80 command (e.g., TutImp.log).

> **Note:** You can also put the above command in a batch file, as a single line.

> **Note:** Be sure that the <logfile> name you choose does not cause the total length of the import command to exceed the limits of your command processing shell.

4. While the import is running, you will see a series of messages about what is being imported. When the import is finished, you will see a message like the following:

   ```
   Import terminated successfully without warnings.
   ```

   The messages from the import are written to the <logfile>.

   After the import finishes, you can close the command shell.

5. Start SQL*Plus, and connect to <impdump>:

   ```
   SQL> conn <impdump>/<impdumppass>@<spxsid>
   ```

6. Update the user recorded as the owner of the imported data, by entering the following command:

   ```
   SQL> UPDATE CZ_END_USERS SET LOGIN_NAME = USER WHERE UPPER(LOGIN_NAME) =
   '<expdump>';
   ```

   Example: (note case-sensitivity):

   ```
   SQL> UPDATE CZ_END_USERS SET LOGIN_NAME = USER WHERE UPPER(LOGIN_
   NAME) = 'DOCMARK';
   ```

7. Commit the change to the database:

   ```
   SQL> COMMIT;
   ```

8. Disconnect from the database. You can exit from SQL*Plus.

```
SQL> DISC
SQL> EXIT
```

## 3.4.4 Verify an Imported Dump File

Any database imported from a dump file should be verified for invalid objects prior to using it. There are multiple reasons why some objects may become invalid after importing form a dump file and it may be impossible and not necessary to make them all valid, but you must ensure that the critical objects are valid. Therefore, after importing an Oracle Configurator schema from a dump file:

1. Start SQL*Plus, and connect to <impdump>:

   ```
   SQL> conn <impdump>/<impdumppass>@<spxsid>
   ```

2. Run the czclnup.sql script. This displays a list of invalid objects, trys to recompile them, and displays a list of any objects that fail the recompilation attempt.

   ```
   SQL> @czclnup.sql
   ```

3. Decide which objects are critical for testing this database (i.e., CZ_UTILS_S and CZ_UTILS_B are always critical).
4. Manually recompile critical objects from sources relevant to the build used to create the original database (this is especially important in the event you need to reproduce a problem).
5. Run the czclnup.sql script again and make sure that all the critical objects are valid.
6. Disconnect from the database. You can exit from SQL*Plus.

   ```
   SQL> DISC
   SQL> EXIT
   ```

**Note:** These steps are specific to an OC schema, but any schema requires similar verification after being imported from a dump file.

## 3.4.5 Using an Imported Dump File

Before you can log into Oracle Configurator Developer and the Oracle SellingPoint application as <impdump>/<impdumppass>, you must execute the REDO_

SEQUENCES procedure, define the data source name for <impdump>, and list it in your spx.ini file.

1. Start SQL*Plus, connect to <impdump>, and type the following at the SQL prompt:

   ```
   SQL> exec CZ_MANAGER.REDO_SEQUENCES ('1');
   ```

2. Set up an ODBC data source name for the <impdump> (see Section 2.8.3, "Create DSNs and DBOwners" on page 2-11).

3. Include the DSN in your spx.ini file. The following are the corresponding lines that must be added to the spx.ini file ("..." indicates existing lines that are omitted here):

   ```
   [DSN]
   ...
   <DSN_for_impdump>
   [MDADSNS]
   ...
   <DSN_for_impdump>
   ...
   [<DSN_for_impdump>]
   DBowner=<impdump>
   ```

# 4

# Data Transfer to the CZ Schema

Oracle runtime configurators use a standard schema for configuration data referred to as the Oracle Configurator schema (CZ tables) in the Oracle Applications database. The Oracle Configurator schema is used to store customer information and all information relative to a configuration model — product data, project structure, configuration rules, and user interface layouts. Customer and product-related data, such as price lists, are generally transferred into the Oracle Configurator schema from data sources external to Oracle Configurator.

Transferring data from Release 11*i* Oracle Applications schemas into the Oracle Configurator schema is accomplished by running ***concurrent programs*** in Oracle Applications.The concurrent programs transfer BOM, Item Master, customer, address, user, and contact data between the Oracle Configurator schema (CZ tables) and tables in Oracle Applications used by other applications that integrate with Oracle Configurator.

Importing data from Release 11.0 or 10.7 Oracle Applications into the Oracle Configurator schema is called ***direct import*** and is accomplished using a set of SQL*Plus scripts. Importing legacy data from non-Oracle Applications or non-Oracle databases is called ***generic import*** and is accomplished through custom programs and existing SQL*Plus scripts.

Regardless of which method you use to make already existing data available to the Oracle Configurator, the Oracle Configurator schema will require occasional refreshes with changes and updates for enterprise-wide consistency. Configuration models in the Oracle runtime configurator maintain all relationships associated with the refreshed data to minimize model maintenance as data are updated.

## 4.1 Overview of Data Transfer or Import

The DBAs need to understand how enterprise or legacy data are used in the Oracle runtime configurator and how the application is refreshed when data changes.

In order to pass data between Oracle Configurator schema and other Oracle Applications tables or external data, the Oracle Configurator schema includes a subschema of integration tables. The tables that stage data coming into the Oracle Configurator schema are the ***import tables***. See Appendix A.2, "List of Import Tables" on page A-1 for additional information about the import tables. The tables that are populated with data from the import tables and used by the Oracle runtime configurator online are called the ***online tables***.

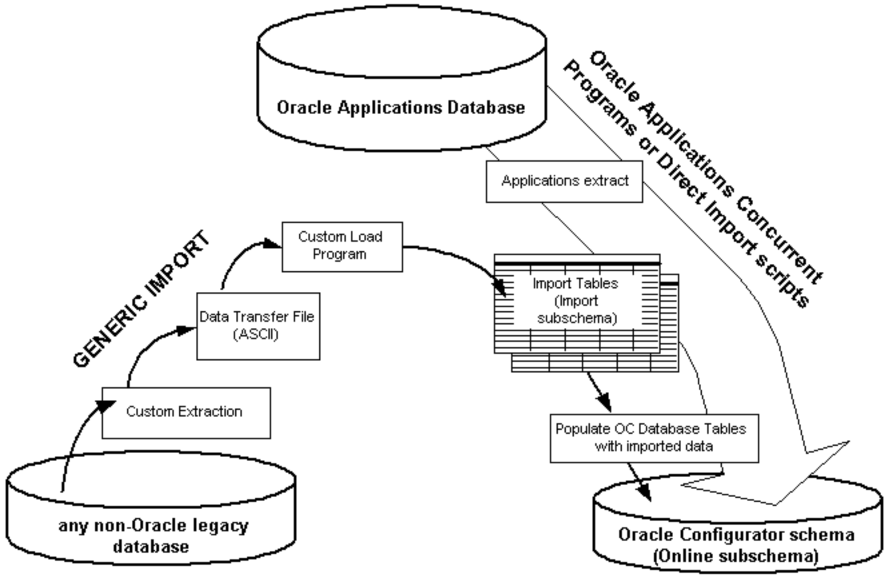
There are several different ways the import tables can be used:

- To transfer specific records or all available records from Oracle Applications (Release 11*i*) tables.
- To import specific records or all available records from Oracle Applications (Release 10.7 or 11.0) tables.
- To import legacy data from non-Oracle database.

It is possible to use a combination of methods on one OC project to achieve the results you need.

Transferring data between the Oracle Configurator schema and other Release 11*i* Oracle Applications tables can be done automatically with concurrent programs. OC SQL*Plus scripts on the Oracle Configurator Developer CD are available for import Releases 10.7 or 11.0 Oracle Applications directly into the Oracle Configurator schema. For importing non-Oracle data, you must create extraction programs, and use available OC SQL*Plus scripts for populating the import tables and then transferring data from those import tables to the Oracle Configurator schema online tables.

*Figure 4–1 Overview of Data Transfer or Import*


Oracle Applications Database
Oracle Applications Concurrent Programs or Direct Import scripts
Applications extract
GENERIC IMPORT
Custom Load Program
Data Transfer File (ASCII)
Import Tables (Import subschema)
Custom Extraction
Populate OC Database Tables with imported data
any non-Oracle legacy database
Oracle Configurator schema (Online subschema)

As you can see in Figure 4–1, both import and data transfer use the import tables in the Oracle Configurator schema to populate the online Oracle Configurator schema tables with the extracted data used by the application. For data not originating in tables that Oracle Applications (Release 11*i*) concurrent programs or available database procedures handle, you must develop a custom extraction. When importing data from a non-Oracle database, the data must be massaged into a compatible format and custom loaded into the import tables (Generic Import in Figure 4–1). Exactly which extracted data are imported depends on the settings in the control tables (CZ_XFR_ tables in the Oracle Configurator schema) or the custom load program, if applicable.

The Oracle Applications concurrent programs or OC SQL*Plus procedures do not provide an automated or scheduled mechanism that clears the import tables.

## 4.1.1 Import Tables

The import tables represent a ‘shadow’ of the online tables in the Oracle Configurator schema. Every online table that receives transferred or imported data is matched by an import table that is equivalent to that table both structurally and relationally. Each import table is named exactly as its online counterpart, but is named with the prefix CZ_IMP_ rather than simply CZ_. For example, the imported data in CZ_IMP_PROPERTIES populate CZ_PROPERTIES. All fields are nullable; no data constraints are applied to any field in an import table.

Each import table contains the same fields that exist in the online table, plus additional fields used specifically for the import process. Import tables consist of fields for:

- Import Control
- Online Data
- Surrogate Keys

#### 4.1.1.1 Import Control Fields

Import control fields contain data that are used only to manage the transfer or generic import process for each record. Import control data are not transferred to the online tables and are not used to resolve key values or anything else. The import control fields are:

***Table 4–1 Import Control Fields***

| Field Name | Type | Description |
|---|---|---|
| RUN_ID | INTEGER | Input field that identifies with which run this record is associated |
| REC_NBR | INTEGER | Input field that is a one-up sequence number uniquely identifying each record *within* a RUN_ID |
| REC_STATUS | VARCHAR(4) | Output field which is the validation status of the record. Null indicates the record status is open. Once this status is set, further processing of the record is suppressed. A REC_STATUS of "OK" indicates that the data in this record now exist in the online database table |
| DISPOSITION | CHAR(1) | Output field that indicates the disposition of the record after an import:<br>I = Insert<br>M = Modify<br>N = No change<br>R = Rejected<br>Null indicates not yet known |

#### 4.1.1.2 Online Data Fields

Online Data Fields exactly match the fields in the corresponding online table and are used to hold the literal data to be put into the online table.

#### 4.1.1.3 Surrogate Key Fields

Surrogate key fields are fields in the import tables that hold the customer-provided "extrinsic" identifications for data to be imported. These include both surrogate primary keys and surrogate foreign keys. A "surrogate foreign key" is a reference to a different table made through that table’s surrogate primary key rather than through the online table’s integer key value.

**Surrogate Primary Key** – as a rule, imported tables contain a single field named ORIG_SYS_REF which is used to hold the external value that uniquely identifies each record. In some cases, however, the online CZ table has a primary key consisting entirely of references to other tables. In this case, the surrogate primary key will actually consist of the foreign surrogate keys that correspond to the native foreign keys in the online table.

**Foreign Surrogate Keys** – one or more fields used to resolve references from one import table to another. These keys are named FSK_`<table>`_`<refno>`_`<fldnum>`, where `<table>` is the name of the referenced table, `<refno>` is the number of the table-to-table reference, and `<fldnum>` is the position of the referenced surrogate-key field in the referenced import table. Note that `<refno>` is required to keep unique names for tables with multiple references to the same table, and generally, the `<fldnum>` is ‘1.’

### 4.1.2 Control Tables (CZ_XFR_)

The import process is controlled by a set of tables (CZ_XFR_ tables) with data records that determine what and how data are transferred or imported. They also determine which import tables are enabled for import.

There are two tables that control the import process at the table and field level; CZ_XFR_TABLES and CZ_XFR_FIELDS. CZ_XFR_TABLES identifies the import table to online table transfer that is to be performed during import.

*Table 4–2 CZ_XFR_TABLES Fields*

| Field Name | Req’d | Type | Description |
|---|---|---|---|
| XFR_GROUP | YES | VARCHAR (20) | Used to name and group sequences of imports, such as "EXTRACT", "IMPORT", "GENERIC". |

*Table 4–2 CZ_XFR_TABLES Fields*

| Field Name | Req’d | Type | Description |
|---|---|---|---|
| ORDER_SEQ | YES | NUMBER | Indicates the order in which this table should be imported. It also serves as an identifier for defining a transfer of data from one import table to one online table. |
| SRC_TABLE | NO | VARCHAR (30) | Identifies the name of the source table in the import subschema. |
| DST_TABLE | NO | VARCHAR (30) | Contains the name of the destination online table |
| DST_SUBSCHEMA | NO | INTEGER | Identifies the subschema in which the destination online tables resides |
| FILTERSYNTAX | NO | VARCHAR (255) | Limits the records imported from the given src_table. |
| PK_USEEXPANSION | NO | CHAR(1) | If "Y", the surrogate primary key is passed through the expansion field user_str03 rather than through the ’natural’ surrogate primary key (for example, name, ref_part_nbr). |
| DISABLED | NO | CHAR(1) | If "1", there is no transfer of data. For XFR_TABLE entries with XFR_GROUP = "EXTRACT", IMP_EXTRACT will not load data into the import tables. When XFR_GROUP = "IMPORT" or "GENERIC", any data in the import table is ignored. |

CZ_XFR_FIELDS identifies transfer rules for fields transferred during the import process. It primarily specifies overrides to normal rules; if a field is transferred for which no entry exists in CZ_XFR_FIELDS, the default is applied.

*Table 4–3 CZ_XFR_FIELDS Fields*

| Field Name | Req’d | Type | Description |
|---|---|---|---|
| XFR_GROUP | YES | VARCHAR (20) | Used to name and group sequences of imports, such as ’EXTRACT’, ’IMPORT’ |

*Table 4–3 CZ_XFR_FIELDS Fields*

| Field Name | Req'd | Type | Description |
|---|---|---|---|
| ORDER_SEQ | YES | NUMBER | With xfr_group, identifies the distinct table import to which this record applies |
| FIELD_ORDER | YES | NUMBER | Distinguishes the field imported |
| SRC_FIELD | NO | VARCHAR 2(40) | Name of the originating field |
| DST_FIELD | NO | VARCHAR 2(40) | Name of the destination field |
| REQUIRED | NO | CHAR(1) | If "1", this field is required by the online table |
| DEFAULTSYNTAX | NO | VARCHAR 2(255) | Not currently supported. Future use is to establish default value when null |
| NOUPDATE | NO | CHAR(1) | If "1", this field is not to be modified when an existing online record is matched |

Each entry in the CZ_XFR_PROJECT_BILLS table identifies a top-level item from a bill of material in Oracle Applications for import into the Oracle Configurator. Every imported bill must be represented in CZ_XFR_PROJECT_BILLS.

*Table 4–4 CZ_XFR_PROJECT_BILLS Fields*

| Field Name | Req'd | Type | Description |
|---|---|---|---|
| ORGANIZATION_ID | YES | NUMBER | Identifies the organization ID that identifies this item in Oracle Applications |
| COMPONENT_ITEM_ID | NO | NUMBER | Provides the item ID that identifies this item in Oracle Applications. |
| DESCRIPTION | NO | VARCHAR2 (255) | Most recent description of this item. |
| LAST_IMPORT_RUN_ID | NO | NUMBER | Identifies the import run in which this bill was last imported. |

*Table 4–4 CZ_XFR_PROJECT_BILLS Fields*

| Field Name | Req’d | Type | Description |
|---|---|---|---|
| LAST_IMPORT_DATE | NO | DATE | Gives the date and time for when this bill was last imported. |
| SOURCE_BILL_DELETED | NO | CHAR(1) | If ’1’, Indicates that this bill has been deleted from Oracle Applications; if ’0’, it is still active. |
| TOP_ITEM_ID | YES | NUMBER | Contains the item ID that identifies this top-level item in this bill. |
| DELETED_FLAG | YES | CHAR(1) | If ’1’, this flag logically deletes (disables) this bill from being imported; if ‘0’, it is still active. |
| EXPLOSION_TYPE | NO (by default) | VARCHAR2 (10) | Instructs the import software as to the type of BOM explosion it should perform on this bill, if automatic explosion is enabled in the CZ_DB_SETTINGS table. |
| BILL_REVISION_DATE | NO | DATE | Describes when the source data for this bill in Oracle Applications were last modified. |

Each entry in the CZ_XFR_PRICE_LISTS table identifies a price list from Oracle Applications that is to be imported into the Oracle Configurator.

*Table 4–5 CZ_XFR_PRICE_LISTS Fields*

| Field Name | Req’d | Type | Descriptions |
|---|---|---|---|
| PRICE_LIST_ID | YES | NUMBER | Identifies the price list in Oracle Applications. |
| IMPORT_ITEM_PRICES | NO | CHAR(1) | If ’1’, indicates to import item prices for this list but do not display them in the Oracle runtime configurator model; if ’2’, import them so they appear in a quote or order; or if ’0’, not to import item prices. |

*Table 4–5    CZ_XFR_PRICE_LISTS Fields*

| Field Name | Req'd | Type | Descriptions |
|---|---|---|---|
| DESCRIPTION | NO | VARCHAR2(240) | Most recent description of this price list from Applications. |
| LAST_IMPORT_RUN_ID | NO | NUMBER | Identifies the import run in which this price list was last imported. |
| LAST_IMPORT_DATE | NO | DATE | Gives the date and time for when this price list was last imported. |
| SOURCE_PRICE_DELETED | NO | CHAR(1) | If '1', indicates that the price list in this record has been deleted from Oracle Applications; if '0', the record is still active. |
| DELETED_FLAG | YES | CHAR(1) | Logically deletes or disables the import of this price list if '1'; '0' if the import is still active. |

## 4.1.3 Data Transfer and Import Setup

Transfer and import setup consists of:

- making sure that data to be transferred or imported are clean, and in the case of BOMs, complete and identified at the desired root
- identifying and modifying what gets transferred or imported. This is accomplished using administrative concurrent programs which use the default values of settings in the CZ_XFR_ control table in the Oracle Configurator schema for transfers from Oracle Applications Release 11*i* databases. It is accomplished by settings with specified values in the CZ_XFR_ control tables for import.

### 4.1.3.1 CZ_DB_SETTINGS for Transfer and Generic Import

The following settings are in effect during the transfer or generic import run:

AUTOCREATE_IMPORTED_USERS
RUN_BILL_EXPLODER
CommitSize
MaximumErrors
DEFAULT_ITEM_TYPE

See Section 3.2, "Oracle Configurator Schema Settings" on page 3-4 for information about these and other CZ_DB_SETTINGS, including their default values.

You can also specify whether you want to apply item prices in the Oracle runtime configurator or in Oracle Order Management (Release 11i) or Oracle Order Entry (Release 10.7 or 11.0). To do this, set the value of OraclePricing in the CZ_DB_SETTINGS table. When OraclePricing='NO' (default), you can specify prices in the Oracle runtime configurator that remain static in the order submitted to Oracle Applications. To enable pricing in the Oracle runtime configurator (OraclePricing='NO'), you must also insert values for the DB_SETTINGS: DefaultPriceGroupID, DiscountID, OrderTypeID, and OrderImportSourceID. For Oracle runtime configurator-based pricing, the value of the DB setting DefaultPriceGroupID must be an empty price list in Oracle Applications.

The DiscountID setting allows manually-applied order-level discounting in Order Management (Release 11i) or Order Entry (Release 10.7 or 11.0) of non-Oracle Applications prices created in the Oracle runtime configurator. To bypass pricing in the Oracle runtime configurator and perform final pricing in Order Management or Order Entry, the OraclePricing flag must be set to 'YES' and you must additionally set the OrderTypeID and OrderImportSourceID. See Section 3.2.4, "CZ_DB_Settings for ORAAPPS_INTEGRATE", on page 3-11 for more information about the OraclePricing setting. See Chapter 6, "Pricing in Oracle Configurator" for more information about how pricing works in Oracle Configurator, Release 11*i*.

The value of the **0** setting in the DB_USER_ROLES section of the CZ_DB_SETTINGS table is assigned 'SPX_USER' role by default to all users defined in the CZ_END_USERS table, including Oracle Applications users transferred with concurrent programs or imported with generic import scripts (GO_IMPORT.sql or RunImport.sql).

## 4.2 Data Transferred From Other Oracle Applications Schemas

Customer and product data may be transferred directly into the Oracle Configurator schema from other Oracle Applications tables (Release 11*i*, 10.7, or 11.0) such as the BOM_EXPLOSIONS table. If it is a direct import from Release 10.7 or 11.0 to 11*i*, use the OC SQL*Plus scripts on the Oracle Configurator Developer CD. If it is a data transfer from one schema to another within the Release 11*i* Oracle Applications database, use the Oracle Applications concurrent programs.

Direct import or data transfer enables you to populate the Oracle Configurator schema with the following data:

- Bills of Material (BOM) structure (models and ATO/PTO structure rules)
- Associated Item Master data
- Customers

- Contacts
- Addresses
- Price List Identifiers (IDs)

> **Note:** List prices can be imported only after items have been imported to populate the Item Master. That means list prices will not be imported in the first import run because there are no records in the Item Master yet. You have to run import a second time to import list prices.

The Oracle Applications tables from which data can be imported or transferred are:

***Table 4–6 Oracle Applications Source and Destination Online Tables***

| Oracle Applications Source Table | --> | Oracle Configurator Schema Destination Online Table(s) |
|---:|---|---|
| BOM_EXPLOSIONS | --> | CZ_PS_NODES, CZ_DEVL_PROJECTS, CZ_INTL_TEXTS |
| MTL_SYSTEM_ITEMS | --> | CZ_ITEM_MASTERS |
| RA_SALESREPS | --> | CZ_END_USERS |
| SO_PRICE_LISTS | --> | CZ_PRICE_GROUPS |
| RA_CUSTOMERS | --> | CZ_CUSTOMERS |
| RA_ADDRESSES | --> | CZ_ADDRESSES |
| RA_SITE_USES | --> | CZ_ADDRESS_USES |
| RA_CONTACTS | --> | CZ_CONTACTS |
| RA_PHONES | --> | CZ_CONTACTS |
| MTL_DESCRIPTIVE_ELEMENTS | --> | CZ_PROPERTIES |
| MTL_ITEM_CATALOG_GROUPS | --> | CZ_ITEM_TYPES |
| MTL_DESCR_ELEMENT_VALUES | --> | CZ_ITEM_PROPERTY_VALUES |

In addition, both the OC SQL*Plus scripts and the concurrent programs use the CZ_XFR_ control tables in the Oracle Configurator schema. The TOP_ITEM_ID and ORGANIZATION_ID for each bill of material to be transferred from the Oracle Applications database are read from the CZ_XFR_PROJECT_BILLS table. The PS_NODE transfer updates the CZ_XFR_PROJECT_BILLS table with the timestamp, ID, and description of the most recent transfer.

CZ_DB_SETTINGS defines which fields from MTL_SYSTEM_ITEMS populates the REF_PART_NBR. Likewise, CZ_XFR_FIELDS defines field-specific controls. For example, NOUPDATE set to "1" for CZ_ITEM_MASTER.DESC_TEXT would inhibit the synchronization of the Item Master description.

Each Bill corresponds to a project record inserted in the CZ_DEVL_PROJECTS table. The concurrent program inserts the root project and hierarchy of each bill in the CZ_PS_NODES table.

The Oracle Configurator schema requires complete BOMs. No partial BOMs or subassemblies can be imported. That means, you must identify the TOP_ITEM_ID for the BOM you want to import from Oracle Applications. The BOM structure is imported at the Product level in the Oracle runtime configurator Model. One BOM is imported per Oracle Configurator Developer Project. The Configurator Developer Project name is defaulted from the BOM model name.

The ORGANIZATION_ID identifies which BOMs are imported. OC uses the ORGANIZATION_ID when exporting quotes to Order Entry or added a configured line item in Order Management. An order is only valid with the ORGANIZATION_IDs that correspond to the ORGANIZATION_IDs on all items of the BOM in Oracle Applications. Parameterizing the BOM import by effective date can be done by manipulating the Oracle Applications database default system date. For instance, setting the system date to a past point in time would cause all BOM up to that date, but not after that date, to be imported.

CZ_INTL_TEXTS contains the text string from the DESCRIPTION field in the BOM_EXPLOSIONS table for each transferred Bill project structure node.

PRICE_LIST_IDS transferred from the Oracle Applications database SO_PRICE_LISTS table go into the CZ_PRICE_GROUPS table in the Oracle Configurator schema.

The OC SQL*Plus scripts and concurrent programs target all or a subset of BOMs exploded in the BOM_EXPLOSIONS table in the Oracle Applications database. Selected BOM Items come from the BOM_BILL-OF-MATERIAL and the BOM_INVENTORY_COMPONENTS tables. The OC SQL*Plus scripts and concurrent programs also target all or a subset of Price List IDs from the SO_PRICE_LISTS table in the Oracle Applications database. The Price List IDs are used by the Oracle runtime configurator when submitting quotes to Oracle Order Entry (Release 10.7 or 11.0) or orders to Oracle Order Management (Release 11*i*).

Discounting can be performed in Oracle Order Entry(Release 10.7 or 11.0) or Oracle Order Management (Release 11*i*). You can also choose not to use Oracle Applications Pricing and perform all pricing and discounting in the Oracle runtime configurator.

Direct import OC SQL*Plus scripts and Oracle Applications concurrent programs automatically transfer all Customer, Contact, Address, and Sales Rep data from the RA_ tables listed in Table 4–6.

If you are transferring data directly from the Oracle Applications database tables, you must run the relevant concurrent programs or OC SQL*Plus scripts, which perform the extraction into the correct format for transfer, load the data into the import tables according to the set up, and populate the online Oracle Configurator schema with transferred data from the import tables. See Section 4.2.2.2, "Run Concurrent Programs to Transfer BOM Data" on page 4-22.

## 4.2.1 Direct Import from Oracle Applications 10.7 or 11.0

The BOM data are in the BILL_OF_MATERIALS and MTL_SYSTEM_ITEMS tables of the Oracle Applications database. All import scripts will extract BOMs, based on TOP_ITEM_ID and ORGANIZATION_ID, from the BOM_EXPLOSIONS table in the Oracle Applications. Some import scripts (e.g., RunImport.sql) will explode BOMS and populate the BOM_EXPLOSIONS table if CZ_DB_SETTINGS indicates the BOM that should be exploded.

You further set up import by indicating in the control tables (CZ_XFR_ tables) what extracted data are loaded into the import tables.

See Appendix C, "OC SQL*Plus Scripts and Procedures" for more details on the scripts and procedures used to set up or configure the import.

### 4.2.1.1 Prepare for Direct Import

The Oracle Configurator schema (CZ) does not contain all the database objects needed to support a data import directly from a Release 10.7 or 11.0 Oracle Applications, so these packages, views, and triggers must first be created in the CZ schema.

1. Connect to the Oracle Configurator schema (CZ) in the Release 11*i* Oracle Applications instance. See Section 1.8.1.1, "Connect to a Database Instance" on page 1-14 for details.

The user with which you connect must have connect and resource privileges to the Oracle Configurator schema so that database objects such as packages and views can be created there.

2. Verify that the Release 11*i* Oracle Configurator schema is 14c. See Section 1.8.1.2, "Verify Oracle Configurator Schema Version" on page 1-14 for details.
3. You are running SQL*Plus in the <OC-scripts> directory connected to the instance <ocsid>. See Section 1.8.1.3, "Run SQL*Plus in the <OC-scripts> Directory" on page 1-15 for instructions on how to do this.
4. Run CZ_PACKAGES.sql, CZ_VIEWS.sql, and CZ_TRIGGERS.sql to insert the database objects in the Oracle Configurator schema that you need for the direct import from Oracle Applications Release 10.7 or 11.0.

   Example:

   ```
   SQL> @CZ_PACKAGES
   SQL> @CZ_VIEWS
   SQL> @CZ_TRIGGERS
   ```

   If AUTOCREATE_IMPORTED_USERS is 'YES', you must have administration privileges when you run these scripts.
5. Run InstAppsIntegrateViaLink.sql to set up the link in the Oracle Configurator schema to an instance of Oracle Applications Release 10.7 or 11.0.

   Example:

   ```
   SQL> @InstAppsIntegrateViaLink cz czpass cz czpass vis115 apps appspass
   appslink vis11 server11 1521
   ```

   where <oc> = <imp> and <ocpass> = <imppass> because the Oracle Configurator schema contains both the online and import tables, and name of database link is any name you care to give the link.
6. Ignore errors generated by InstAppsIntegrateViaLink.sql about grants from / to yourself. Errors about packages such as CZ_XF_MGR not being compiled also don't matter to a successful data import.

#### 4.2.1.2 Run Direct Import

Complete the steps in Section 4.2.1.1, "Prepare for Direct Import". Then continue with the following steps to complete the import process.

7. Edit CZ_XFR tables to manipulate which data are imported. See Section 4.2.1.2.2, "Selective Transfer" on page 4-15.

8. Run LoadAllBills.sql.

   Example:

   ```
   SQL> @LoadAllBills
   ```

   For details about what LoadAllBills.sql does, see Section 4.2.1.2.1, "Load All Bills" and Appendix C.2.5, "Using LoadAllBills.sql".

   To populate the import tables and the Oracle Configurator schema from a single BOM, run the interactive script ImportSingleBill.sql. ImportSingleBill.sql prompts you for the ORGANIZATION_ID and TOP_ITEM_ID of the BOM. If you don't specify EXPLOSION_TYPE, the script assumes the default value "Optional".

9. Run the RunImport.sql to push the imported Oracle Applications Release 10.7 or 11.0 data from the import tables to the online tables in the Oracle Configurator schema.

   Example:

   ```
   SQL> @RunImport
   ```

   For details about RunImport.sql, see Appendix C.2, "Scripts".

#### 4.2.1.2.1 Load All Bills

You can use the script LoadAllBills.sql to build your CZ_XFR_PROJECT_BILLS table with all BOMs from Oracle Applications. You can also control which Bills of Material (BOMs) will be imported by manually editing the Product Control Table (CZ_XFR_PROJECT_BILLS). If you use the script LoadAllBills.sql, you can delete those Bills of Material that you don't want imported.

#### 4.2.1.2.2 Selective Transfer

You can selectively insert BOM and price list data into the control tables (CZ_XFR_) that determine what data are imported when you run an import script such as RunImport.sql. You can selectively insert BOM and price list data in the following ways:

- select specific items from the BOM and price list data to be imported
- load all data and identify specific items for removal

Before running an import script for either of these methods, be sure you have completed the steps in Section 4.2.1.1, "Prepare for Direct Import".

### To Select Specific Items From the BOM and Price List Data to be Imported: 1.

1. Insert BOMs in CZ_XFR_PROJECT_BILLS

   For example (entered all on one line):

   ```
   SQL> insert into CZ_XFR_PROJECT_BILLS (TOP_ITEM_ID, ORGANIZATION_ID,
   EXPLOSION_TYPE, DELETED_FLAG) values (<ASSEMBLY_ITEM_ID>, <ORGANIZATION_
   ID>, <EXPLOSION_TYPE>, <DELETED_FLAG>);
   ```

   When executing this command, <EXPLOSION_TYPE> can be ’OPTIONAL’ (default), ’INCLUDED’, or ’ALL’, and <DELETED_FLAG> can be ’0’ (= not deleted (default)), or ’1’. You will have to know the numeric <ASSEMBLY_ITEM_ID> and <ORGANIZATION_ID> for the BOMs you are importing.

2. Insert SO_PRICE_LIST price list data into CZ_XFR_PRICE_LISTS

   For example (entered all on one line):

   ```
   SQL> insert into CZ_XFR_PRICE_LISTS (PRICE_LIST_ID, IMPORT_ITEM_PRICES,
   DELETED_FLAG) values (<PRICE_LIST_ID>, <IMPORT_ITEM_PRICES>, <DELETED_
   FLAG>);
   ```

   When executing this command, <IMPORT_ITEM_PRICES> can be ’0’ (= not imported (default)), ’1’ (import item prices into Oracle Configurator schema prices tables -- not implemented), and ’2’ (import Oracle Applications item prices so that they can be used in the Oracle runtime configurator). Furthermore, the DELETED_FLAG can be ’0’ (= not deleted (default)) or ’1’. You will have to know the numeric <PRICE_LIST_ID> for the price lists you are importing.

### To Load All Data and Identify Specific Items For Removal:

1. Run LoadAllBills.sql.

   Example:

   ```
   SQL> @LoadAllBills
   ```

   For details about what LoadAllBills.sql does, see (Section C.2.5, "Using LoadAllBills.sql").

2. Edit the control tables to manipulate which data are imported.

   Change the value of <DELETED_FLAG> to indicate that specific BOMS and price lists in the CZ_XFR_PROJECT_BILLS and CZ_XFR_PRICE_LISTS, respectively, should not be imported.

For example (entered all on one line):

```
SQL> update CZ_XFR_PROJECT_BILLS set DELETED_FLAG='1' where ORGANIZATION_
ID=<ORGANIZATION_ID> and TOP_ITEM_ID=<ASSEMBLY_ITEM_ID> and EXPLOSION_
TYPE=<EXPLOSION_TYPE>;
```

In this example, you are marking for deletion (by setting DELETED_FLAG to "1") and thus preventing the import of all BOMs in CZ_XFR_PROJECT_BILLS that have the <ORGANIZATION_ID>, <ASSEMBLY_ITEM_ID>, and <EXPLOSION_TYPE> specified.

3. You can also specify the type of data you want to import by manipulating which tables participate in the extraction and import operations of RunImport.sql by editing CZ_XFR_TABLES (and possibly CZ_XFR_FIELDS).

   In the CZ_XFR_TABLES, the value of DST_TABLE, establishes which destination tables in the Oracle Configurator schema to populate. The value of XFR_GROUP, determines whether the imported table is to participate during the extraction or import process. The value of DISABLED determines whether or not a table participates at all. The value of ORDER_SEQ determine the order in which the participating tables are imported.

   For example (entered all on one line):

```
SQL> update CZ_XFR_TABLE set DISABLED='1' where DST_TABLE='CZ_PS_NODES' and
CZ_XFR_GROUP='EXTRACT';
```

   In this example, you are disabling the extraction of data from the Oracle Applications CZ_IMP_PS_NODE table so that the load of data from CZ_IMP_PS_NODE to CZ_PS_NODES in the Oracle Configurator schema will not be a refresh of changed data when you run import. You are preserving PS_NODE data during an import that otherwise may modify PS_NODE with Oracle Applications changes.

   For additional information about customizing the extraction, see Section 4.2.1.3, "Customize Extraction Views" on page 4-17.

### 4.2.1.3 Customize Extraction Views

Although it is not encouraged for direct import, advanced DBAs may wish to customize the Extraction Views of large data import projects to enhance performance.

To populate the import tables by customizing the data extracted from the Oracle Applications 10.7 or 11.0 database, you can run **CREATE_EXP_VIEWS.sql** to create the Oracle Applications Extraction Views. Extraction Views are used to view the data to be extracted from the Oracle Applications 10.7 or 11.0 database for import into the Oracle Configurator schema. The result of executing these Extraction Views is data in the format required by the Oracle Configurator schema Import and Export Tables. CREATE_EXP_VIEWS.sql uses the system date as the effective date to create the extraction views. If the effective date is to be something other than the system date, this script must be customized to do so.

Table 4–7, "Direct Import Extraction Views" shows which Extraction View to customize for the import of specific data from Oracle Applications tables.

***Table 4–7 Direct Import Extraction Views***

| To modify import for... | Customize... |
|---|---|
| CZ_PS_NODES | CZ_EXV_ITEM_MASTER |
| CZ_ITEM_MASTERS | CZ_EXV_ITEMS |
| CZ_ITEM_TYPES | CZ_EXV_ITEM_TYPES |
| CZ_PROPERTIES | CZ_EXV_ITEM_PROPERTIES |
| CZ_ITEM_PROPERTY_ VALUES | CZ_EXV_ITEM_PROPERTY_VALUES |
| CZ_CUSTOMERS | CZ_EXV_CUSTOMERS |
| CZ_ADDRESSES | CZ_EXV_ADDRESSES |
| CZ_ADDRESS_USES | CZ_EXV_ADDRESS_USES |
| CZ_CONTACTS | CZ_EXV_CONTACTS |
| CZ_PRICE_GROUPS | CZ_EXV_PRICE_LISTS |
| CZ_PRICES | CZ_EXV_PRICE_LIST_LINES |
| CZ_END_USERS | CZ_EXV_END_USERS |

List prices will not be imported in the first import run because there are no records in the Item Master yet. You have to run import a second time to import list prices.

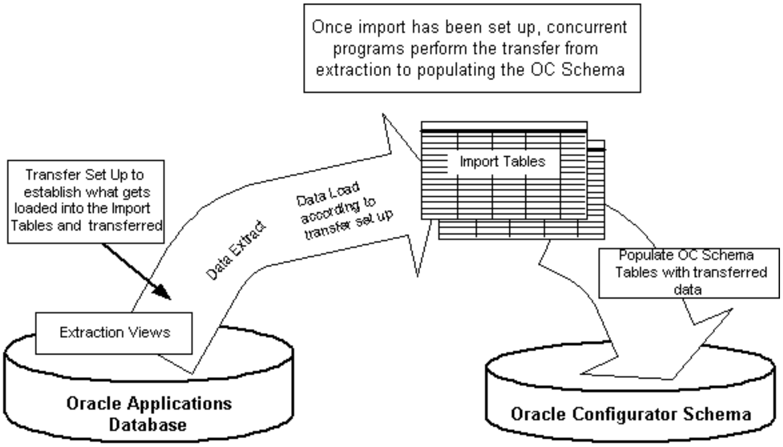
Parameterizing the BOM import by effective date import can be done by manipulating the sysdate.

## 4.2.2 Data Transfer from Release 11*i*

### 4.2.2.1 Setup for Using Concurrent Programs

Oracle Configurator integrates with Bills of Material and Order Management in Oracle Applications Release 11*i*. BOM models and their associated data are transferred from the Oracle Applications database into the Oracle Configurator schema for use in the Oracle Configurator Developer and orders are transferred from the Oracle Configurator schema to the Order Management schema by means of concurrent programs.

***Figure 4–2 Overview of Data Transfer with Concurrent Programs***


Once import has been set up, concurrent programs perform the transfer from extraction to populating the OC Schema
Import Tables
Transfer Set Up to establish what gets loaded into the Import Tables and transferred
Data Load according to transfer set up
Data Extract
Populate OC Schema Tables with transferred data
Extraction Views
Oracle Applications Database
Oracle Configurator Schema

Before executing a concurrent program to transfer data into the Oracle Configurator schema of the Oracle Applications (Release 11*i*) database, you have to select what data will be extracted (identify which BOMs should be transferred) and what tables will be involved in the transfer.

In order to transfer data between the Oracle Applications database and the Oracle Configurator schema, you use a series of concurrent programs provided in Oracle Applications to specify control and other settings for the data transfer. You run the following administrative concurrent programs to establish data transfer controls and settings for data transfer to the Oracle Configurator:

- Set Configurator Import Control

- Show Configurator Import Control
- Set Configurator Developer Settings
- Show Configurator Developer Settings
- Run Configurator Import

When you run the Oracle Configurator administrative concurrent programs, you are prompted to enter the required parameters, if any. Once you complete the entry of these parameters, they are automatically added to the Submit Request form.

**Note:** For the initial installation of Oracle Applications, these administrative concurrent programs have been registered, but not associated or linked with any given responsibility. You can link these concurrent programs to a menu definition and responsibility of your choice in Oracle Applications (i.e., a System Administrator menu). See the *Oracle Applications Administrator's Guide* for more information about defining and linking menus. You must also assign and link the associated responsibility for these concurrent programs to the appropriate Oracle Applications users.

You can log into the Oracle Applications System Administrator responsibility and navigate to Concurrent>Program>Define> and click Open to query Oracle Configurator concurrent programs at any time. The Short Name field value for Oracle Configurator is 'SP%'.

Table 4–8, "Oracle Configurator Administrative Concurrent Programs" describes the Oracle Configurator administrative concurrent programs.

*Table 4–8 Oracle Configurator Administrative Concurrent Programs*

| Configurator Concurrent Program | Description | Parameters |
|---|---|---|
| Set Configurator Import Control | Indicates which table to import or extract data to or from during the next data transfer operation. | Phase Name - enter (or select) the data transfer phase, Extract or Import.<br>Destination Table Name - enter (or choose from the drop-down list) the name of the table where the data is being transferred.<br>Enable - enter Y or N to indicate whether or not the specified table participates in the Import phase of the data transfer. |
| Show Configurator Import Control | Shows whether or not the specified table is enabled or disabled for the specified data transfer phase. | Phase Name - enter (or choose from the drop-down list) the data transfer phase, Extract or Import.<br>Destination Table Name - enter (or choose from the drop-down list) the name of the table being queried. |
| Set Configurator Developer Settings | Sets values for the following settings in the Oracle Configurator schema:<br>Schema Major Version<br>Schema Minor Version<br>Oracle Sequence Incrementer<br>Run Bill Exploder | Section Name - enter (or choose from the drop-down list) the name of the section in the Oracle Configurator schema where the setting resides.<br>Setting - enter (or choose from the drop-down list) the setting_id in the Oracle Configurator schema.<br>Value - enter the value you wish to set the setting_id to. See Table 3–1, "CZ_DB_SETTINGS", on page 3-5 for valid values for these settings.<br>Type - enter (or choose from the drop-down list) the type of setting, for example, integer, varchar, etc.<br>Description - enter a brief description for this value selection. |
| Show Configurator Developer Settings | Shows the value of the specified Oracle Configurator schema setting. | Section Name - enter (or choose from the drop-down list) the name of the section in the Oracle Configurator schema where the setting resides.<br>Setting - enter (or choose from the drop-down list) the setting_id in the Oracle Configurator schema. |
| Run Configurator Import | Transfers data from the Oracle Applications database to the Oracle Configurator schema based on the control table values. These are the values you enter when you run the Set Configurator Import Control concurrent program. | No parameters are required to run this concurrent program. |

#### 4.2.2.2 Run Concurrent Programs to Transfer BOM Data

If you want to build configuration models based on BOM models, you can use a series of concurrent programs provided in Bills of Material to transfer BOM and Item Master data from the Bills of Material schema to the Oracle Configurator schema. The Oracle Configurator uses this data in its development environment, Oracle Configurator Developer, to define a configuration model.

Login to Oracle Applications Manufacturing responsibility. Navigate to **Bills of Material>Other>Configurator** and select one of the following concurrent programs to transfer, disable transfer, or refresh BOM Model data from the Bills of Material schema (Release 11*i*) to the Oracle Configurator schema:

- Populate Configuration Models
- Refresh Configuration Models
- Disable Import of Configuration Model

When you run the Oracle Configurator concurrent programs from these submenus, you are prompted to enter the required parameters, if any. Once you complete the entry of these parameters, they are automatically added to the Submit Request form.

> **Note:** For the initial installation of Oracle Configurator, these concurrent programs have been registered, and associated or linked with the Manufacturing:Bills of Material responsibility. You can link these concurrent programs to another menu definition and responsibility of your choice in Oracle Applications (i.e., a System Administrator menu). See the *Oracle Applications Administrator's Guide* for more information about defining and linking menus. You must also assign and link the associated responsibility for these concurrent programs to the appropriate Oracle Applications users. If you are familiar with these concurrent programs and know the required parameters, you can also run these concurrent programs directly from the Submit Request form.

Table 4–9, "Oracle Configurator Concurrent Programs in Bills of Material" describes the Oracle Configurator concurrent programs found in Oracle Bills of Material.

*Table 4–9 Oracle Configurator Concurrent Programs in Bills of Material*

| Configurator Concurrent Program | Description | Parameters |
|---|---|---|
| Populate Configuration Models | Transfers BOM Model tree structures and associated Item Master data from the BOM schema to the Oracle Configurator schema. This data is then available in Oracle Configurator Developer for developing configuration models. | Organization Code - enter (or choose from the drop-down list) the code defined for the applicable organization.<br>Model Inventory Item From - enter (or choose from the drop-down list) the first Model Inventory Item in the range of items for which you want to transfer data.<br>Model Inventory Item To - enter (or choose from the drop-down list) the last Model Inventory Item in the range of items for which you want to transfer data.<br>**Note:** All Model Inventory Items between and including the first and last specified, are in the data transfer. The range can include multiple Model Inventory Item Types. For example, from ATO800 to CNO500 is a valid range. |
| Refresh Configuration Models | Refreshes data between the BOM schema and the Oracle Configurator schema. Updates Model tree structures and associated Item Master data with new or revised BOM data in Oracle Configurator Developer. | No parameters are required to run this concurrent program. |
| Disable Import of Configuration Model | Disables transfer of BOM model data to the Oracle Configurator schema for the specified Model Inventory Item. | Organization Code - enter (or choose from the drop-down list) the code defined for the applicable organization.<br>Model Inventory Item - enter (or choose from the drop-down list) the Model Inventory Item for which you want to disable data transfer to the Oracle Configurator schema. |

You can directly transfer data for a single table or for many tables at once. In order to transfer data, import parameters must be set (see Section 4.2.2.1, "Setup for Using Concurrent Programs" on page 4-19).

## 4.3 Generic Import

For legacy data residing in non-Oracle databases, Oracle Configurator provides a generic import mechanism and database procedures. The generic import facility utilizes the import tables to load data into the online Oracle Configurator schema.

The following tables can be populated through generic import:

CZ_ADDRESSES
CZ_ADDRESS_USES
CZ_CONTACTS
CZ_CUSTOMERS
CZ_CUSTOMER_END_USERS
CZ_DEVL_PROJECTS
CZ_END_USERS
CZ_END_USER_GROUPS
CZ_INTL_TEXTS
CZ_ITEM_MASTERS
CZ_ITEM_PROPERTY_VALUES
CZ_ITEM_TYPES
CZ_ITEM_TYPE_PROPERTIES
CZ_PRICES
CZ_PRICE_GROUPS
CZ_PROPERTIES
CZ_PS_NODES
CZ_USER_GROUPS

If you are importing legacy data from non-Oracle databases, you must:

1. Identify and cleanse data for import.
2. Create and run custom extraction programs for the data you want to import and either:
   a. generate an ASCII file in the data transfer (DAT) format the import tables require, then create and run load programs that load the transfer data into the import tables. See Section 4.3.1, "Setup for Generic Import" on page 4-24.

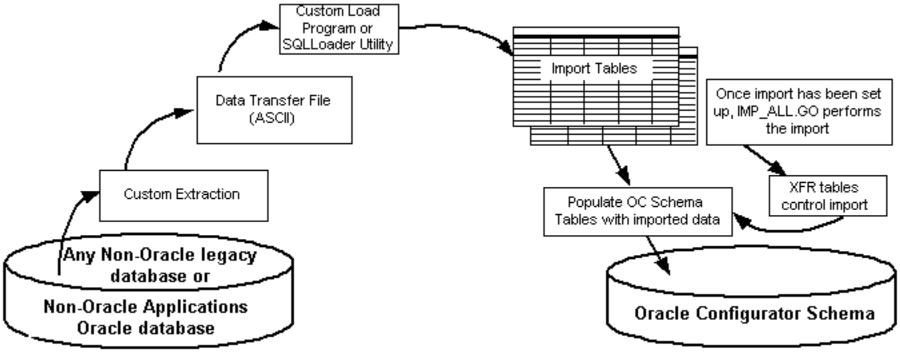
      OR

   b. load the data generated by the custom extraction programs directly into the import tables.
3. Run the import script, RunGenImport.sql, which populates the Oracle Configurator schema tables with imported data from the import tables. See Section 4.3.2, "Run Generic Import" on page 4-27.

### 4.3.1 Setup for Generic Import

If you are importing legacy data or Oracle Applications data not accessible through transfer using concurrent programs, you must perform a generic import. Generic import requires writing queries to extract the product, customer/contact, and pricing data in a format that satisfies the import tables in the Oracle Configurator schema. You must then develop custom load programs to populate the import tables in OC with that extracted data. The import procedure provided by Oracle Configurator then populates the Oracle Configurator schema from the import tables.

***Figure 4–3 Overview of Generic Import***


Custom Load Program or SQLLoader Utility
Import Tables
Data Transfer File (ASCII)
Once import has been set up, IMP_ALL.GO performs the import
Custom Extraction
XFR tables control import
Populate OC Schema Tables with imported data
Any Non-Oracle legacy database or
Non-Oracle Applications Oracle database
Oracle Configurator Schema

In order to know what data to extract for populating the import tables, you need to know what fields are available in the import tables for data population. See Appendix A.4, "Import Tables", for detailed information about all Import Table fields and Appendix A.3, "Dependencies Among Import Tables", on page A-2 for information about the dependencies among the Import Tables.

### 4.3.1.1 Required ASCII File Format for Generic Import

For a generic import, you must define the extraction from your legacy database, create the data transfer files containing the extracted data, and the load program that loads the import tables with that data. The format of the data transfer files you load into the import tables must match their target import tables exactly, field for field.

The data transfer files include all data in text (ASCII) format, with fields separated by delimiters such as a vertical bar (“ | ”).

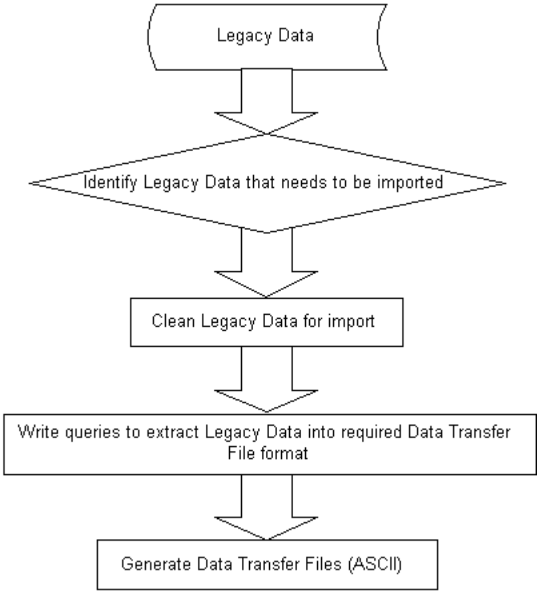
The following example imports item types. Item type populates the third column of the 21-column import table CZ_IMP_ITEM_MASTER.

```
||Memory Board||||||||||||||||||
||Dual CPU||||||||||||||||||
||Country|||||||||||||||||||
||System Console|||||||||||||||||||
||Server Console|||||||||||||||||||
||Disk Drive|||||||||||||||||||
||Storage Media|||||||||||||||||||
||Server Size|||||||||||||||||||
||Power Supply|||||||||||||||||||
||Matrix Printer|||||||||||||||||||
||SCSI Disk Drive|||||||||||||||||||
||Cache Memory|||||||||||||||||||
||Disk Array Model|||||||||||||||||||
||SCSI Type|||||||||||||||||||
||SCSI Cable|||||||||||||||||||
||SCSI Chaining|||||||||||||||||||
||SCSI Cabling Configuration|||||||||||||||||||
||Server Type|||||||||||||||||||
||System Size|||||||||||||||||||
```

#### 4.3.1.2 Generic Import Setup Process

The process for setting up the extraction portion of a generic import is as follows:

*Figure 4–4 Generic Import Setup Process*


Legacy Data
Identify Legacy Data that needs to be imported
Clean Legacy Data for import
Write queries to extract Legacy Data into required Data Transfer File format
Generate Data Transfer Files (ASCII)

Once you have the Data Transfer Files required by the Oracle Configurator schema import tables, you must also create the load programs that populate those import tables with the data contained in the Data Transfer Files.

## 4.3.2 Run Generic Import

If you are generically importing data from another Oracle database (non-Oracle Applications) or a non-Oracle database, you are responsible for implementing a custom extraction of your legacy data. You can use the data transfer database procedure for populating the import tables.

You must have established namespaces or packages for data import and the integration tables in the Oracle Configurator schema.

Before running the generic import script, be sure you have completed the following:

1. You are running SQL*Plus in the <OC-scripts> directory connected to the instance <ocsid>. See Section 1.8.1.3, "Run SQL*Plus in the <OC-scripts> Directory" on page 1-15 for instructions on how to do this.
2. Be sure your legacy data are available in the ASCII file format required by the import tables. See Appendix 4.3.1.1, "Required ASCII File Format for Generic Import", on page 4-25.
3. You have extracted legacy data and loaded them into the Oracle Configurator schema import tables. See Section 4.3.1, "Setup for Generic Import", on page 4-24.
4. Set the 'DISABLED' setting in the CZ_XFR_TABLES to '0' for any table into which you are importing data. Disable all import tables that you will not be importing data into by setting the 'DISABLED' setting to '1'. Minimally, the following default tables are used for generic import and should have the 'DISABLED' setting set to '0':
   - CZ_ITEM_MASTERS
   - CZ_ITEM_TYPES
   - CZ_ITEM_TYPE_PROPERTIES
   - CZ_ITEM_PROPERTY_VALUES
   - CZ_ITEM_PARENTS (reserved for future use)
   - CZ_PROPERTIES
5. Connect to the instance <ocsid> as the online user.

   Example:

   ```
   SQL> conn <imp> <imppass> @<ocdbhost>
   ```
6. Import the data to the Oracle Configurator schema by running the interactive script RunGenImport.sql.

   Example:

   ```
   SQL> @RunGenImport.sql
   ```

   RunGenImport.sql prompts you for the RUN_ID. Either specify a particular RUN_ID or press Enter to have the RUN_ID automatically generated. The RUN_ID displays on the screen for future use, if necessary. The import procedure populates the Oracle Configurator schema with data from the import tables. The source of the import is the data extracted from the legacy database by means of custom extraction and load programs. The destination of the imported data is the Oracle Configurator schema.

## 4.3.3 Re-Run Generic Import

You may need to re-run a generic import because:

- a situation may arise that requires you to re-run a specific import session for all the same data.
- some of the records have failed due to incorrect data in some columns.

### 4.3.3.1 Re-Run an Entire Import Session

You can repeat an import session for all the same data that were previously imported. To do so, run RunGenImport.sql and specify the RUN_ID for the previous session when prompted.

### 4.3.3.2 Re-Run Specific Records

You can repeat an import session for specific records. If some of the imported records fail due to incorrect data in some columns:

1. Correct the column data and nullify or supply a special value in the RUN_ID column for those records.
2. Run `RunGenImport.sql` and press Enter at the prompt for records where RUN_ID is null or specify the special value entered.

# 4.4 Verify Data Transfer or Import

Once you have transferred or imported your data into the Oracle Configurator schema, start Oracle Configurator Developer to view the Item Master and Model Structure containing the transferred or imported data. All Items transferred or imported into the Oracle Configurator schema are displayed in the Oracle Configurator Developer Item Master. Standard Items located in the Oracle Configurator schema Item Master are read/write. Any modifications to the Oracle Configurator schema Item Master (except Property additions and item type assignments) are overwritten with values from the Oracle Applications database when data are refreshed with another transfer or import unless the CZ_XFR_ table record for that item type is flagged ‘DELETED’ to control the refresh.

## 4.5 Refresh and Update Imported Data

Oracle recommends that changes to legacy data be limited during construction of an Oracle runtime configurator to avoid potential problems introduced by interim data transfers and imports. Oracle suggests that testing be completed before you import changes from Oracle Applications or legacy data, so that the test cases are up-to-date with the application that has been constructed. Your project's full application testing should include importing changed data and upgrading the application to match current enterprise or legacy data before deploying the Oracle runtime configurator. Test cases may have to be updated to match the changes.

Although updating imported data in the Oracle Configurator schema randomly during a development phase is not recommended, Oracle recognizes that project managers may need to synchronize with Oracle Applications data frequently. Such refreshes and updates may require careful control of what data get imported and/or corrections to the definitions of the configuration model in the Oracle runtime configurator. In many cases, a refresh causes deletion of previously imported data.

Once an Oracle runtime configurator has been deployed, customer data are stored in the Oracle Configurator schema directly through networked use. During deployment, further imports are done to refresh the Oracle Configurator schema as Oracle Applications or legacy data change. The procedures that perform the import prevent customer-specific groups of fields in the Oracle Configurator schema from being altered or nulled out even when other fields in the row are replaced during an import session.

For additional information about maintaining data currency in your Oracle Configurator schema, see Section 3.3, "Oracle Configurator Schema Maintenance" on page 3-16.

# 5

# Data Transfer from the CZ Schema

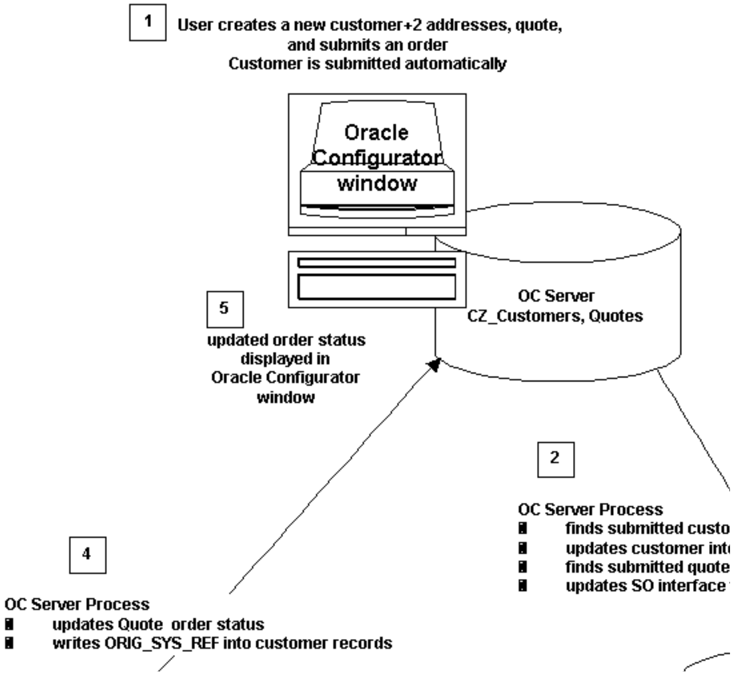
Oracle runtime configurators use a standard schema for configuration data referred to as the Oracle Configurator schema(CZ tables) in the Oracle Applications database. The Oracle Configurator schema is used to store customer information and all information relative to the configuration model — product data, model structure, configuration rules, and user interface layouts. Customer and product configuration order data can be transferred or exported from the Oracle Configurator schema to the Oracle Applications database for use by Oracle Order Entry and Oracle Receivables (Release 10.7 or 11.0) and Oracle Order Management (Release 11*i*).

## 5.1 Overview of Data Transfer from the CZ Schema

Data transfer or export to the Oracle Applications Database involves preparing the Oracle Configurator and the Oracle Applications environments to support data and status transmission. The Oracle runtime configurator manipulates and modifies transferred or imported BOM data into valid product configurations. The Oracle runtime configurator end user creates a product configuration quote for a specified customer and submits the quote as an order. The order and customer data are transferred or exported to the Oracle Applications database, so that the orders created in the Oracle runtime configurator are available to Oracle Order Management (Release 11*i*) or Oracle Order Entry (Release 10.7 or 11.0) . When an order is booked in Oracle Order Management or Oracle Order Entry, the order status is transferred back through export tables of the Oracle Configurator schema to the Oracle runtime configurator.

New Customer data can also be transferred or exported to the Oracle Applications Database.

***Figure 5–1 Overview of Data Transfer from the CZ Schema***



## 5.2 Data Transfers from the CZ Schema

Transferred or exported configuration and order data used in Oracle Order Management (Release 11*i*) or Oracle Order Entry (Release 10.7 or 11.0) refer to the data that are transferred from the Oracle Configurator schema. New customer data created by the Oracle runtime configurator are transferred to the Oracle Applications database for use by Oracle Receivables.

For information about what data are imported into the Oracle Configurator schema, see Section 4.2, "Data Transferred From Other Oracle Applications Schemas", on page 4-10 and Section 4.3, "Generic Import", on page 4-23.

Order and customer data are first transferred from the CZ schema to Oracle Applications interface tables. Oracle Applications concurrent programs then extract the appropriate data columns from the interface tables and place them in the appropriate Oracle Applications table.

The data transferred to Oracle Applications (Order Entry and Receivables for Release 10.7 or 11.0, Order Management for Release 11*i*) are:

- Order Headers
- Order Lines

Appendix B, "Export Tables" presents a mapping of the source and destination for data transferred from the Oracle Configurator schema tables to the Oracle Applications database so that orders and new customer information submitted by the Oracle runtime configurator are complete in Oracle Applications.

## 5.3 Prepare for Transfers from the CZ Schema

An Oracle runtime configurator submits orders which pass enough parameters to Oracle Applications database for entering orders, but *not* for booking orders. Orders are booked in Order Management (Release 11*i*) or Order Entry (Release 10.7 or 11.0). Parameters that should be defaulted in a Standard Value Rule Set in Oracle Applications for easy downstream booking of orders submitted from the Oracle runtime configurator are:

- Invoicing Rule
- Accounting Rule
- Payment Terms
- Sales Channel

Additionally, if you use the Oracle SellingPoint application for testing or a mobile deployment, the `Organization ID` field in the **Tools** > **Administration** > **Users... > General** tab must be filled in with the end user's organization ID so that orders can be submitted to Order Management (Release 11*i*) or Order Entry (Release 10.7 or 11.0). This end user must have write permissions on the Oracle Applications database interface tables. The value specified in the `Organization ID` field corresponds to the value of ORGANIZATION_ID in the Oracle Applications database.

Preparation for transfers from the CZ schema to Order Management (Release 11*i*) or Order Entry (Release 10.7 or 11.0) includes:

- completing a successful transfer or import of data to the Oracle Configurator schema. See Section 4.2.2.2, "Run Concurrent Programs to Transfer BOM Data" on page 4-22 or Section 4.3.2, "Run Generic Import" on page 4-27.

- setting order and customer parameters in the CZ_DB_SETTINGS table in the Oracle Configurator schema. See Section 5.3.1, "CZ_DB_Settings for Data Transfers from the CZ Schema", on page 5-4.
- ensuring that currency is set to only one currency throughout the configuration, such as US currency (dollar) in the CZ_PRICE_GROUPS table (this is typically done during the generic import process or entered when running the administrative concurrent programs).
- ensuring that there are data to transfer to Oracle Applications. Oracle runtime configurator end users must have submitted orders or customer information to be transferred to Oracle Applications (order_request_flag=S in CZ_QUOTE_HDRS and/or exported_flag=S in CZ_CUSTOMERS).
- if transferring to Oracle Applications, Release 11*i*, optionally creating and scheduling the export and update processes as database jobs in the Oracle Configurator schema.
- if transferring to Oracle Applications, Release 10.7 or 11.0, optionally registering and scheduling the export and update processes as concurrent programs in the instance where both the Oracle Applications Database and the Oracle Configurator schema are running.

### 5.3.1 CZ_DB_Settings for Data Transfers from the CZ Schema

When transferring data to Oracle Applications, Release 10.7 or 11.0, you can adjust the way the transfers and exports are executed with settings in the CZ_DB_SETTINGS table in the Oracle Configurator schema. For example, you can export just order information, order and customer information, or both. When transferring data to Oracle Applications, Release 11*i*, you enter these parameters when running the export concurrent processes. The following settings determine which data are exported:

- CustomerExportEnabled
- OrderEntry
- OrderImportSourceID
- OrderTypeID
- OraclePricing

For more information about these and other DB_SETTINGS parameters, see Section 3.2, "Oracle Configurator Schema Settings", on page 3-4.

## 5.4 Transfer Data from the CZ Schema

If you want to build configurations in the Oracle runtime configurator and transfer the data to Oracle Order Management (Release 11*i*,) or Oracle Order Entry (Release 10.7 or 11.0), you use a series of concurrent programs provided in Oracle Applications. Order Management and Order Entry use this data in the order processing environment.

Submit the **Order Import Request** concurrent program from the Oracle Applications **Order Management > Orders, Returns > Import Orders>** menu to transfer order data from the Oracle Configurator schema to Oracle Order Management.

When you run this concurrent program, you are prompted to determine whether or not you want validation only on the TRANSFERRED orders. Enter "Yes" or "No" or select the desired option from the drop down list. You must now enter Oracle Configurator (select Oracle Configurator from the drop down list) as the source of the orders you want to import into Oracle Order Management (Release 11*i*,) or Oracle Order Entry (Release 10.7 or 11.0). Once you complete the entry of these parameters, they are automatically added to the Submit Request form.

**Note:** For the initial installation of Oracle Configurator, these concurrent programs have been registered, but not associated or linked with any given responsibility. You can link these concurrent programs to a menu definition and responsibility of your choice in Oracle Applications. (i.e., a System Administrator menu.) See the *Oracle Applications Administrator's Guide* for more information about defining and linking menus. You must also assign and link the associated responsibility for these concurrent programs to the appropriate Oracle Applications users.

If you are familiar with these concurrent programs and know the required parameters, you can also run these concurrent programs directly from the Submit Request form.

Table 5–1, "Concurrent Programs for Transferring Data From CZ Schema" describes the Oracle Configurator administrative concurrent programs:

*Table 5–1 Concurrent Programs for Transferring Data From CZ Schema*

| Configurator Concurrent Program | Description | Parameters |
|---|---|---|
| Order Import Request | Transfers order data from the Oracle Configurator schema to the Oracle Order Management (Release 11*i*) or Order Entry (Release 10.7 or 11.0) schema. This data is then available for processing orders. | Validation Only? Indicates whether or not you want to just validate an order. Valid values are "Yes" or "No".<br>Order Source indicates where the order(s) are to be imported from. |
| Order Export Update Status Process | Retrieves the status of a previously submitted order from the Order Management (Release 11*i*) or Order Entry (Release 10.7 or 11.0) schema and returns it to the Oracle Configurator schema so that the order status is updated in the Oracle runtime configurator. | No parameters are required to run this concurrent program. |

## 5.4.1 Transfer Process

Once Oracle runtime configurator end users have created configurations and submitted orders, the data must be transferred to Oracle Order Management (Release 11*i*) or Order Entry (Release 10.7 or 11.0). Order transfer may occur for a single order or for many orders at once. For information on transferring a single order at a time, see Section 5.4.2, "Transfer a Single Order", on page 5-8.

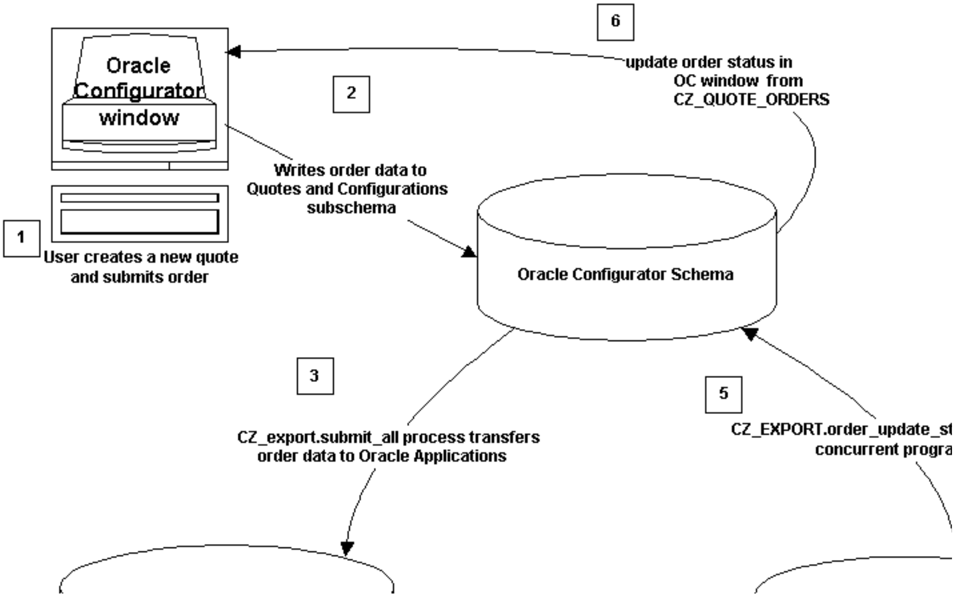
When you submit an order in the Oracle runtime configurator, the data are stored in the Quotes and Configurations subschema of the Oracle Configurator schema, including a record for the order in the CZ_QUOTE_HDRS table. The concurrent programs transfer the order data from the Quotes and Configurations subschema of the Oracle Configurator schema to the Oracle Applications database interface tables (SO_HEADERS_INTERFACE_ALL, etc.). See Appendix B.1, "CZ Schema Source and Oracle Applications Destination Tables", on page B-1 for the exact field to field mapping. See Section 5.3, "Prepare for Transfers from the CZ Schema", on page 5-3 for information about creating database programs to run the export script automatically.

The Order Import Request concurrent program transfers the order and customer data from the Oracle Applications database interface tables into Oracle Order Management (Release 11*i*) or Order Entry and Oracle Receivables (Release 10.7 or 11.0). This concurrent program can be set up to run automatically on a schedule. For information about scheduling current programs, See the *Oracle Applications Administrator's Guide.*

Once an order has been processed, its status must be updated in the Oracle runtime configurator. The Order Export Update Status Process concurrent program transfers status data from SO_HEADERS and other tables back to CZ_QUOTE_ORDERS and other tables in the Quotes and Configurations subschema of the Oracle Configurator schema. The concurrent program updates status in the status field for an order in the Oracle runtime configurator, and updates the customer information.

If an existing order (i.e., Ov1) in the Oracle runtime configurator is revised (Ov2) and resubmitted, Oracle Configurator schema export tables pass a new order (Ov2) to the Oracle Applications database interface table SO_HEADERS_INTERFACE_ ALL flagged for 'insert', and mark the old order (Ov1) 'deleted'. Import from the interface tables to Order Management or Order Entry (i.e., SO_HEADERS) updates the status of the original order (Ov1) to 'cancelled'. When the status update transfers the order data back to the Oracle Configurator schema, the first order's status (Ov1) in the Oracle runtime configurator is 'cancelled' and the second order's status (Ov2) is active (e.g., 'entered'). The changed status of the old order will not appear in the Oracle runtime configurator unless every step in the transfer process has succeeded. See Section 5.5, "Verify Transfer from CZ Schema" for additional information on tracking an order through transfer from the CZ schema.

***Figure 5–2 Overview of Transfer Process from CZ Schema***



### 5.4.2 Transfer a Single Order

You might transfer individual orders manually for testing. These instructions are valid for transferring to Oracle Order Management (Release 11*i*) or Oracle Order Entry ((Release 10.7 or 11.0) whether the Oracle Configurator schema and Oracle Applications database are in the same instance or not.

1. You have set up your Oracle Configurator schema to prepare for transferring orders from the Oracle runtime configurator (see Section 5.3, "Prepare for Transfers from the CZ Schema").

2. Create a configuration in the Oracle runtime configurator and submit it as an order.

   Make a notation of the order number (<ordernumber>) that is assigned to the new quote in the Oracle runtime configurator if you want to transfer only that one order.

3. Start SQL*Plus, and connect to the Oracle Configurator schema (e.g., <oc>) from which you will transfer the data to Oracle Applications.

4. To transfer one order at a time, run the CZ_EXPORT.submit_for_quote procedure.

   Example:

   ```
   SQL> set serveroutput on;
   SQL> exec cz_export.submit_for_quote<ordernumber>;
   SQL> commit;
   ```

   where <ordernumber> is the number you noted above.

   Leave SQL*Plus in <OC-scripts> directory connected to <ocsid> running.

5. To transfer all orders in the Quotes and Configurations subschema of the Oracle Configurator schema, run the CZ_EXPORT.submit_all procedure.

   Example:

   ```
   SQL> set serveroutput on;
   SQL> exec cz_export.submit_all;
   SQL> commit;
   ```

   Leave SQL*Plus in <OC-scripts> directory connected to <ocsid> running.

6. Start Oracle Applications.
7. Import into Oracle Order Management or Order Entry:
   a. Select Order Management (or Order Entry), <your global organization>.
   b. Click OK.
   c. From the right-hand side of the Navigator window, select Requests: Order Import Request.
   d. Click Open.
   e. In the Parameters window, click on the icon with the down arrow in the upper-left corner.
   f. From the Order Source window, select the source from which your orders were submitted (i.e. Oracle runtime configurator).
   g. Click OK.
   h. From the Import Orders window, click Submit Requests.
   i. The Decision window displays the request submitted and the request ID. (note this ID, if desired). One request represents all the orders being transferred to Oracle Order Management (or Order Entry, 10.7 or 11.0).
   j. Submit another request? Y or N.
   k. To update the status of the order in the Oracle Configurator schema, return to the SQL*Plus window and run the CZ_EXPORT.order_status_update_ process procedure.

   Example:

   ```
   SQL> exec cz_export.order_status_update_process;
   SQL> commit;
   ```

   In the Oracle runtime configurator, select Order Status from the left pane. The updated order status is displayed for each order.

### 5.4.3 Transfer Customers Only to Oracle Order Entry (R10.7 or 11.0)

You might transfer new customer data to Oracle Order Entry (R10.7 or 11.0). (New customer data cannot be transferred to Oracle Order Management, Release 11*i*). These instructions are valid whether the Oracle Configurator schema and Oracle Applications database are in the same instance or not.

Customer transfer may occur only for new customers created in the Oracle runtime configurator. Customer information imported to the Oracle Configurator schema from Oracle Applications is used for building configurations and orders, but cannot be modified in the Oracle runtime configurator and transferred back to Oracle Applications.

Data can be transferred for a single new customer or for many new customers at once.

1. You have set up your Oracle Configurator schema to prepare for transferring from the Oracle runtime configurator (see Section 5.3, "Prepare for Transfers from the CZ Schema").
2. Create a customer in the Oracle runtime configurator and submit it while logged on as <apps> in the Oracle runtime configurator.

   Make a note of the customer identification number (<customer_id>) that is assigned to the new customer you created in the Oracle runtime configurator, if you plan to transfer just that one customer manually to Oracle Applications.
3. Start SQL*Plus, and connect to the Oracle Configurator schema (e.g., <oc>) from which you will transfer data to Oracle Applications. See Section 1.8.1.1, "Connect to a Database Instance", on page 1-14 for details.
4. Confirm that customer information has been submitted by typing the following command:

   ```
   SQL> select customer_id, exported_flag, orig_sys_ref from cz_customers;
   ```

   A list of each customer in the CZ_CUSTOMERS table displays with the value of the CUSTOMER_ID (<customer_id>), EXPORTED_FLAG, and ORIG_SYS_REF fields. If the EXPORTED_FLAG value is "S", the customer information has been submitted and is ready for transfer to Oracle Receivables.
5. To display possible errors when the export script runs, type the following command.

   ```
   SQL> set serveroutput on;
   ```
6. To transfer one customer at a time, run the CZ_EXPORT.customer_export procedure.

   Example:

   ```
   SQL> exec cz_export.customer_export <customer_id>;
   SQL> commit;
   ```

where <customer_id> is the customer identification number you noted in step 2, above.

Leave SQL*Plus in <OC-scripts> directory connected to <ocsid> running.

7. To export all the customers that have been submitted, run the CZ_EXPORT.all_customers_export procedure.

   Example:

   ```
   SQL> exec cz_export.all_customers_export;
   SQL> commit;
   ```

   When you transfer all customers, output shows all customers, but only customers that meet the selection criteria of the procedure are transferred. Likewise, customers that failed to transfer previously will also be transferred if they meet the selection criteria of the procedure.

   Leave SQL*Plus in <OC-scripts> directory connected to <ocsid> running.

8. Start Oracle Applications.
9. Import into Oracle Receivables:
   a. Select Receivables, <your global organization>.
   b. Click OK.
   c. From the Navigator window, select Customers from the left panel.
   d. Click Open.
   e. Double-click on Run Customer Interface in the right panel. The Oracle Applications toolbar and Navigator windows minimize and the Run Customer Interface window displays.
   f. Verify the appropriate language.
   g. Click Submit Request.
   h. The Decision dialog displays the request submitted and the request ID. (note this ID, if desired). One request represents all the orders being transferred to Oracle Order Entry.
   i. Submit another request? Y or N.

10. Verify completion of the export request.

    a. From the Oracle Applications toolbar select Help --> View My Requests. The Find Requests dialog displays.

    b. Select All My Requests.

    c. Click Find. The Requests dialog displays the request_ID, name of the request, parent request (if any), request phase, status, and parameters.

11. To update the status of the customer in the Oracle Configurator schema, return to the SQL*Plus window and run the CZ_EXPORT.customer_update_process procedure.

    Example:

    ```
    SQL> exec cz_export.customer_update_process;
    SQL> commit;
    ```

    In the Oracle runtime configurator, select Customer from the left pane. The updated customer status is displayed for each customer.

## 5.5 Verify Transfer from CZ Schema

Verify a successful transfer by viewing the order that was transferred from the Oracle Configurator schema to Oracle Order Management (Release 11*i*) or Oracle Order Entry (Release 10.7 or 11.0).

To verify that export is working in a client/server environment, submit an order in the Oracle runtime configurator, and check order submission in Order Entry. The Oracle runtime configurator end user submitting the order must have write permissions in the Oracle Applications database.

When an order or new customer is submitted from the Oracle runtime configurator, it is only flagged for submission in the Oracle Configurator schema. The following steps need to occur before an order or new customer is created in Oracle Applications and the Oracle runtime configurator is aware of it:

1. Oracle runtime configurator order or customer submission process runs. This process inserts data into the Oracle Configurator schema interface tables.

2. The order or customer transfer process must run successfully. The transfer process populates the Oracle Applications database interface tables with the order and customer data for extraction to source tables (see Section 5.4.1, "Transfer Process", on page 5-6).

3. The order or customer update process must run successfully. This propagates changes to the order or customers made in Order Management (Release 11*i*), Order Entry and Receivables (10.7 or 11.0) back to the Oracle Configurator schema so that the status of the order and customer information is updated in the Oracle runtime configurator.

For information about resubmitting a new version of an already submitted order, see Section 5.4.1, "Transfer Process".

You can track the request that the Oracle runtime configurator submits and that the Oracle Applications concurrent processes read from the interface tables.

- There should be a record for the submitted order in CZ_QUOTE_HDRS
- SO_HEADERS_INTERFACE_ALL has a record flagged ’insert’ for the order being submitted until the concurrent process importing that order into Order Management or Order Entry succeeds. If there is a previous version of the transferred order (same order number, previous version), SO_HEADERS_INTERFACE_ALL contains a record flagged ’delete’ for the previous version until the concurrent process importing that order into Order Management or Order Entry succeeds.
- If the transfer process into Order Management or Order Entry succeeded, the concurrent process deletes the record(s) from the SO_HEADERS_INTERFACE_ALL table. If the transfer process failed, the record for the order in SO_HEADERS_INTERFACE_ALL is updated with an error flag.
- If the status of an order is not being updated in the Oracle runtime configurator, no record for a transferred and updated order is in the CZ_QUOTE_ORDERS table and the CZ_EXPORT.order_status_update_process has not succeeded.

# 6

# Pricing in Oracle Configurator

How Oracle Configurator handles pricing is entirely dependent upon the type of Oracle runtime configurator you choose to use. An Oracle Configurator window is designed to be called from a variety of different applications and requires an interface between the Oracle Configurator window and the calling application’s pricing mechanism. In the Oracle SellingPoint application, pricing depends on whether or not you are running a standalone application or are integrated with Oracle Applications Order Entry (10.7 or 11.0) or Order Management (11*i*). It also depends on whether or not you choose to use Oracle Pricing and the availability of price lists.

## 6.1 Pricing in an Oracle Configurator Window

An Oracle Configurator window is designed to be called from a variety of different applications:

- Order Management
- TeleSales
- Sales Online
- Order Capture
- iStore
- non-Oracle Applications, such as a custom webstore.

The Oracle Configurator window presents list prices for all selectable options, selling prices (e.g., discounts) for all selected options, and a total price for the configuration as a whole.

When the calling application is an Oracle Applications product such as Order Management or iStore, pricing comes from Oracle Pricing (QP). The QP interface is highly configurable. Depending on how it is configured, it may be necessary that appropriate data records are defined in the calling application in order to determine pricing parameters. It is not possible for the Oracle Configurator window to call the QP engine directly to obtain the same pricing that would be obtained by the calling application. Likewise, when the calling application is not an Oracle Applications product, the Oracle Configurator window cannot know how to determine prices at all.

Therefore, the interface between the Oracle Configurator window and the calling application’s pricing mechanism is an interface package, written in PL/SQL as part of the calling application, and passed to the Oracle Configurator window at initialization.

The Oracle Configurator window is displayed when the user clicks the Configure button in the calling application. A calling application to the Oracle Configurator window provides an interface PL/SQL package that interacts between the Oracle Configurator window and the calling application's pricing engine. The Oracle Configurator window calls the package to get:

- list prices of options
- selling prices of all selected components in the configuration
- total price for configuration

The Oracle Configurator window displays the list prices in the Options Selection window and in the Selected Items or Summary window. The Oracle Configurator window caches list prices of the items until it is closed. The Oracle Configurator window assumes that the list price of any item does not depend on which other items are selected and is unchanged during the configuration session.

When the end user clicks the Update Prices or Summary button, the Oracle Configurator window calls the pricing package to get the selling prices for all the selected components and the total price for the configuration. The selling prices are displayed in the Selected Items and Options Selection windows.

> **Note:** The Oracle Configurator window’s performance depends critically on the performance of the supplied pricing interface package. List prices in particular must be returned very quickly, since they are demanded for every option that is displayed.

If the calling application requires access to prices computed for the Oracle Configurator window after the configuration session ends, it is up to the calling application’s interface package to save the computed prices. Prices should be saved together with enough information to allow them to be correlated with the components of the saved configuration.

If the Oracle Configurator window is initialized with a previously saved configuration, it is up to the calling application to either return the saved list and selling prices or to call the pricing engine to get the current price. Direct or manual editing of prices, adjustments, discounts, etc. is the responsibility of the calling application.

### 6.1.1 Oracle Configurator Window Pricing Architecture

The calling application sends an initialization message to the Oracle Configurator window with the interface package and procedure name. The Oracle Configurator window calls this interface package to get current pricing information for a single item or a list of items.

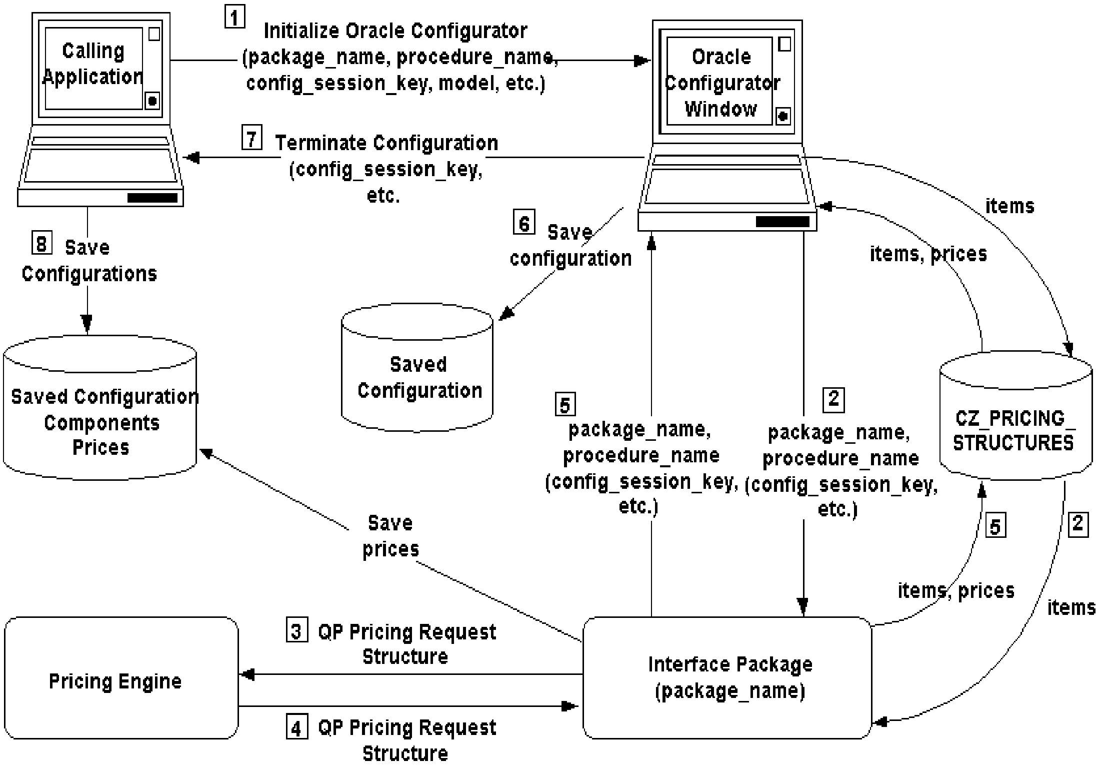
The interface package determines the full context in which to call the target pricing engine. The interface package then calls the pricing engine and captures all of the results, storing these results in tables (or some other Oracle session-insensitive place) for future reference when the Oracle Configurator window session exits. The Oracle Configurator window does not reference the contents of these tables.

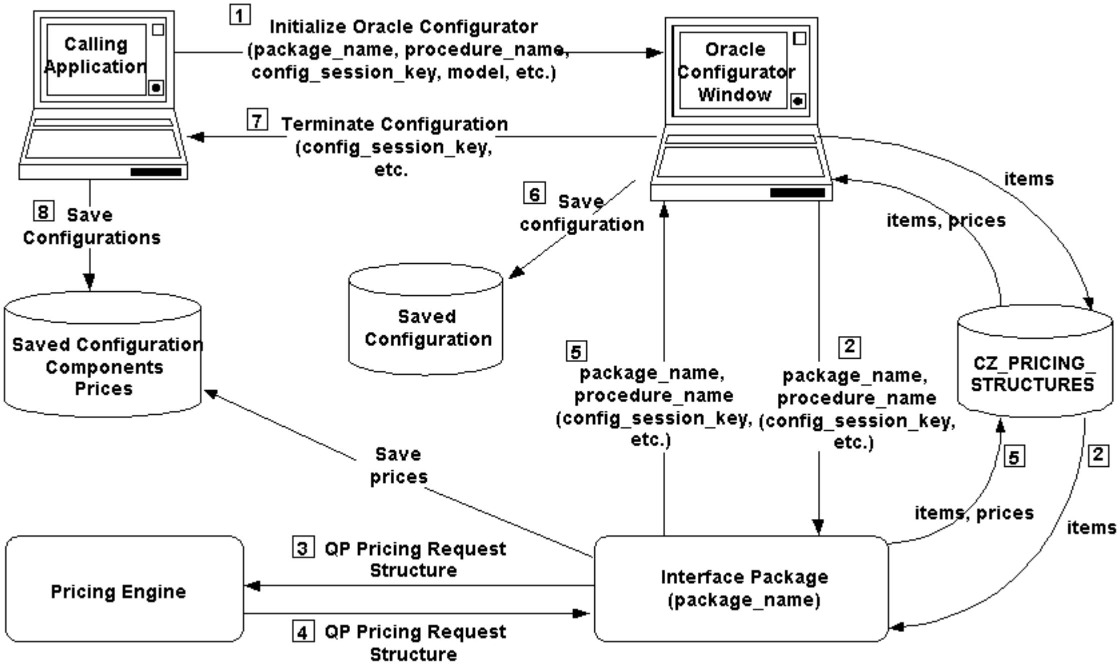
The interface package writes the list and/or selling prices for the configuration components in the temporary CZ_PRICING_STRUCTURES table so that they can be presented to the end user.

The Oracle Configurator window saves the configuration information in the appropriate CZ tables. The Oracle Configurator window does ***not*** save list or selling prices. It is up to the calling application to save configuration data, list prices, and selling prices in their own tables (e.g., OM stores the configuration in OE_ORDER_LINES_ALL, and stores the pricing data in OE_PRICE_ADJUSTMENTS, etc.). The calling application decides whether it is necessary to recalculate prices depending on the value of the prices_calculated_flag in the Oracle Configurator window termination message.

When the calling application calls the Oracle Configurator window to edit an existing configuration, the Oracle Configurator window asks the interface package for the current list and selling prices of the already selected components. Figure 6–1, "Oracle Configurator Window Pricing Architecture", illustrates this architecture. Illustrated steps 2 through 5 can be repeated many times.

***Figure 6–1    Oracle Configurator Window Pricing Architecture***


Calling Application
1 Initialize Oracle Configurator (package_name, procedure_name, config_session_key, model, etc.)
Oracle Configurator Window
7 Terminate Configuration (config_session_key, etc.
8 Save Configurations
6 Save configuration
items
items, prices
Saved Configuration Components Prices
Saved Configuration
5 package_name, procedure_name (config_session_key, etc.)
2 package_name, procedure_name (config_session_key, etc.)
CZ_PRICING_ STRUCTURES
Save prices
5
2
items, prices
items
3 QP Pricing Request Structure
Pricing Engine
Interface Package (package_name)
4 QP Pricing Request Structure


Calling Application
1 Initialize Oracle Configurator (package_name, procedure_name, config_session_key, model, etc.)
Oracle Configurator Window
7 Terminate Configuration (config_session_key, etc.)
8 Save Configurations
6 Save configuration
items
items, prices
Saved Configuration Components Prices
Saved Configuration
5 package_name, procedure_name (config_session_key, etc.)
2 package_name, procedure_name (config_session_key, etc.)
CZ_PRICING_ STRUCTURES
Save prices
5
2
items, prices
items
3 QP Pricing Request Structure
Pricing Engine
Interface Package (package_name)
4 QP Pricing Request Structure

**Note:** See the *Oracle Configurator Toolkit Developer's Guide* for details about the pricing interface package, and about the initialization and termination messages for an Oracle Configurator window session.

## 6.2 Pricing in the Oracle SellingPoint Application

You can quote the price of a product configuration for a customer in Oracle SellingPoint application. Pricing in quotes depends on:

- whether or not your Oracle SellingPoint application is integrated with Oracle Applications - Bills of Material, OrderEntry (10.7 or 11.0), or Order Management
- whether or not you are using Oracle Pricing
- the availability of imported price groups

If you are *not* integrated with Oracle Applications, you can manually enter prices and discounts directly in a quote or you may import price groups into the Oracle Configurator schema.

If you are integrated with Oracle Applications and intend to submit the quote as an order to Oracle Order Entry (10.7 or 11.0) or Order Management (11*i*), you can choose to use Oracle Pricing or use price groups.

The Oracle SellingPoint application only supports prices in United States currency (dollar).

Table 6–1 summarizes Oracle SellingPoint application pricing options in the Quote module:

***Table 6–1 Oracle SellingPoint Application Pricing Options***

| Integrated with Oracle Applications | Oracle Pricing Enabled | Price List Selection Required | Price Groups Imported | Price Group Selected | Price Display |
|---|---|---|---|---|---|
| Yes | Yes | Yes | Yes | Yes | Price Group prices only; *no* discounts allowed. |
| Yes | Yes | Yes | Yes | No | No prices displayed. |
| Yes | Yes | Yes | No | No | No prices displayed. |
| Yes | No | N/A | Yes | Yes | Price Group prices only; discounts allowed. |
| No | N/A | N/A | Yes | Yes | Price Group prices only; discounts allowed. |

## 6.2.1 Oracle Pricing in the Oracle SellingPoint Application

To use Oracle Pricing you must:

- Your Oracle SellingPoint application must transfer data to or from the CZ Schema

- Enable Oracle Pricing in your Oracle SellingPoint application setup by setting the OrclePricing CZ_DB_SETTING to ’Yes’. See Section 3.2, "Oracle Configurator Schema Settings" on page 3-4 for more information about the CZ_DB_SETTINGS.
- Import at least one Oracle Price list (SO_PRICE_LIST). See Chapter 4, "Data Transfer to the CZ Schema" for information on importing data to the CZ Schema.

If Oracle Pricing is enabled in your set up, no prices are displayed and no discounting is allowed in the Quote module. All prices and discounting are associated with the quote in Order Entry (10.7 or 11.0) or Order Management (11*i*) and not replicated back to the Oracle SellingPoint application. When reviewing a submitted order on a quote using Oracle Pricing, total order price is not available in the Oracle SellingPoint application.

You can use Oracle Pricing together with Price Groups imported into the Oracle Configurator schema. Selecting Price Groups causes budgetary prices for the groups to be displayed, but no discounting or price changes are allowed.

When using Oracle Pricing you cannot enter taxes, handling charges, and other manipulations of price in the Oracle SellingPoint application because these are specified by Oracle Pricing.

The data exported to Oracle Order Entry (10.7 or 11.0) or Order Management (11*i*) is the price list id, item, and quantity.

## 6.2.2 Non-Oracle Pricing

If your Oracle SellingPoint application is integrated with Oracle Applications but Oracle Pricing is not enabled in your CZ_DB_SETTINGS table, no Price List is available (meaning the Price List is empty). However, Price Groups and Item Prices may be imported into the Oracle Configurator schema and selected in the Oracle SellingPoint application. Discounts are allowed on these imported prices because no Price List rules apply. The data exported to Oracle Order Entry (10.7 or 11.0) or Order Management (11*i*) in the price list is the item, quantity, discount price, and list price.

If your Oracle SellingPoint application is *not* integrated with Oracle Applications, you can use pricing that is available in your Oracle SellingPoint application Database.

Discounting is also allowed if your Oracle SellingPoint application is not integrated with Oracle Applications. Discounts are allowed on any imported prices.

# 7

# Installing the Servlet

In order to run an Oracle Configurator window (JAVA applet or DHTML), you must first install the UI Servlet on your Internet server. This process consists of the following tasks:

***Table 7–1 Overview of tasks for installing the servlet***

| For This Task ... | See ... |
|---|---|
| Installing the servlet on your Internet server. | Installing the Apache Internet Server and Supporting Software on page 7-2 |
| Modifying the configuration files for your Internet server. | Configuring Apache and JServ on page 7-4 |
| Making adjustments to accommodate special needs, such as load balancing. (Conditional) | Load Balancing on page 7-10 |

> **Note:** The installation procedure is specified for Apache on the Solaris platform. You may need to adapt the procedure for other Internet servers.

## 7.1 Related Documentation

You will need to consult the following documentation when installing the Oracle Configurator servlet:

- Apache 1.3 User's Guide and Apache web site (`java.apache.org`).

## 7.2 Prerequisites

- You must have installed Oracle Applications.
- You must have installed your Internet server. The following Internet servers have been tested for use with Oracle Configurator:
  - Apache version 1.3.9 (with Apache JServ version 1.1)
  - Oracle Application Server (OAS), version 4.0.8.1
- You must have JDBC connectivity between Apache and your Oracle Applications database. This may include copying or editing connectivity parameters in the TNSNAMES.ORA file for Apache.
- Check the *Oracle Configurator and SellingPoint Release Notes* for any other requirements.

## 7.3 Installing the UI Servlet

Installing the UI Servlet consists of installing the Apache Internet Server, installing supporting software, and configuring Apache and JServ to work with the Oracle Configurator UI Servlet.

### 7.3.1 Installing the Apache Internet Server and Supporting Software

In the usual case, Apache will be installed as part of your Oracle Applications installation. This section summarizes the procedure for installing the Apache Internet server. It is intended to provide context for the instructions in Configuring Apache and JServ on page 7-4, which are specific to configuring Apache for the Oracle Configurator UI Servlet. For full details on installing the Apache Internet server, consult the Apache documentation.

In the examples below, the following values are substituted for the placeholders, as shown:

| **Placeholder** | **Example Value** | **Comment** |
|---|---|---|
| `apache_install` | `/d1/apache/oct` | The directory in which you install Apache. |
| `hostname` | `www.mysite.com` | The name of the host machine. |
| `portnum` | `7070` | The port number used by the Apache listener, which is specified by `Port` in `httpd.conf`. |

| Placeholder | Example Value | Comment |
|---|---|---|
| *html_vpath* | html | A directory located under /htdocs, which is specified by DocumentRoot in httpd.conf. |
| *media_vpath* | media | A directory located under /htdocs, which is specified by DocumentRoot in httpd.conf. |
| *images_vpath* | images | A directory located under /htdocs, which is specified by DocumentRoot in httpd.conf. |
| *servlet_vpath* | servlets | A mounting location specified by ApJServMount in jserv.conf. |

### 7.3.1.1 Prerequisite Software

You will need the following software to install Apache for use with Oracle Configurator:

- The GNU Make 3.78.1 source distribution
- The Sun JDK 1.1.8 binary distribution
- The Sun JSDK 2.0 binary distribution
- The Apache 1.3.9 binary distribution
- The Apache JServ 1.1 source distribution (which requires Sun JSDK 2.0)

### 7.3.1.2 Summary Installation Procedure

In these instructions, the directory in which you install Apache will be referred to as *apache_install*.

1. Build GNU make.

   Begin by unpacking GNU make from its tar file.

   ```
   cd /where/you/unpacked/the/tar/file

   ./configure --prefix=/where/you/want/it/installed

   make

   make install

   setenv PATH /where/you/installed/make:$PATH
   ```

2. Install JDK 1.1.8

   Begin by unpacking the JDK tar file.

   ```
   cd /where/you/unpacked/the/tar/file

   pkgadd -d . SUNWjvrt SUNWjvdev SUNWjvjit SUNWjvman SUNWjvdem

   rehash
   ```

3. Install JSDK 2.0.

   Begin by unpacking the JSDK tar file.

   ```
   setenv CLASSPATH /where/you/unpacked/jsdk/lib/jsdk.jar:$CLASSPATH
   ```

4. Install Apache 1.3.9

   Begin by unpacking the Apache tar file.

   ```
   ./install-bindist.sh apache_install
   ```

5. Install JServ 1.1.

   Begin by unpacking the JServ tar file.

   ```
   cd /where/you/unpacked/tar/file

   ./configure --prefix=/where/you/install/jserv \
   --with-apxs=apache_install/bin/apxs \
               --with-jdk-home=/where/you/installed/jdk \
               --with-JSDK=/where/you/unpacked/jsdk/lib/jsdk/jar

   make

   make install
   ```

## 7.3.2 Configuring Apache and JServ

After you have installed Apache and its supporting software, you must modify certain configuration files to work with the OC servlet. This section contains a summary of the modifications you must make to the Apache configuration files.

- For full details on this task, consult the instructions on cartridge and component administration in the Apache documentation.

- You will need to log in as the owner of the Apache files in order to perform these steps.
- The files to be edited are:
  - `httpd.conf` on page 7-5
  - `jserv.conf` on page 7-5
  - `jserv.properties` on page 7-6
  - `zone.properties` on page 7-8
- See Java System Property Parameters for the UI Servlet on page 7-14 for descriptions of the individual parameters.

***Table 7–2 Modifications to Apache Configuration Files***

| File to modify | Modifications |
|---|---|
| `httpd.conf` | Modify the following parameters to point to *apache_install*:<br>`ServerRoot "`*apache_install*`"`<br>`DocumentRoot "`*apache_install*`/htdocs"`<br>`<Directory "`*apache_install*`/htdocs">`<br>`Alias /icons/ "`*apache_install*`/icons/"`<br>`<Directory "`*apache_install*`/icons">`<br>`ScriptAlias /cgi-bin/ "`*apache_install*`/cgi-bin/"`<br>`<Directory "`*apache_install*`/cgi-bin">`<br>Change Apache’s listening port to one that is not being used on the server machine:<br>`Port` *portnum*<br>At the very end of `httpd.conf`, add a line that points to the location of the JServ configuration file `jserv.conf`, which is located in *jserv_install*`/conf`, for example:<br>`Include` *apache_install*`/conf/jserv/jserv.conf` |
| `jserv.conf` | Modify the following parameters to point to *apache_install*:<br>`LoadModule jserv_module` *apache_install*`/libexec/mod_jserv.so`<br>`ApJServLogFile` *apache_install*`/logs/mod_jserv.log` |

*Table 7–2 (Cont.) Modifications to Apache Configuration Files*

| File to modify | Modifications |
|---|---|
| | Set `ApJServProperties` to point to the location of the JServ properties file `jserv.properties`, which is located in *`jserv_install`*`/conf`:<br>`ApJServProperties` *`apache_install`*`/conf/jserv/jserv.properties`<br>Change JServ's listening port to one that is not being used on the server machine (and different from the Port setting in `httpd.conf`):<br>`ApJServDefaultPort` *`portnum_jserv`*<br>Ensure that there is a valid mount point for the UI Servlet zone, and the virtual path for the zone:<br>`ApJServMount` *`/name_of_zone`* *`/virtual_path_of_jserv`*<br>Example:<br>`ApJServMount /servlet /root` |
| `jserv.properties` | Add the following parameters as `wrapper.bin.parameters`. These are values to be passed to the Java Virtual Machine (JVM):<br>`wrapper.bin.parameters=-Dcz.activemodel=/where/you/want/lce_files`<br>`wrapper.bin.parameters=-Dcz.html.source.treeview=http://`*`hostname`*`:`*`portnum/html_vpath`*`/cztree.htm`<br>`wrapper.bin.parameters=-Dcz.html.source.display=http://`*`hostname`*`:`*`portnum/html_vpath`*`/czdisp.htm`<br>`wrapper.bin.parameters=-Dcz.uimanager.logpath=/where/you/want/exception/logs`<br>If you are using Oracle Configurator within Oracle Applications, also add the following parameter as `wrapper.bin.parameters` . This is a value to be passed to the Java Virtual Machine (JVM):<br>`wrapper.bin.parameters=-DFND_TOP=/where/this/apps/FND_TOP/points/to` |

*Table 7–2 (Cont.) Modifications to Apache Configuration Files*

| File to modify | Modifications |
|---|---|
| | Add an entry like one of the following examples to set the maximum heap size. (We recommend at least 128MB; your configuration may differ.)<br>`wrapper.bin.parameters=-mx209715200`<br>`wrapper.bin.parameters=-mx128m`<br>You can also add the following optional `wrapper.bin.parameters` entries to set other system properties.:<br>`wrapper.bin.parameters=-Dcz.uiserver.wakeintervalseconds=`*number_seconds*<br>`wrapper.bin.parameters=-Dcz.uiserver.timeoutseconds=`*number_seconds*<br>`wrapper.bin.parameters=-Dcz.frameset.allocations.top=`*frameset_widths*<br>`wrapper.bin.parameters=-Dcz.scrolling.treeview=auto`<br>If you are configuring Apache for using the Oracle Configurator window outside of Oracle Applications, add the following `wrapper.classpath` values, modifying the values to point to *`apache_install`*:<br>`wrapper.classpath=`*apache_install*`/libexec/ApacheJServ.jar`<br>`wrapper.classpath=`*jsdk_install*`/lib/jsdk.jar`<br>`wrapper.classpath=`*servlet_vpath*`/servlets`<br>`wrapper.classpath=`*oct_install*`/cz3rdpty.jar`<br>`wrapper.classpath=`*oct_install*`/config.jar`<br>`wrapper.classpath=`*jdbc_drivers_install*`/classes111.zip`<br>If you are configuring Apache for using the Oracle Configurator window embedded in Oracle Applications, add the following `wrapper.classpath` values, modifying the values to point to *`apache_install`*:<br>`wrapper.classpath=`*apache_install*`/libexec/ApacheJServ.jar`<br>`wrapper.classpath=`*jsdk_install*`/lib/jsdk.jar` |

*Table 7–2 (Cont.) Modifications to Apache Configuration Files*

| File to modify | Modifications |
|---|---|
| | wrapper.classpath=*servlet_vpath*/servlets<br>wrapper.classpath=*value_of_$AU_TOP*/115/java/3rdparty.zip<br>wrapper.classpath=*value_of_$AU_TOP*/115/java/apps.zip<br>wrapper.classpath=*jdbc_drivers_install*/classes111.zip<br>Add the following to the `wrapper.env.copy` section:<br>wrapper.env.copy=LD_LIBRARY_PATH<br>where LD_LIBRARY_PATH contains the directory in which the `*.so` files are available (i.e., `libcz.so` and `libczjni.so`).<br>wrapper.env.copy=ORACLE_HOME<br>Change the port number to match the ApJServDefaultPort that you set in `jserv.conf`:<br>port=*port_number*<br>Modify the following parameters to point to *apache_install*<br>root.properties=*apache_install*/conf/jserv/zone.properties<br>log.file=*apache_install*/logs/jserv.log |
| `zone.properties` | Modify the following parameter to point to the directories that contain the classes to be reloaded when they are modified.<br>repositories=*/where/modified/classes/are/reloaded/from*<br>Example:<br>repositories=*apache_install*/servlets,/home/classes<br>Add the following `servlets.default.initArgs` values to set system properties (this replaces the use of a separate .initArgs file):<br>servlets.default.initArgs=-Dcz.uiservlet.templateurl=http:*//hostname:portnum/html_vpath*/US/czFraNS.htm<br>servlets.default.initArgs=-Dcz.uiservlet.templateurl.ie=http:*//hostname:portnum/html_vpath*/US/czFraIE.htm<br>servlets.default.initArgs=-Dcz.uiservlet.stylesheet.applet=http:*//hostname:portnum/html_vpath*/czcmdcvt.xsl |

*Table 7–2 (Cont.) Modifications to Apache Configuration Files*

| File to modify | Modifications |
|---|---|
| | servlets.default.initArgs=-Dcz.uiservlet.stylesheet.applet.client=http://*hostname:portnum/html_vpath*/client.xsl |
| | servlets.default.initArgs=-Dcz.uiservlet.stylesheet.applet.server=http://*hostname:portnum/html_vpath*/server.xsl |
| | servlets.default.initArgs=-Dcz.uiservlet.stylesheet.dhtml=http://*hostname:portnum/html_vpath*/czxml2js.xsl |
| | servlets.default.initArgs=-Dcz.uiservlet.url=http://*hostname:portnum/servlet_vpath*/oracle.apps.cz.servlet.UiServlet |
| | servlets.default.initArgs=-Dcz.uiservlet.proxyurl=http://*hostname:portnum/servlet_vpath*/Proxy |
| | servlets.default.initArgs=-Dcz.uiservlet.proxyscript=http://*hostname:portnum/html_vpath*/czProxy.js |
| | servlets.default.initArgs=-Dcz.uiservlet.sourcefile=http://*hostname:portnum/html_vpath*/czSource.htm |
| | servlets.default.initArgs=-Dcz.uiservlet.jdbcdriver=oracle.jdbc.driver.OracleDriver |
| | servlets.default.initArgs=-Dcz.uiservlet.logfilename=*apache_install*/logs/cz |
| | servlets.default.initArgs=-Dcz.uiservlet.applet.tmp=*apache_install*/logs/ |
| | servlets.default.initArgs=-Dcz.uiservlet.renderjsdirect=true |

### 7.3.3 Starting Apache

After you have installed and configured the Apache server and supporting software, you can start and test the Oracle Configurator UI Servlet.

1. Start Apache with this command:

   *apache_install*/bin/apachectl start

2. If the listener does not start, check whether you are logged on as the owner of the Apache files, thus allowing the server process to write to the log files.
3. Test the Oracle Configurator UI Servlet, by opening a browser and entering the following URL:

   ```
   http://hostname:portnum/servlet_
   vpath/oracle.apps.cz.servlet.UiServlet?test=test_string
   ```

   where *portnum* is the number for Apache's listening port, which you set in the `Port` parameter in `httpd.conf`.

   For example:

   ```
   http://www.mysite.com:7070/servlet/oracle.apps.cz.servlet.UiServlet?test=thi
   s_works
   ```

   The browser should display a page containing only the test string:

   ```
   this_works
   ```

## 7.3.4 Load Balancing

You may need to adjust your Apache configuration to balance the load of visits to the UI Servlet by your end users.

Load balancing can be performed at the level of the Apache web listener (HTTPD), and also by creating multiple instances of the JServ servlet engine. The second approach is outlined here.

In order to run multiple instances of the JServ engine, you must first turn off the automatic spawning of JServ and start your instances in manual mode.

The Solaris shell in which you run JServ should set the maximum number of file descriptors to 1024:

```
ulimit -n 1024
```

You should use JDK 1.1.8.

You should set the maximum heap size for the Java Virtual Machine to an optimal value, such as 200MB. Do this by passing a runtime parameter, like this:

```
-mx209715200
```

Further information about load balancing, from the Apache provider, is available at:

```
http://java.apache.org/jserv/howto.load-balancing.html
```

### Load Balancing Procedure

Here is a possible procedure to set up load balancing:

1. In `jserv.conf`, change the `ApJServManual` setting to `on`:

   ```
   ApJServManual on
   ```

2. In `jserv.conf`, change the `ApJServMount` property to:

   ```
   ApJServMount /servlet balance://set1/root
   ```

   Where `set1` is your arbitrary name for the set of JServ instances that you are spawning.

3. In `jserv.conf`, add settings for the JServ instances that you intend to run:

   ```
   ApJServBalance set1 host1jserv1
   ApJServBalance set1 host1jserv2
   ApJServBalance host1jserv1 ajpv12://hostname:portnum_jserv1
   ApJServBalance host1jserv2 ajpv12://hostname:portnum_jserv2
   ApJServRoute JS1 host1jserv1
   ApJServRoute JS2 host1jserv2
   ```

   Where:

   | | |
   |---|---|
   | `set1` | the name of the set of JServ instances that you specified for `ApJServMount`, in step 2. |
   | `host1jserv1`<br>`host1jserv2` | your arbitrary names for use in tracing in log files. |
   | `ajpv12` | indicates the protocol used (Apache Jserv Protocol version 1.2) |
   | *`hostname`* | the host machine(s) on which you are running your JServ instances. |
   | *`portnum_jserv1`*<br>*`portnum_jserv2`* | the ports for JServ instances that you specified in `jserv.conf`, with `ApJServDefaultPort portnum_jserv`. |

4. Set up your JServs instances to run standalone. To do this you should use a shell script. An example script is shown in Example 7–1 on page 7-13.

Be aware that none of the `wrapper.*` properties are applied when you start a JServ instance manually, so you must pass them as command line arguments in the script. So, in the script, you set your CLASSPATH and command line arguments to be passed to the JVM. You also pass the properties file that you want to use to start the JServ (`jserv.properties`).

5. In `jserv.properties`, change `bindaddress` to:

   ```
   bindaddress=*
   ```

   Note that setting this parameter to `*` presents a possible security risk. Consult the Apache documentation and comments in `jserv.properties` for details.

   In a production environment `bindaddress` would be the IP address where you will be receiving requests.

6. In `jserv.properties`, change `port` to:

   ```
   port=portnum_jserv
   ```

   *`portnum_jserv`* is the port for JServ that you specified in `jserv.conf`, with `ApJServDefaultPort portnum_jserv`.

7. You may optionally want to change the `root.properties` setting to point to a different file for each servlet, in order to read a different set of runtime parameters (for instance, to send your logs to go to a different directory for each servlet). See Configuration for Multiple Properties Files for details.

8. Start up Apache (see Starting Apache on page 7-9).

   ```
   apache_install/bin/apachectl start
   ```

9. Start up your JServ instances by running the script in Example 7–1.

### Configuration for Multiple Properties Files

If you are following step 7 on page 7-12, edit the configuration files as follows, for each JServ instance that you are creating.

Create another version of `jserv.properties` (e.g., `jserv2.properties`). In this alternate version:

- Comment out the `wrapper.bin.parameters` and `wrapper.classpath` properties that you are setting in the script.
- Point to a new `zone.properties` file that you will create:

  ```
  root.properties=/d1/apache/oct/conf/jserv/zone2.properties
  ```

- Point to a different log file:

  ```
  log.file=/d1/apache/oct/logs2/jserv.log
  ```

- Change the port number, according to what you set in step 6:

  ```
  port=portnum_jserv2
  ```

Create another version of `zone.properties` (e.g., `zone2.properties`). In this alternate version:

- Point to different log files:

  ```
  servlets.default.initArgs=cz.uiservlet.logfilename=/d1/apache/oct/logs2/cz
  servlets.default.initArgs=cz.uiservlet.applet.tmp=/d1/apache/oct/logs2/
  ```

Create another version of the script in Example 7–1. In this alternate version, change the lines indicated in the example by the comment "# for 2nd JServ instance...".

***Example 7–1 Shell Script for Starting JServ in Manual Mode for Oracle Configurator window embedded in Oracle Applications***

```
#!/bin/sh

ulimit -n 1024

properties=/d1/apache/oct/conf/jserv/jserv.properties
# for 2nd JServ instance, change above line to:
# properties=/d1/apache/oct/conf/jserv/jserv2.properties

log=/d1/apache/oct/logs/jserv1.log
# for 2nd JServ instance, change above line to:
# log=/d1/apache/oct/logs/jserv2.log

cmdlineargs="-Dcz.activemodel=/d1/apache/oct/logs \
           -Dcz.html.source.treeview=http://www.mysite.com:2525/html/cztree.htm
\
            -Dcz.html.source.display=http://www.mysite.com:2525/html/czdisp.htm
\
            -Dcz.uimanager.logpath=/d1/apache/oct/logs/ \
            -mx209715200"
# for 2nd JServ instance, change the "logpath" line above to:
# -Dcz.uimanager.logpath=/d1/apache/oct/logs2/ \

echo $cmdlineargs >> $log
```

```
CLASSPATH=/local/java/jdk1.1.6/lib/classes.zip
CLASSPATH=$CLASSPATH:/home/user123/jsdk.jar
CLASSPATH=$CLASSPATH:/d1/apache/oct/libexec/ApacheJServ.jar
CLASSPATH=$CLASSPATH:/d1/my-oct/classes
CLASSPATH=$CLASSPATH:/d1/my-oct/classes/apps.zip
CLASSPATH=$CLASSPATH:/d1/my-oct/classes/3rdparty.zip
CLASSPATH=$CLASSPATH:/local/db/8.0.6/jdbc/lib/classes111.zip
echo $CLASSPATH >> $log

java $cmdlineargs -classpath $CLASSPATH org.apache.jserv.JServ $properties $1
2>> $log &
sleep 3
```

In Example 7–1, `apps.zip` is `config.jar`, and `3rdparty.zip` is `cz3rdpty.jar`, if the UI Servlet is for the Oracle Configurator window running outside of Oracle Applications.

## 7.4 Java System Property Parameters for the UI Servlet

Most of these Java system property parameters are compatible with either Apache or Oracle Application Server (OAS).

In Apache, they are specified in `jserv.properties` like this:

wrapper.bin.parameters=-D*property_name*=*property_definition*

and in `zone.properties` like this:

servlets.default.initArgs=-D*property_name*=*property_definition*

In Oracle Application Server, they are specified as Java Environment Parameters of the type SYSTEM_PROPERTY.

### cz.activemodel

Path to read and write logic files on the server. The Active Model Path comes from either cz.activemodel, cz.uimanager.logpath, or user.dir, in that order. If the UI Servlet is for the Oracle Configurator window running embedded within Oracle Applications, this parameter is not needed.

Syntax:

cz.activemodel=*activemodel_dir/*

Example for Solaris:

```
cz.activemodel=/d1/my-oct/activemodel/
```

Example for Windows NT:

```
cz.activemodel=D:\orant\OSP\Shared\ActiveModel\
```

### cz.uiservlet.templateurl

URL of HTML template for DHTML client running in a Netscape browser.

Syntax:

cz.uiservlet.templateurl=http://*hostname:portnum/html_vpath/*US/czFraNS.htm

Example:

```
cz.uiservlet.templateurl=http://www.mysite.com/html/US/czFraNS.htm
```

### cz.uiservlet.templateurl.ie

URL of HTML template for DHTML client running in a Microsoft Internet Explorer browser.

Syntax:

cz.uiservlet.templateurl.ie=http://*hostname:portnum/html_vpath/*US/czFraIE.htm

Example:

```
cz.uiservlet.templateurl.ie=http://www.mysite.com/html/US/czFraIE.htm
```

### cz.uiservlet.stylesheet.applet

Syntax:

cz.uiservlet.stylesheet.applet=http://*hostname:portnum/html_vpath/*czcmdcvt.xsl

Example:

```
cz.uiservlet.stylesheet.applet=http://www.mysite.com/html/czcmdcvt.xsl
```

### cz.uiservlet.stylesheet.applet.client

Syntax:

cz.uiservlet.stylesheet.applet.client=http://*hostname:portnum/html_*

```
vpath/czclient.xsl
```

Example:

```
cz.uiservlet.stylesheet.applet.client=http://www.mysite.com/html/czclient.xsl
```

### cz.uiservlet.stylesheet.applet.server

Syntax:

```
cz.uiservlet.stylesheet.applet.server=http://hostname:portnum/html_
vpath/czserver.xsl
```

Example:

```
cz.uiservlet.stylesheet.applet.server=http://www.mysite.com/html/czserver.xsl
```

### cz.uiservlet.stylesheet.dhtml

Syntax:

```
cz.uiservlet.stylesheet.dhtml=http://hostname:portnum/html_vpath/czxml2js.xsl
```

Example:

```
cz.uiservlet.stylesheet.dhtml=http://www.mysite.com/html/czxml2js.xsl
```

### cz.uiservlet.url

URL of the Oracle Configurator UI Servlet.

Syntax:

```
cz.uiservlet.url=http://hostname:portnum/servlet_
vpath/oracle.apps.cz.servlet.UiServlet
```

Example:

```
cz.uiservlet.url=http://www.mysite.com/servlet/oracle.apps.cz.servlet.UiServlet
```

### cz.uiservlet.proxyurl

URL of the Oracle Configurator Proxy.

Syntax:

```
cz.uiservlet.proxyurl=http://hostname:portnum/servlet_
vpath/oracle.apps.cz.servlet.Proxy
```

Example:

```
cz.uiservlet.proxyurl=http://www.mysite.com/servlet/oracle.apps.cz.servlet.Proxy
```

## cz.uiservlet.proxyscript

URL of JavaScript source for proxy frame.

Syntax:

```
cz.uiservlet.proxyscript=http://hostname:portnum/html_vpath/czProxy.js
```

Example:

```
cz.uiservlet.proxyscript=http://www.mysite.com/html/czProxy.js
```

## cz.uiservlet.sourcefile

Specifies the HTML template file that produces the Source Frame in the configuration window.

Syntax:

```
cz.uiservlet.sourcefile=http://hostname:portnum/html_vpath/czSource.htm
```

Example:

```
cz.uiservlet.sourcefile=http://www.mysite.com/html/czSource.htm
```

## cz.uiservlet.jdbcdriver

JDBC driver for connecting the UI Servlet to a database.

Syntax:

```
cz.uiservlet.jdbcdriver=driver_class
```

Example:

```
cz.uiservlet.jdbcdriver=oracle.jdbc.driver.OracleDriver
```

### cz.uiservlet.logfilename

The path into which the UI Servlet will write logging files. Do not specify a specific file name. If this parameter is omitted, no logging files will be written.

Syntax:

```
cz.uiservlet.logfilename=logging_dir
```

Example for Solaris:

```
cz.uiservlet.logfilename=/d1/my-oct/log
```

Example for Windows NT:

```
cz.uiservlet.logfilename=D:\orant\OSP\OSP\log
```

### cz.html.source.treeview

Specifies the HTML template file that produces the tree view in the configuration window.

Syntax:

```
cz.html.source.treeview=http://hostname:portnum/html_vpath/cztree.htm
```

Example:

```
cz.html.source.treeview=http://www.mysite.com/html/cztree.htm
```

### cz.html.source.display

Specifies the HTML template file that produces the primary display view in the configuration window.

Syntax:

```
cz.html.source.display=http://hostname:portnum/html_vpath/czdisp.htm
```

Example:

```
cz.html.source.display=http://www.mysite.com/html/czdisp.htm
```

### cz.uimanager.logpath

The path into which the UI Server will write a log file when it handles an exception and exits.

Syntax:

```
cz.uimanager.logpath=logging_dir
```

Example for Solaris:

```
cz.uimanager.logpath=/d1/my-oct/log/
```

Example for Windows NT:

```
cz.uiservlet.logfilename=D:\orant\OSP\OSP\log\
```

### cz.uiservlet.applet.tmp

Syntax:

```
cz.uiservlet.applet.tmp=logging_dir
```

Example for Solaris:

```
cz.uiservlet.applet.tmp=/d1/my-oct/log/
```

Example for Windows NT:

```
cz.uiservlet.applet.tmp=D:\orant\OSP\OSP\log\
```

### cz.uiservlet.renderjsdirect

Deprecated.

Controls whether the UI Server renders DHTML output directly through JavaScript, instead of translating XML through XSL style sheets.

Set the value to `true` to turn on direct JavaScript rendering. Set the value to `false`, or omit the property, to render through XML and XSL. The default value is `false`.

Syntax:

```
cz.uiservlet.renderjsdirect=[true|false]
```

Example:

```
cz.uiservlet.renderjsdirect=true
```

### cz.uiserver.wakeintervalseconds

The number of seconds between executions of the UI Server's session timeout thread (which is always running, as a scavenger thread). The default value is 480 seconds (8 minutes).

See also cz.uiserver.timeoutseconds.

Syntax:

```
cz.uiserver.wakeintervalseconds=number_seconds
```

Example:

```
cz.uiserver.wakeintervalseconds=960
```

### cz.uiserver.timeoutseconds

The number of seconds that a session is inactive before it is timed out by the UI Server's session timeout thread. The default value is 1200 seconds (20 minutes).

See also cz.uiserver.wakeintervalseconds.

Syntax:

```
cz.uiserver.timeoutseconds=number_seconds
```

Example:

```
cz.uiserver.timeoutseconds=2400
```

### cz.frameset.allocations.top

Controls the relative sizes of the tree view and display frames in the Content Frame of the configuration window. The default value is `"30%, *"`.

Syntax:

```
cz.frameset.allocations.top=frameset_widths
```

Example:

```
cz.frameset.allocations.top=25%,*
```

### cz.scrolling.treeview

Controls whether there is scrolling in the tree view frame in the configuration window. The default value is "auto".

Syntax:

```
cz.scrolling.treeview=["auto"|"yes"|"no"]
```

Example:

```
cz.scrolling.treeview="yes"
```

### FND_TOP

Required for locating the DBC file used for database connectivity. To determine the value to use, enter the command

```
% echo $FND_TOP
```

in a command shell having the desired Oracle Applications environment.

Syntax:

```
FND_TOP=local value of $FND_TOP
```

Example:

```
FND_TOP=/d1/fnd/test
```

# 8

# Oracle Configurator Deployment

This chapter contains information regarding deployment of the Oracle runtime configurators. See Section 1.6, "Oracle Configurator Installations" on page 1-6 for an overview of possible deployments.

## 8.1 Oracle Configurator Window

The Oracle Configurator window add-on to forms-based Oracle Applications such as Order Management is a Java applet. HTML-based applications such as iStore, or a custom web application use a DHTML Oracle Configurator window. In either case, Oracle Applications use an internet server to run the UI Servlet that connects the Oracle Configurator window URL to the Oracle Configurator schema. That URL is set in the profile option BOM:CZ_UIMGR_URL.

See Chapter 7, "Installing the Servlet" for information about installing the UI Servlet and configuring the internet server.

### Java Applet

The requirements for the Oracle Configurator window as a Java applet running in an Oracle Applications Form are already in place when running Oracle Applications:

- Initialization strings must contain the appropriate parameters for the calling application to call a Java applet Oracle Configurator window.
- Termination strings must contain the appropriate parameters for the calling application to call a Java applet Oracle Configurator window.

### DHTML

The requirements for the Oracle Configurator window running in an internet (DHTML) application are:

- Initialization strings must contain the appropriate parameters for the calling application to call a DHTML Oracle Configurator window. These must be set up manually for integration in a custom web application.
- Stylesheets and Javascript are essential components of DHTML. Do not disable display of Javascript and Stylesheets in the browser running the DHTML Oracle Configurator window.
- The DHTML Oracle Configurator window is best viewed with Internet Explorer 4.0 or higher, and Netscape 4.07 or higher.
- The browser must be set up to accept and/or send cookies.
- Recommended screen resolution is 800 X 600 or higher.
- To EXIT the Oracle Configurator window, always use the Done or Cancel button and then EXIT the web browser if necessary.
- End users should make a note of any confirmation numbers displayed when you request changes to an existing configuration.
- Do not use the browser's Back button to view a prior Oracle Configurator window screen.
- Termination strings must contain the appropriate parameters for the calling application to terminate a DHTML Oracle Configurator window. These must be set up manually for integration in a custom web application.

For details about the initialization and termination messages in custom web applications, see the *Oracle Configurator Toolkit Developer's Guide*.

## 8.1.1 Oracle Configurator Window Profile Options

During your implementation, you set a value for each profile option to specify how Oracle Applications accesses the Oracle Configurator window.

You must set up and updates profile option values for users of Oracle Configurator windows within Oracle Applications. See the *Oracle Configurator and SellingPoint ReadMe* or Oracle Applications Online Help for information about profile options required for running an Oracle Configurator window within Oracle Applications.

### 8.1.2 Accessing an Oracle Configurator Window

End users for the Oracle Configurator window (DHTML or Java Applet) are established through Oracle Applications administration and reside in the Oracle Applications database.

End users click on the Configure button in Oracle Applications that call the Oracle Configurator window. This action causes the calling application to initialize an Oracle Configurator window session in an initialization message. When an end user saves a configuration and exits an Oracle Configurator window session, the Oracle Configurator window sends the calling application a termination message to terminate the Oracle Configurator window session.

For more information about the initialization messages, see the *Oracle Configurator Toolkit Developer's Guide*.

## 8.2 Oracle SellingPoint Application

The Oracle SellingPoint application is a standalone runtime end-user environment for configuring products and services. The Oracle SellingPoint application includes opportunity management, quotes, proposals, and a configurator. The configurator and customizations to the configurator screens in the Oracle SellingPoint application are developed in Oracle Configurator Developer.

The Oracle SellingPoint application is available as a test environment in Oracle Configurator Developer and can be deployed as a client/server or mobile production application. No matter what the deployment method, the administrator must prepare for installations and login access.

### 8.2.1 Oracle SellingPoint Application User Access

To set up a networked or mobile Oracle SellingPoint application for specific end users to log into the application and have access to specific configuration models, user interfaces, and account information you must:

- Add end users to the CZ_END_USERS table
- Assign end users to Projects
- Assign end users to Customers

#### 8.2.1.1 Add End Users

When you install your Oracle SellingPoint application, make the DBOwner (<oc>) the default user. The <oc> user must be inserted into the CZ_END_USERS table in the Oracle Configurator schema. When transferring data from the Oracle Applications database, Oracle Applications users are inserted in the CZ_END_USERS table in the Oracle Configurator schema (see Section 2.2.2, "Create Users and Responsibilities" on page 2-3).

When deploying an Oracle SellingPoint application, you can add more end users and associate user groups with the end users, as needed.

To add end users and associate user groups:

1. Log into the Oracle SellingPoint application on the end user's machine as the DBOwner of the Oracle Configurator schema serving that Oracle SellingPoint application.

   Example, log in as <oc> with the DSN that points to the Oracle Configurator schema on the server.

2. Go to the Tools menu.
3. Select Administration --> Users...
4. Change user group assignments or contact information to existing users, or add new end users. To add new end users, click the New User... button.
5. In the New User dialog, enter Login Name, Organization ID (relevant if integrated with Oracle Applications) and the password for the new user. Users cannot change their passwords on their own.
6. Click OK.
7. With the newly added end user highlighted in the Users list, open the User Groups Tab.
8. In the User Groups tab, make the selected end user a member of one of the available user groups. Only End User is available and implemented in Release 11*i*
9. In the Contact Details tab, insert any relevant information for the selected end user.
10. Click Apply to commit the new information to the Oracle Configurator schema serving the current Oracle SellingPoint application.

#### 8.2.1.2 Assign End Users to Projects

In Oracle Configurator Developer, the work space for building a configuration model is a Project. A Project in Configurator Developer corresponds to an Active Model in the Oracle SellingPoint application. The Active Model is the configuration model. It is accessible in the Configuration module in the Oracle SellingPoint application where the end user makes selections to create any number of possible configurations allowable by that Active Model.

Before an end user can log in to an Oracle SellingPoint application and access an Active Model, the administrator of the deployment must associate each end user with the Project that corresponds to that Active Model.

To associate end users with Projects:

1. Log into the Oracle SellingPoint application on the end user’s machine as the DBOwner of the Oracle Configurator schema serving that Oracle SellingPoint application.

   Example, log in as <oc> with the DSN that points to the Oracle Configurator schema on the server.

2. Go to the Tools menu.
3. Select Administration --> Assign Projects...
4. Select the User Group "End User". This is the user group to which end users of a deployed Oracle SellingPoint application belong.
5. Select from the Projects: drop down list to display candidate projects in the Name/Description view below.
6. Select from the candidate projects list and establish access to the selected project by clicking the arrow to move the Project to the Associated Projects: list.
7. Close the Assign Projects dialog.

### 8.2.1.3 Assign End Users to Customers

Before an end user can log in to an Oracle SellingPoint application and access available account information (customer, address, contact), the administrator of the deployment must associate each end user with at least one account.

To associate end users with customers:

1. Log into the Oracle SellingPoint application on the end user’s machine as the DBOwner of the Oracle Configurator schema serving that Oracle SellingPoint application.

   Example, log in as <oc> with the DSN that points to the Oracle Configurator schema on the server.

2. Go to the Tools menu.
3. Select Administration --> Assign Customers...
4. Use the same method to assign customers as is described for assigning Projects (see Section 8.2.1.2, "Assign End Users to Projects" on page 8-4).

## 8.2.2 Oracle SellingPoint Application Installation

You can install a default Oracle SellingPoint application simply by installing it from the Oracle Configurator Developer CD and establishing connectivity to the Oracle Configurator schema containing the configuration models you wish to access.

Early during development of a customized Oracle SellingPoint application the set of files needed to run the application must be bundled into an installable file structure for further development and testing. Depending on whether you want to create an installation wizard for end-users to install the Oracle SellingPoint application themselves, you will need to make decisions about how much of the installation to automate. Independent of the degree of automation or which installation software you choose to create your install program, you need to:

- run OCRuntime.exe to install all of Oracle Configurator required core files
- add required project-specific files to the default file structure (i.e. bitmaps, functional companions, etc.)
- know what file and directory structure is required
- know what registry settings must be created by the installation

### 8.2.2.1 Oracle SellingPoint Application Files and File Structure

The Oracle SellingPoint application adheres to a particular default file structure. In addition, development may have included extending your Oracle SellingPoint application with bitmaps, functional companions, etc. The default file structure is:

`%ORACLE_HOME%\OSP\OSP` (contains required core OC files)
`%ORACLE_HOME%\OSP\shared` (contains core subdirectories for shared files)
`%ORACLE_HOME%\OSP\shared\ActiveMedia` (contains shared bitmap files)
`%ORACLE_HOME%\OSP\shared\ActiveModel` (contains shared.lce files)
`%ORACLE_HOME%\OSP\shared\Database` (contains shared database files)

Oracle Configurator Developer and WebUI files are not needed for a production deployment.

During development, the following information is inserted into the Oracle Configurator schema that has dependencies with the local file system of the machine running Oracle Configurator Developer:

- Bitmaps for logos on the home and summary screens referenced by the default Oracle SellingPoint application if not in default location (see Section , "HLOGO" on page 2-19 and Section , "SLOGO" on page 2-19). Non-default locations must be specified in the spx.ini file.
- Bitmaps referenced by a customized UI
- Functional Companions referenced by the Active Model

When deploying a customized Oracle SellingPoint application, install your files in adherence to the default file structure. Install local databases in `%ORACLE_HOME%\OSP\shared\Database` and Functional Companions anywhere in your classpath.

### 8.2.2.2 Oracle SellingPoint Application Proposals

In the Oracle SellingPoint application, the Create Proposal command lets you run the Proposal Wizard to create proposals of your quotes. Oracle SellingPoint application Proposals are generated in Microsoft Word 97 (8.0). The Oracle SellingPoint application provides a default test template for generating the proposal document. OC users can easily create new templates. There are however some administrative tasks associated with providing templates to the Proposal Wizard.

#### Naming Proposals

Oracle SellingPoint application provides one test template called Test Proposal 1. Test Proposal 1 and any proposal template you use with Oracle SellingPoint application contains specifically formatted text that imports other Word documents to populate the sections of the proposal.

The document title that you give the template, as specified in File --> Preferences in Word, is then displayed in the list of available templates in the Proposal Wizard.

#### Location of Proposals

Oracle SellingPoint application provides one test template called Test Proposal 1. It is located in the `%ORACLE_HOME%\OSP\Proposal\Templates\` directory of your install directory. Test Proposal 1 and any proposal template you use with Oracle SellingPoint application contains specifically formatted text that imports other Word documents to populate the sections of the proposal.

The templates for formatting Quote Reports documents inserted into proposals are located in the `%ORACLE_HOME%\OSP\Proposal\ReportTemplates\` directory of your install directory.

#### 8.2.2.3 Installing an Oracle SellingPoint Application

To install your customized Oracle SellingPoint application:

1. Have an Oracle Configurator schema populated with what you need for your Oracle SellingPoint application.
2. Run `OCRuntime.exe` from the Oracle Configurator Developer CD to install only the Oracle SellingPoint application on a deployment machine.

   The OC Installer will have created the files and file structure you need for proposal, quote output, and bitmaps, etc.
3. Install custom files such as bitmaps, local databases, and Functional Companions in the above recommended structure.
4. Establish data connectivity (DSN) to the Oracle Configurator schema, either local or networked. See Section 2.8.3, "Create DSNs and DBOwners" on page 2-11 for information about establishing data connectivity.
5. Edit the spx.ini file to point to the appropriate DSN. See Section 3.2, "Oracle Configurator Schema Settings" on page 3-4 for information about editing and setting parameters in the `spx.ini` file.

When you start up the Oracle SellingPoint application and log in to your Oracle Configurator schema as the DBOwner or a user listed in the CZ_END_USERS table, you have access to all the OC files needed.

### 8.2.3 Client/Server Deployment

For client/server (networked) deployment of the Oracle SellingPoint application (including test networked deployments), the Oracle SellingPoint application is installed from CD. If any customization has occurred, meaning the Oracle SellingPoint application is not the default application generated from Oracle Configurator Developer, use a project-specific installation program.

Each installed Oracle SellingPoint application runs on a client machine where an `spx.ini` file provides parameter settings and the end users who have access to the application are associated with particular projects and customers. The Oracle Configurator schema providing the data to run the Oracle SellingPoint application is commonly on a different (server) machine.

For information specifically about mobile and web deployments, see Section 8.3, "Oracle SellingPoint Mobile Deployment" on page 8-9and Section 8.1, "Oracle Configurator Window" on page 8-1.

### 8.2.3.1 Requirements for Client/Server Deployment

The requirements for a deployed client/server setup of the Oracle SellingPoint application are:

- The Oracle Configurator schema is running on the server providing the client with data needed by the runtime Oracle SellingPoint application.
- Oracle8 Client is running on the client machine running the Oracle SellingPoint application.
- The client is configured to connect to the Oracle Configurator schema (Oracle Net8 Easy Config). See Section 2.8.2, "Enable the Client for Database Connectivity" on page 2-10.
- A datasource name (DSN) for the Oracle Configurator schema is defined in the ODBC Administrator on the client. See Section 2.8.3, "Create DSNs and DBOwners" on page 2-11.
- The `spx.ini` file on the client is edited to include the correct DSN and DBOwner for the Oracle Configurator schema. See Section 2.8.6, "Parameters in spx.ini for Deployment" on page 2-21.
- The end user logged into the Oracle SellingPoint application is an end user included in the CZ_END_USERS table of the Oracle Configurator schema.
- The end user logged into the Oracle SellingPoint application is associated with one Project (Active Model) and at least one account.
- The end user logged into the client is a user defined in the Oracle Applications database.

## 8.3 Oracle SellingPoint Mobile Deployment

Mobile deployment is possible with Release 11*i*, but requires consulting services and installation of the Oracle SellingPoint application.

In a mobile deployment, the Oracle SellingPoint application runs on a laptop computer, using a local replica of the Oracle Configurator schema called the Oracle Configurator Mobile Database. In order to set up for mobile deployment, including testing mobile deployment, you must first prepare the server and the laptop for replication of the Oracle Configurator schema, and then create the replica. You must also prepare the server and the laptop so that the Oracle SellingPoint application end user can periodically synchronize data between the Oracle Configurator schema and the Oracle Configurator Mobile Database.

Oracle does not support a mobile development environment running Oracle Configurator Developer. However, it is possible to develop some demonstrations and prototypes using Oracle Configurator Developer in a mobile environment.

The standard supported method of mobile deployment is network-based using SQL*Net over a LAN. The Oracle Configurator also provides minimal support for file-based replication. See Section 8.3.7, "File-Based Replication" on page 8-23 for information about implementing file-based replication.

See Figure 1–2, "Deployment Installations" on page 1-10 for an illustration of the mobile deployment architecture.

You must complete the following initial tasks when you install your Oracle SellingPoint application. These tasks are necessary to prepare for a mobile deployment on a mobile laptop machine:

- Install your local replica database in `%ORACLE_HOME%\OSP\shared\Database` or establish a replica of the Oracle Configurator schema from the Oracle SellingPoint application, i.e., the Oracle Configurator Mobile Database. See Section 8.3.5, "Create the Replica Database" on page 8-18 for information about creating a replica database.
- Set up a DSN to point to the local replica database.
- Ensure that the end user logged into the Oracle SellingPoint application is associated with one Project (Active Model) and at least one Customer.

You must be aware of the following when developing your Oracle SellingPoint application for mobile deployment:

- When establishing an instance of the Oracle Configurator schema, the Oracle Applications System Administrator customizes the DBOwner, access roles, and privileges.
- You must insert a default <SPX_USER> record in the CZ_END_USERS table of the Oracle Configurator schema. Data transfer from Oracle Applications Release 11*i* does not populate the Oracle Configurator schema CZ_END_USERS table with the Oracle Applications user names and passwords. Generic import populates this table provided the CZ_DB_SETTING AUTOCREATE_ IMPORTED_USERS is non-Null.

  This enables you to log into the Oracle SellingPoint application to:

  - create more users (optional)
  - assign customers to users (required)
  - assign projects to users (required)

  This is done in the Tools -> Administration menu.

- You must set CZ_DB_SETTINGS for Import.
- You must set CZ_DB_SETTINGS for Export.
- You must set CZ_DB_SETTINGS for Replication.

### 8.3.1 Replication Method

The Oracle Configurator mobile deployment uses snapshot replication to copy model, configuration, and opportunity management data between the Oracle8*i* Enterprise Edition server database and a mobile Oracle8*i* Lite database. Oracle8*i* Lite integrates with the Oracle8*i* Enterprise Edition server's log-based replication facilities using snapshot replication only, not peer-to-peer. Oracle Configurator uses read-only snapshots to replicate the model information that is needed by the runtime Oracle SellingPoint application. This data effectively replicates "one-way", that is, only from the server to the replica, and the replica database does not allow modification to this data. Actual customer configurations, and other opportunity management data such as customer information, quotes, and proposals, are replicated both ways using read-write snapshots. Oracle Configurator replication, however, does not download the entirety of this data, but only those records to which the user has access: the user's customers, opportunities, and configurations.

Oracle's replication facility is based on using database logs that precisely describe each modification to a replicated table. While conceptually identical, the replication logs are actually redundant to the low-level DBMS logs (the before-image and after-image journals) and are stored in standard server tables. The server is first configured by executing replication API calls to set up replication logging on the desired tables. Each laptop replica is then initialized using a special Oracle8*i* Lite replication component. The runtime Oracle SellingPoint application creates an empty Oracle8*i* Lite database, then uses the replication control to populate it with data using read-only and read-write snapshots.

Some additional database operations populate the Oracle8*i* Lite database with fixed, reference or configuration data. Later replication activities refresh the data between the laptop replica and server using the same replication control. This operates by uploading the replica's logs of modifications on its read-write tables to the server and applying those modifications to the server database. Similarly, the replication logs for the server are downloaded and those modifications are applied to the replica. When the replica activity is uploaded and applied to the server, the server replication tools may detect that a record change on the replica conflicts with a server change to the same record. In this case, the replica modification is discarded and the conflict is logged for later review by the user.

The log-based replication algorithm provides a strong guarantee of data integrity because it applies exactly the same data modifications in the same sequence to the target database as were applied on the source database. However, it comes at the expense of transferring large amounts of data that may be redundant or unnecessary. Each operation includes complete before-and-after information: each update is logged with before and after copies of the updated record, and deletes are logged with the full contents of the record before it was deleted. So, if a record is updated twice on the server, then deleted, the next refresh may send as many as five copies of the record:

- the original copy
- the result from the first update
- the updated record
- the results of the second update
- the last contents of the record

After all that, the record ends up just being removed from the database.

## 8.3.2 Requirements for Mobile Deployment

The following list summarizes the requirements for a standard mobile deployment of the Oracle SellingPoint application using SQL*Net.

- The Oracle Configurator schema must be running on the server to perform either initial database replication or periodic data synchronization between the server and the mobile client.
- The Replication setting is enabled in the CZ_DB_SETTINGS table in the Oracle Configurator schema. See Table 3–1, "CZ_DB_SETTINGS" on page 3-5. This is done automatically when you run rep_setup.sql.
- Oracle8*i* Lite is running on the laptop.
- Oracle8 Client is running on the client laptop machine.

- The laptop is configured to connect to the Oracle Configurator schema on the server machine (Oracle Net8 Easy Config). See Section 2.8.2, "Enable the Client for Database Connectivity" on page 2-10.
- The replica Oracle Configurator Mobile Database is running on the laptop.
- The laptop is configured so that the Oracle SellingPoint application uses the Oracle Configurator Mobile Database.
- A datasource name (DSN) for the Oracle Configurator Mobile Database on the laptop is defined in ODBC Administrator. This is done automatically when you run the Initialize Remote DB command in the Oracle SellingPoint application running on the laptop.
- The `spx.ini` file on the laptop is edited to include the correct DSN and DBOwner for the Oracle Configurator Mobile Database (the replica). This is done automatically by the Initialize Remote DB command in the Oracle SellingPoint application.
- The end user logged onto the laptop is an end user included in the CZ_END_USERS table of both the Oracle Configurator schema and Oracle Configurator Mobile Database.
- The end user logged onto the laptop is a user defined in Oracle8*i* Enterprise Edition running the Oracle Configurator schema on the server.
- If the Oracle SellingPoint application is being integrated with Oracle Applications, the end user logged onto the laptop is a user defined in the Oracle Applications database.

## 8.3.3 Prepare the Server for Database Replication

Oracle provides several scripts to aid you in preparing the server system to replicate the Oracle Configurator schema. These scripts are available on the Oracle Configurator Developer CD in the DBAdmin folder. Using SQL*Plus (not SQL Worksheet), run the scripts from the CD or the directory into which you have copied them (<OC-scripts>). The state you achieve by running these scripts is summarized in Section 8.3.3.4, "Database Replication: Server Checklist" on page 8-17.

### 8.3.3.1 Prerequisites for Running the Replication Setup Scripts

The following requirements must be met before you run the replication setup scripts:

- You must have DBA privileges to set up the server machine.

- Oracle8*i* Enterprise Edition must be installed on the server with the advanced replication option on. The advance replication support should be installed before the Oracle Configurator schema objects are installed. All other replication setup can be done independently of creating the Oracle Configurator schema.
- Oracle8*i* Enterprise Edition must be running on the server.
- You must have a DBOwner in the Oracle Configurator schema instance that has access to all tables and data required for the deployed Oracle SellingPoint application.
- The Oracle Configurator schema must be accessible from the machine where you are running the scripts.

#### 8.3.3.2 Replication Setup Scripts

The server setup scripts should be run in the order listed.

- **rep_prop_setup.sql** creates the <repadmin> and registers that user as propagator, if such a user does not already exist.
- **rep_grants.sql** grants appropriate privileges to <SPX_USER>, the database role for end users of the Oracle SellingPoint application.
- **rep_setup.sql** performs most of the tasks required to set up the server.
- **RepAppsIntegrate.sql** recreates tables and replication objects that are dropped as a result of running InstAppsIntegrate.sql or InstAppsIntegrateViaLink.sql. If InstAppsIntegrate.sql or InsAppsIntegrateViaLink.sql is run before rep_ setup.sql is run, this script must be run after rep_setup.sql. If rep_setup.sql is run before InstAppsIntegrate.sql or InstAppsIntegrateViaLink.sql, this script must be run after the InstAppsIntegrate script.
- **rep_admin.sql** sets permissions for Oracle SellingPoint application end users who perform data synchronization between the mobile client and the server.

The next section, Section 8.3.3.3, "Run the OC Replication Setup Scripts" on page 8-14, provides instructions for running these scripts.

#### 8.3.3.3 Run the OC Replication Setup Scripts

The OC replication setup scripts prompt for numerous parameters. Oracle recommends that you read through the instructions to prepare for the values of those parameters before you begin running the scripts.

Set up your Oracle Configurator schema to support replication as follows:

1. Connect as <dba> to the database instance <ocsid> where you are setting up the production Oracle Configurator schema to support replication (see Section 1.8.1.1, "Connect to a Database Instance" on page 1-14).
2. Run the scripts in <OC-scripts> directory using SQL*Plus, not SQL Worksheet (see Section 1.8.1.3, "Run SQL*Plus in the <OC-scripts> Directory" on page 1-15).
3. Run rep_prop_setup.sql.

   This script creates the replication administration user and registers that user as propagator. If you already have a user registered as propagator, running this script overwrites that user. You can use Oracle Replication Manager to determine if you have a user registered as propagator.

   Example:

   ```
   SQL> @rep_prop_setup
   ```

   Rep_prop_setup.sql prompts you for the following parameters:

   a. <repadmin>, the name of the replication administration user
   b. <repadminpass>, the password of the <repadmin>
   c. <defaultspace>, the default tablespace name
   d. <tempspace>, the temporary tablespace name
   e. <TNSalias>, full Oracle service name to which you connect for database access
4. Run rep_grants.sql.

   This script prompts for the name of the user who owns the Oracle Configurator schema (<oc>). The script then grants appropriate privileges to <SPX_USER>, the database role for end users of the Oracle SellingPoint application.

   Example:

   ```
   SQL> @rep_grants
   ```

   The <SPX_USER> role is present in the Oracle Configurator schema. If running rep_grants.sql results in the error `‘DB role for SellingPoint users is missing!’`, you must create the <SPX_USER> role manually.

5. Run rep_setup.sql

   This is the main script to set up the server, and can take several minutes to run. It creates triggers and adds other objects to the server database with which the replica will communicate.

   Example:

   ```
   SQL> @rep_setup
   ```

   This script prompts you for the following parameters.

   a. <TNSalias>, the Oracle service name (the TNS alias)
   b. <DBAUser>, the user name of the SellingPoint administrator, who has DBA privileges
   c. <DBAUserpass>, the password for <DBAUser>
   d. <repadmin>, the replication administration user, a user with access to the packages dbms_repcat, etc.
   e. <repadminpass> the password for <repadmin>
   f. <oc>, the Oracle Configurator schema (user) name
   g. <repgroup>, the name of the replication group in the Oracle Configurator schema.

   The rep_setup.sql script prepares the Oracle Configurator schema <oc> for replication. The rep_setup.sql script changes several settings in the CZ_DB_SETTINGS to ‘YES” (see Section 8.3.3.4, "Database Replication: Server Checklist" on page 8-17).

   If you need to create replicas of other schemas, you must run rep_setup.sql for each one.

   The rep_setup.sql script logs you out when it completes.

6. Run rep_admin.sql

   This script sets appropriate permissions for those end users who use the Oracle SellingPoint application on a laptop and need to perform data replication between the mobile client and the server.

   Example:

```
SQL> @rep_admin
```

This script prompts for the end users username (listed in CZ_END_USERS table) and the DBOwner schema name of the Oracle Configurator schema where that end user is listed in the CZ_END_USERS table. If you provide a ‘%’ instead of the end user username, the script grants permissions to all users in the CZ_END_USERS table.

#### 8.3.3.4 Database Replication: Server Checklist

The following list summarizes the state you must achieve on the server prior to creating the replica database. The scripts described in the previous sections perform this setup for you.

- The Oracle Configurator schema is accessible from the server.
- The Replication setting in the CZ_DB_SETTINGS table is enabled. See Table 3–1, "CZ_DB_SETTINGS" on page 3-5. This setting is enabled by rep_setup.sql.
- Several rows are added to the CZ_DB_SETTINGS table.

  The following CZ_DB_SETTINGS are changed by rep_setup.sql for network-based replication.

  - Replication
  - RepConType
  - RepConInfo
  - RepTimeout
  - RepOliteDriver
  - RepOliteVersion

  See Section 3.2, "Oracle Configurator Schema Settings" on page 3-4 for more information about these settings.

### 8.3.4 Prepare the Client for Replication

You must also prepare the client laptop to create the replica database. This section describes the required actions.

1. Install Oracle8*i* Lite on the client laptop with JRE 1.1.1, Replication, Runtime, Utilities, and Open Client Adapter for ODBC installed. You must install Oracle8*i* Lite on the client before you install the Oracle SellingPoint application.
2. Install Oracle8 Client on the client laptop.
3. Use Oracle Net8 Easy Config to configure the laptop to connect to the Oracle Configurator schema on the server machine. See Section 2.8.2, "Enable the Client for Database Connectivity" on page 2-10.
4. Set up the client machine with an ODBC DSN for <ocsid>, as described in Section 2.8.3, "Create DSNs and DBOwners" on page 2-11.
5. Install your Oracle SellingPoint application.

   See the *Oracle Configurator and SellingPoint ReadMe* and *Oracle Configurator and SellingPoint Release Notes* (on the Oracle Configurator Developer CD) for information about installing your Oracle SellingPoint application.

6. You must verify that the logic has been generated (Generate Active Model command in Oracle Configurator Developer) for the Project to which the Oracle SellingPoint application end users need access.You must also verify that the UI has been generated (Generate Active UI command in Oracle Configurator Developer) for the Project to which the Oracle SellingPoint application end users need access.

### 8.3.5 Create the Replica Database

The replica database is the Oracle Configurator Mobile Database on the laptop. Use the following procedure to create the Oracle Configurator Mobile Database:

1. Start the Oracle SellingPoint application, logged in as a user with administrative privileges (e.g., <oc>). Using the Tools>Administration menu in the Oracle SellingPoint application, assign Customers and Projects to an appropriate end user. See Section 8.2.1, "Oracle SellingPoint Application User Access" on page 8-3 for more information.
2. On your client machine, connect to the server machine <ocdbhost> using either a network or dial-up connection. Start your Oracle SellingPoint application on the client machine. Log in as yourself on <oc>.
3. Go to the Tools > Administration menu and select Initialize Remote DB. This creates an Oracle8*i* Lite replica of <oc> on your client machine, creates an ODBC DSN for <oc>_replica, and puts the DSN entry in the `spx.ini` file.

The command Initialize Remote DB creates the Oracle Configurator Mobile Database as a replica of the server Oracle Configurator schema. You must be running Oracle8 Client to perform this task, and you must perform this task on each laptop where Oracle SellingPoint application is installed.

4. Exit your Oracle SellingPoint application.

Initialize Remote DB modifies the `spx.ini` file by adding the DSN for the <oc>_ replica as follows.

- The Data Source Name for <oc> on <ocsid) is added to the `[DSN]` section.

  Example:

  ```
  [DSN]
  <ocsid>
  ```

- Parameters for the replica and the schema are added to the `[MDADSNS]` section.

  Example:

  ```
  [MDADSNS]
  <oc>_replica
  <ocsid>
  ```

- The schema section is added.

  ```
  [ocsid]
  DBowner = oc
  ```

- The replica section is added.

  ```
  [<oc>_replica]
  DBOwner=<oc>
  master_name=<oc>
  master_schema_name=<oc>
  connection_type=0
  connection_info=
  ReplicationRefreshMode=OPTIMUM
  timeout=600
  olite_version_enum=3
  last_replication=Friday, September 10, 1999 at 3:31:53 PM
  ```

At this point, creation of the replica database is complete, and the laptop is ready for use by a mobile end user. See Section 2.8.5.7, "[<DSN>_replica]" on page 2-21 for detailed information about the `spx.ini` parameters in this section. The procedure described in this section must be repeated on each laptop used in mobile deployment. The mobile end users synchronize work on the laptop with the server database (<oc>), as described in Section 8.3.6, "Data Synchronization" on page 8-21.

### 8.3.5.1 Install the Replica Database on Multiple Laptops

Although Oracle recommends that you create a replica database on each laptop that will be accessing the server Oracle Configurator schema, this may not be logistically possible. You can create a primary replica and each end user can then install a copy of that primary replica on their laptop, access and modify it, and proceed with synchronizing their data with the Oracle Configurator schema on the server.

The procedure for creating a primary replica and installing a copy of the primary replica on a laptop is as follows:

1. Create a replica database as instructed in Section 8.3.5, "Create the Replica Database" on page 8-18.
2. Create a directory containing copies of the <replica>.odb, <replica>.opw, and `spx.ini` files (output from creating the primary replica) for the end users.
3. Create a file containing instructions for installing a copy of a replica. Be sure to include:
   a. Copy the <replica>.odb and <replica>.opw to `<oracle_home>\%OSP\shared\Database\`.
   b. Make a backup copy of your Oracle Configurator initialization file, `spx.ini`, which is located in `c:\windows\` (for windows 95) or `c:\winnt\` (for windows NT).
   c. Copy the provided `spx.ini` file to replace your previous version. (Users can switch between these files depending on how they want to connect to Oracle Configurator.)
   d. Create a new datasource for the replica (<replica>.odb). To do this you need to add a new system DSN, using the Oracle8*i* Lite client driver. See Section 2.8.2, "Enable the Client for Database Connectivity" on page 2-10 and Section 2.8.3, "Create DSNs and DBOwners" on page 2-11 for the specific instructions you need to include here.
   e. Modify the new `spx.ini` file with the new DSN pointing to the local replica.
4. Send the instructions and necessary files to each end user.

## 8.3.6 Data Synchronization

The Oracle SellingPoint application end user must synchronize the data in the production Oracle Configurator Mobile Database on a laptop running Oracle8*i* Lite with a production or maintenance Oracle Configurator schema on the server running Oracle8*i* Enterprise Edition.

Depending on the geographic range of the Oracle SellingPoint application deployment, it may be necessary to establish more than one server database with which Oracle Configurator Mobile Databases synchronize. Multiple server databases, in turn, must be synchronized with one another.

### 8.3.6.1 Synchronize

Follow these steps to synchronize data.

1. On your client machine where you created the Oracle Configurator Mobile Database (<oc>_replica), connect to the server machine <ocdbhost> using either a network or dial-up connection. This connection cannot be made directly from the Oracle SellingPoint application.
2. Start your Oracle SellingPoint application, logging on to <oc>_replica as the application end user.
3. Select the Replicate Now button on the Home screen or choose Replicate from the Tools menu.

   This uploads new data from <oc>_replica to the server, and downloads new data on the server to <oc>_replica.

This procedure does not synchronize data in a production Oracle Configurator schema with an Oracle Applications database. You must use the data transfer mechanisms described in Chapter 4, "Data Transfer to the CZ Schema" and Chapter 5, "Data Transfer from the CZ Schema" to keep your server Oracle Configurator schema data up to date with the Oracle Applications database. If integration with Oracle Applications is a factor in your deployment, you should verify that Order Transfers to and from the Oracle Configurator schema are operational.

Any synchronization failures are logged in a local log file. By default, these failures are logged in the `%ORACLE_HOME%/OSP/OSP/Oracle SellingPoint.log` file along with other Oracle SellingPoint application errors, where `%ORACLE_HOME%/` is the default installation directory for your Oracle SellingPoint application. You can view the `Oracle SellingPoint.log` file in the Oracle SellingPoint application by selecting the Show Log button in the Tools > Options menu. The only way you can change this local filename is by modifying or adding the `LOGFILE` parameter in the `spx.ini` file.

Customer/Contact/Address data are replicated as per the account or Project assigned by the replication administrator.

Opportunities/Configurations/Quotes/Proposals data are replicated only for the user who created the replica.

If the same data has been changed on both the Server machine and the Client machine, the change made on the Server machine always supersedes the change made on the client machine.

Report files for Quote and Proposal are not replicated.

Image files, such as .gif files, are not replicated.

### 8.3.6.2 Test Synchronization

#### Test One Replica and One Server

Test each of the following scenarios.

- Test synchronizing a replica with changes only on the replica.
- Test synchronizing a replica with changes only on the server.
- Test synchronizing a replica with changes on both the replica and the server where the changes are disjoint, that is they don't affect the same records.
- Test synchronizing a replica with changes on both the replica and the server where some changes affect the same records.

#### Test Multiple Replicas

To test replica regeneration, add data to a replica, synchronize with the server, delete the replica and create a new replica.

Create two or more replicas as different users, update the same data across all the replicas, and synchronize all the replicas.

To verify that export is working from a client, submit an order in the Oracle SellingPoint application, replicate to the server, and check the order submission in Order Management. See Section 5.5, "Verify Transfer from CZ Schema" on page 5-12 for more information.

## 8.3.7 File-Based Replication

Network-based replication using SQL*Net over a LAN is the default method used for synchronizing data between the Oracle SellingPoint application and the Oracle Configurator schema. File-based replication is a method that may be used to improve performance of synchronization over a dial-up connection. It uses Oracle Mobile Agents (OMA) to batch replication activity and use slower dial-up connections more efficiently.

In file-based replication, the Oracle SellingPoint application uses the client replication (an OCX) in much the same way as direct SQL*Net replication, but instead of communicating with the Oracle Configurator schema server, it communicates with an OMA server. The OMA replication protocol operates asynchronously when possible, thus making it considerably faster than SQL*Net over a dial-up connection.

**Note:** The increased efficiency of this file-based replication method is only worthwhile if the Oracle SellingPoint Client has a slower connection to the Oracle Configurator schema server than the OMA server does. Therefore, if end users synchronize daily or weekly by going in to the office and running over a LAN, they may as well use direct SQL*Net.

### 8.3.7.1 Setting Up File-based Replication

To set up file-based replication:

1. Be sure the requirements for mobile deployment, described in Section 8.3.2, "Requirements for Mobile Deployment" on page 8-12, have been met.
2. Prepare the server machine for data replication as described in Section 8.3.3, "Prepare the Server for Database Replication" on page 8-13, except change the CZ_DB_SETTINGS for file-based replication as described in Table 3–1, "CZ_DB_SETTINGS" on page 3-5 for the following settings:
   - Replication
   - RepConType
   - RepConInfo
   - RepTimeout
   - RepOliteDriver
   - RepOliteVersion

3. Install the OMA Message Gateway on the server. This is very site-specific. The gateway is typically installed on a separate machine from the databases(s) it is servicing. It requires an Oracle8*i* Enterprise Edition database for managing messages. This database must be created before installing the gateway. See the *Oracle Mobile Agents Installation and Configuration Guide* for more information about installing and configuring the Oracle Mobile Agents product and its components.

   At minimum during installation, use PPP to dial up and set up the LAN driver. This requires a server with a static IP address and UDP support.

   Create at least one gateway user for the Oracle SellingPoint application. (For more secure sites each user may have a separate gateway user. This is different from the Oracle database user.)

   For example, use the mgwctl command-line tool (Unix or NT) to create a gateway user for replication:

   ```
   mgwctl> add user SpRepUser password SpRepUser
   ```

   Then start the gateway:

   ```
   mgwctl> startup
   ```

   On Windows NT, use the services tab of the settings panel to set the message gateway service to start up automatically (this may have been an option during setup). On UNIX, you may need to alter the Oracle dbstart/dbstop scripts to start the gateway deamon on boot and stop it on shutdown.

4. Install Oracle Mobile Agent Controller on the machine where the agent(s) will be running. This is used to configure the replication agent to communicate with the message gateway and end user nodes. Because the replication agent only runs on Windows NT, you must install the OMA Controller on a Windows NT machine.

5. Install the Oracle8*i* Lite replication agent on an NT server.

   > **Note:** This can be different from the database and gateway server, but for performance reasons, it is preferable to have the agent run on the same machine as the Oracle Configurator schema instance.

   Use the Agent Controller (agentadm.exe) to configure the agent:

- Network Tab:

  This is site-specific. If using the LAN-based driver, make sure the agent's address port doesn't conflict with the gateway's address port. By default the gateway port is 3287. Use a value like 3487.

- Configuration Tab:

  This is site-specific. You can use the default username of mxagent for the Agent Owner. If you don't use the mxagent user, you must create a user with the mgwctl tool and assign the CAN-RUN-AGENT capability.

- Parameter Tab:

  You must add the following parameters:

  ```
  access = write

  share_path = ORACLE_HOME\mobile\agents\repoma8
  ```

6. Prepare each client machine (for each end user) as described in Section 8.3.4, "Prepare the Client for Replication".

7. Install the 32-bit version of the OMA message manager and run the message manager configurator tool (MMCONF32.exe). From the Advanced menu, select New Installation. This will register a new node on the gateway server for the laptop.

   Configure the driver(s) section based on the settings from the message gateway. For example, if you're using the LAN driver, enter the IP address of the gateway server and the port address (default 3287). Enter the gateway user information supplied by your database administrator (for example, username: SpRepUser, password: SpRepUser for all users, and check the store password button). For description, use something like "SellingPoint Replication User" or your name.

   From the advanced menu, select Services > Edit and type:

   ```
   polo8.services.mobile.oracle.com
   ```

   for the services name.

   From the advanced menu, select Driver > LAN. For local address, check Determine Automatically, and enter a port address that doesn't conflict with other ports, such as 3387.

Before replicating, start the message manager (msgman32.exe).

8. Edit the `spx.ini` settings. Each replica database has its own `[DSN]` section in the `spx.ini` file. There can also be a default replication section for setting initialization parameters for replication. If there is not a default section, the parameters from the CZ_DB_SETTINGS table are used.

   The replication section of the `spx.ini` file must be used and can be configured as:

   ```
   [replication]
   connection_type=1
   connection_info=oma-network
   timeout=1200
   ```

   Once a replica database has been initialized, the connection type and the info parameters are set under the section specific to the replica DSN. Changes to the default replication section do not effect existing replicas. Changes must be made to the specific section.

   For example:

   ```
   [oc_replica]
   DbOwner=XYZCorp
   master_name=xyz_db
   master_schema_name=xyz
   connection_type=1
   connection_info=oma-network
   timeout=1200
   last_replication=Wednesday, October 1, 1999 at 6:20:51 PM
   ```

9. Create the replica database for each client as described in Section 8.3.5, "Create the Replica Database" on page 8-18.

10. Test synchronization. See Section 8.3.6.2, "Test Synchronization" on page 8-22 for guidance.

Once configured, file-based replication works just as network-based replication using SQL*Net does. The Oracle SellingPoint application automatically starts the message manager as a background job. If a message manager is already running, a message box stating that a message manager is running is displayed. The user should click ok to close the message box. There is no other effect.

If problems occur during replication that cause the replication refresh to fail, the Oracle8*i* Lite replication control may not properly release the database resources. You should exit the Oracle SellingPoint application and restart it.

For more information about Proposal, see the *Oracle SellingPoint Application Help* provided with the default Oracle SellingPoint application.

# A

# Import Tables

## A.1 Overview

This appendix contains the following information:

"List of Import Tables" in the Oracle Configurator schema

"Dependencies Among Import Tables"

"Import Tables", which describes all import table fields and the data they expect.

For a mapping of Oracle Applications database source tables to Oracle Configurator schema destination tables, see Table 4–6, "Oracle Applications Source and Destination Online Tables" on page 4-11.

## A.2 List of Import Tables

The import tables are listed in the order in which the concurrent programs and generic import procedure populates them.

CZ_IMP_ITEM_MASTER
CZ_IMP_INTL_TEXT
CZ_IMP_DEVL_PROJECT
CZ_IMP_PS_NODE
CZ_IMP_CUSTOMER
CZ_IMP_CUSTOMER_END_USER
CZ_IMP_ADDRESS
CZ_IMP_ADDRESS_USE
CZ_IMP_CONTACT
CZ_IMP_PRICE_GROUP
CZ_IMP_PRICE
CZ_IMP_END_USER

CZ_IMP_END_USER_GROUP
CZ_IMP_ITEM_PARENT (not currently used by Release 11*i* import procedures)
CZ_IMP_ITEM_PROPERTY_VALUE
CZ_IMP_ITEM_TYPE
CZ_IMP_ITEM_TYPE_PROPERTY
CZ_IMP_PROPERTY
CZ_IMP_USER_GROUP (not currently used by Release 11*i* import procedures)

## A.3 Dependencies Among Import Tables

Dependencies among import tables must be heeded especially when generically importing single tables. In Table A–1, below, "Foreign Surrogate Key" lists the column in the import table whose value is dependent on the table listed in "Depends on". For instance, the FSK_ITEMTYPE_1_1 or FSK_ITEMTYPE_1_EXT column in CZ_IMP_ITEM_MASTER gets its value from CZ_IMP_ITEM_TYPE and helps in key resolution. FSK_ITEMTYPE_1_1 (default) or FSK_ITEMTYPE_1_EXT are populated per some indicator (0 or 1) in CZ_XFR_TABLE.

> **Note:** Oracle recommends that the usage of FSK_***_EXT columns be very limited as they will not be supported in the near future.

A strong dependency means a value is required for a successful import of that record. If "Default" is YES, there is already a default value in that column and import will succeed even if the dependency is strong and no value is imported.

*Table A–1 Dependencies Among Oracle Configurator Schema Import Tables*

| Import Table Name | Depends on | for Foreign Surrogate Key | Type of dependencies | Default |
|---|---|---|---|---|
| CZ_IMP_CUSTOMER | CZ_IMP_PRICE_GROUP | FSK_PRICEGROUP | WEAK | NO |
| CZ_IMP_CUSTOMER_END_USER | CZ_IMP_CUSTOMER | FSK_CUSTOMER | STRONG | NO |
| CZ_IMP_CUSTOMER_END_USER | CZ_IMP_END_USER | FSK_ENDUSER | STRONG | NO |
| CZ_IMP_ADDRESS | CZ_IMP_CUSTOMER | FSK_CUSTOMER | STRONG | NO |
| CZ_IMP_ADDRESS_USE | CZ_IMP_ADDRESS | FSK_ADDRESS | STRONG | NO |

***Table A–1 Dependencies Among Oracle Configurator Schema Import Tables***

| | | | | |
|---|---|---|---|---|
| CZ_IMP_CONTACT | CZ_IMP_CUSTOMER | FSK_CUSTOMER | STRONG | NO |
| CZ_IMP_CONTACT | CZ_IMP_ADDRESS | FSK_ADDRESS | STRONG | NO |
| CZ_IMP_DEVL_PROJECT | CZ_IMP_INTL_TEXT | FSK_INTLTEXT | STRONG | NO |
| CZ_IMP_END_USER | CZ_IMP_USER_GROUP | FSK_USERGROUP | WEAK | NO |
| CZ_IMP_END_USER_GROUP | CZ_IMP_END_USER | FSK_ENDUSER | STRONG | NO |
| CZ_IMP_END_USER_GROUP | CZ_IMP_USER_GROUP | FSK_USERGROUP | STRONG | YES |
| CZ_IMP_INTL_TEXT | NO | NO | NO | NO |
| CZ_IMP_ITEM_MASTER | CZ_IMP_ITEM_TYPE | FSK_ITEM_TYPE | STRONG | YES |
| CZ_IMP_ITEM_PARENT | CZ_IMP_ITEM_MASTER | FSK_ITEMMASTER | STRONG | NO |
| CZ_IMP_ITEM_PROPERTY_VALUE | CZ_IMP_PROPERTY | FSK_PROPERTY | STRONG | NO |
| CZ_IMP_ITEM_PROPERTY_VALUE | CZ_IMP_ITEM_MASTER | FSK_ITEMMASTER | STRONG | NO |
| CZ_IMP_ITEM_TYPE | NO | NO | NO | NO |
| CZ_IMP_ITEM_TYPE_PROPERTY | CZ_IMP_ITEM_TYPE | FSK_ITEMTYPE | STRONG | NO |
| CZ_IMP_ITEM_TYPE_PROPERTY | CZ_IMP_PROPERTY | FSK_PROPERTY | STRONG | NO |
| CZ_IMP_PRICE | CZ_IMP_ITEM_MASTER | FSK_ITEMMASTER | STRONG | NO |
| CZ_IMP_PRICE | CZ_IMP_PRICE_GROUP | FSK_PRICEGROUP | STRONG | NO |
| CZ_IMP_PRICE_GROUP | NO | NO | NO | NO |
| CZ_IMP_PROPERTY | NO | NO | NO | NO |
| CZ_IMP_PS_NODE | CZ_IMP_INTL_TEXT | FSK_INTLTEXT | STRONG | NO |
| CZ_IMP_PS_NODE | CZ_IMP_ITEM_MASTER | FSK_ITEMMASTER | STRONG | NO |
| CZ_IMP_PS_NODE | CZ_IMP_DEVL_PROJECT | FSK_DEVLPROJECT | STRONG | NO |
| CZ_IMP_USER_GROUP | NO | NO | NO | NO |

## A.4 Import Tables

Table A–2, "Import Table Field Disposition Codes", describes the disposition codes that may result when required columns are queried against the source table. Table A–3, "Import Table Record Status Codes", describes the record status codes for required columns that have not resulted in a successful query. Table A–4 through Table A–22, describe all of the columns in the Oracle Configurator schema import tables. Column order is not necessarily fixed. A column that is denoted as required in this table, means that the column is required in the source table for a successful generic import and is queried against a corresponding column in the target import table.

Disposition codes for the required fields are:

***Table A–2 Import Table Field Disposition Codes***

| Code | Disposition |
|---|---|
| M | marked for modification |
| I | marked for insertion |
| R | rejected |

Record status codes for required columns of the import tables are:

***Table A–3 Import Table Record Status Codes***

| Code | Status |
|---|---|
| PASS | marked for either modification or insertion after the key resolution stage |
| OK | modified / inserted into the target table |
| ERR | not modified / inserted into the target table because of an error in the transfer stage |
| DUPL | marked as duplicate |

*Table A–4 Description of Fields in CZ_IMP_CUSTOMER Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| CUSTOMER_ID | N | M | NUMBER | Designates the OC identifier for the account (mandatory null) |
| PRICE_LIST_ID | N | Y | NUMBER | Contains the OC price list ID for this account |
| WAREHOUSE_ID | N | Y | NUMBER | Contains the OC warehouse ID for this account |
| ORIG_SYS_REF | Y | N | VARCHAR2 (255) | Unique identification of a record in this table. Contains a foreign surrogate-key value matching CZ_ CUSTOMERS.ORIG_SYS_REF.<br>Disposition:<br>■ if found - M<br>■ if not found - I<br>■ if not unique - R-DUPL<br>■ if null - R-N8 |
| CUSTOMER_ NAME | N | Y | VARCHAR2 (150) | Contains name of the account |
| PARENT_ID | N | Y | NUMBER | Contains parent account ID for an account |
| DIVISION | N | Y | NUMBER | Contains division number for the account |
| NOTE | N | Y | VARCHAR2 (255) | If there is any specific note for the account |
| DESC_TEXT | N | Y | VARCHAR2 (255) | Describes this account |
| CUSTOMER_ STATUS | N | Y | VARCHAR2 (20) | Contain status of an account |
| REC_NBR | N | Y | NUMBER | Provides a one-up sequence number identifying the record in this run |
| RUN_ID | N | Y | NUMBER | Contains a number indicating the Import run in which this record was loaded / processed |
| REC_STATUS | N | Y | VARCHAR2 (4) | Records a coded status describing the import results for this record; if null it indicates the record has not yet been completely processed |
| DISPOSITION | N | Y | CHAR (1) | Indicates whether the record was inserted, updated, unchanged, or rejected |

*Table A–4 Description of Fields in CZ_IMP_CUSTOMER Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| DELETED_FLAG | N | Y | CHAR (1) | Indicates (’1’/ ’0’) that this record has been deleted |
| EFF_FROM | N | Y | DATE | Indicates the beginning date for which this record is effective |
| EFF_TO | N | Y | DATE | Indicates the ending date through which this record is effective |
| EFF_MASK | N | Y | VARCHAR2 (40) | Reserved for future use |
| USER_STR01 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_STR02 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_STR03 | N | Y | VARCHAR2 (255) | Textual user expansion field; may be used as an alternate ’surrogate key’ for the record |
| USER_STR04 | N | Y | VARCHAR2 (25) | Textual user expansion field |
| USER_NUM01 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM02 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM03 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM04 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| CHECKOUT_USER | N | Y | VARCHAR2 (40) | Reserved for future use in checkout user |
| CREATION_DATE | N | Y | DATE | Indicates the date on which this record was created |
| LAST_UPDATE_ DATE | N | Y | DATE | Contains the date on which the record was last modified |
| CREATED_BY | N | Y | NUMBER | Identifies the user that created this record |

*Table A–4 Description of Fields in CZ_IMP_CUSTOMER Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| LAST_UPDATED_BY | N | Y | NUMBER | Records the login name for the user that last modified this record |
| SECURITY_MASK | N | Y | VARCHAR2 (40) | Reserved for future use in record-level access control |
| FSK_PRICEGROUP_1_1 | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_IMP_PRICE_GROUP on NAME |
| FSK_PRICEGROUP_1_EXT | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_IMP_PRICE_GROUP on USER_STR03. Note: It is recommended that the usage of this column be very limited as it will not be supported in the near future |
| FSK_CUSTOMER_2_1 | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_IMP_CUSTOMER on ORIG_SYS_REF |
| FSK_CUSTOMER_2_EXT | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_IMP_CUSTOMER on USER_STR03 |

*Table A–5 Description of Fields in CZ_IMP_CUSTOMER_END_USER Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| CUSTOMER_ID | N | Y | NUMBER | Contains the OC account ID for an end user |
| END_USER_ID | N | Y | NUMBER | Contains the OC end user ID for an account |
| CREATION_DATE | N | Y | DATE | Indicates the date on which this record was created |
| LAST_UPDATE_DATE | N | Y | DATE | Contains the date on which the record was last modified |
| DELETED_FLAG | N | Y | CHAR (1) | Indicates ('1'/'0') that this record has been deleted |
| EFF_FROM | N | Y | DATE | Indicates the beginning date for which this record is effective |

*Table A–5 Description of Fields in CZ_IMP_CUSTOMER_END_USER Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| EFF_TO | N | Y | DATE | Indicates the ending date through which this record is effective |
| CREATED_BY | N | Y | NUMBER | Identifies the user that created this record |
| LAST_UPDATED_ BY | N | Y | NUMBER | Records the login name for the user that last modified this record |
| SECURITY_MASK | N | Y | VARCHAR2 (40) | Reserved for future use in record-level access control |
| EFF_MASK | N | Y | VARCHAR2 (40) | Reserved for future use |
| REC_NBR | N | Y | NUMBER | Provides a one-up sequence number identifying the record in this run |
| RUN_ID | N | Y | NUMBER | Contains a number indicating the Import run in which this record was loaded / processed |
| REC_STATUS | N | Y | VARCHAR2 (4) | Records a coded status describing the import results for this record; if null it indicates the record has not yet been completely processed |
| DISPOSITION | N | Y | CHAR (1) | Indicates whether the record was inserted, updated, unchanged, or rejected |
| CHECKOUT_USER | N | Y | VARCHAR2 (40) | Reserved for future use in checkout user |
| FSK_CUSTOMER_ 1_1 | Y | N | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ CUSTOMERS.ORIG_SYS_REF.<br>Disposition:<br>■ if found - assign customer ID<br>■ if not found - R-F13<br>■ if not unique - R-DUPL<br>■ if null - R-N13 |

**Table A–5 Description of Fields in CZ_IMP_CUSTOMER_END_USER Import Table**

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| FSK_ENDUSER_2_1 | Y | N | VARCHAR2(255) | Contains a foreign surrogate-key value matching CZ_END_USERS.LOGIN_NAME.<br>Disposition:<br>■ if found - assign end user ID<br>■ if not found - R-F15<br>■ if not unique - R-DUPL<br>■ if null - R-N15 |
| FSK_ENDUSER_2_EXT | N | Y | VARCHAR2(255) | Contains a foreign surrogate-key value matching CZ_IMP_END_USER on USER_STR03. Note: It is recommended that the usage of this column be very limited as it will not be supported in the near future |
| Assigned customer ID and end user ID | Y | | | Additional columns required in source table. Queried against CZ_CUSTOMER_END_USERS.CUSTOMER_ID and CZ_CUSTOMER_END_USERS.END_USER_ID.<br>Disposition:<br>■ if found - M<br>■ if not found - I |

**Table A–6 Description of Fields in CZ_IMP_ADDRESS Import Table**

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| ADDRESS_ID | N | Y | NUMBER | Designates the OC identifier for the address |
| CUSTOMER_ID | N | Y | NUMBER | Contains the OC account ID |
| COUNTRY | N | Y | VARCHAR2(240) | Contains country name for the address |
| ADDR_LINE1 | N | Y | VARCHAR2(240) | Contains first line of the address |
| ADDR_LINE2 | N | Y | VARCHAR2(240) | Contains second line of the address |

*Table A–6 Description of Fields in CZ_IMP_ADDRESS Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| CITY | N | Y | VARCHAR2 (60) | Contains city name of the address |
| POSTAL_CODE | N | Y | VARCHAR2 (60) | Contains postal code (ZIP) of the address |
| STATE | N | Y | VARCHAR2 (60) | Contains state name of the address |
| PROVINCE | N | Y | VARCHAR2 (60) | Contains province name of the address |
| COUNTY | N |  | VARCHAR2 (60) | Contains county name of the address |
| BILL_TO_FLAG | N | Y | CHAR (1) | Flag indicates if it can be bill to this address |
| SHIP_TO_FLAG | N | Y | CHAR (1) | Flag indicates if it can be ship to this address |
| ORIG_SYS_REF | Y | N | VARCHAR2 (255) | Unique identification of a record in this table. Contains a foreign surrogate-key value matchingCZ_ ADDRESSES.ORIG_SYS_REF.<br>Disposition:<br>■ if found - M<br>■ if not found - I<br>■ if not unique - R-DUPL<br>■ if null - potentially wrong results |
| REC_NBR | N | Y | NUMBER | Provides a one-up sequence number identifying the record in this run |
| RUN_ID | N | Y | NUMBER | Contains a number indicating the Import run in which this record was loaded / processed |
| REC_STATUS | N | Y | VARCHAR2 (4) | Records a coded status describing the import results for this record; if null it indicates the record has not yet been completely processed |
| DISPOSITION | N | Y | CHAR (1) | Indicates whether the record was inserted, updated, unchanged, or rejected |
| DELETED_FLAG | N | Y | CHAR (1) | Indicates ('1' / '0') that this record has been deleted |
| EFF_FROM | N | Y | DATE | Indicates the beginning date for which this record is effective |

*Table A–6 Description of Fields in CZ_IMP_ADDRESS Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| EFF_TO | N | Y | DATE | Indicates the ending date through which this record is effective |
| SECURITY_MASK | N | Y | VARCHAR2 (40) | Reserved for future use in record-level access control |
| EFF_MASK | N | Y | VARCHAR2 (40) | Reserved for future use |
| FSK_CUSTOMER_1_1 | Y | N | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_CUSTOMERS.ORIG_SYS_REF.<br>Disposition:<br>■ if found - assign customer ID<br>■ if not found - R-F28<br>■ if null - R-N28 |

*Table A–7 Description of Fields in CZ_IMP_ADDRESS_USE Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| ADDRESS_USE_ID | N | Y | NUMBER | Designates the OC identifier for the address use |
| ADDRESS_ID | N | Y | NUMBER | Contains the OC address ID for address use |
| ORIG_SYS_REF | Y | N | VARCHAR2 (255) | Unique identification of a record in this table. Contains a foreign surrogate-key value matching CZ_ADDRESSES_USES.ORIG_SYS_REF.<br>Disposition:<br>■ if found - M<br>■ if not found - I<br>■ if not unique - R-DUPL<br>■ if null - R-N7 |
| SITE_USE_CODE | N | Y | VARCHAR2 (20) | Contains code for site use |
| WAREHOUSE_ID | N | Y | NUMBER | Contains the OC warehouse ID for address use |

*Table A–7 Description of Fields in CZ_IMP_ADDRESS_USE Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| DELETED_FLAG | N | Y | CHAR (1) | Indicates ('1'/'0') that this record has been deleted |
| RUN_ID | N | Y | NUMBER | Contains a number indicating the Import run in which this record was loaded/processed |
| REC_STATUS | N | Y | VARCHAR2 (4) | Records a coded status describing the import results for this record; if null it indicates the record has not yet been completely processed |
| DISPOSITION | N | Y | CHAR (1) | Indicates whether the record was inserted, updated, unchanged, or rejected |
| FSK_ADDRESS_1_1 | Y | N | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ ADDRESS_USES.ORIG_SYS_REF.<br>Disposition:<br>■ if found - assign address ID<br>■ if not found - R-F5<br>■ if null - R-N5 |

*Table A–8 Description of Fields in CZ_IMP_CONTACT Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| CONTACT_ID | N | Y | NUMBER | Designates the OC identifier for this contact |
| CUSTOMER_ID | N | Y | NUMBER | Contains the OC account ID for this contact |
| ADDRESS_ID | N | Y | NUMBER | Contains the OC address ID for this contact |
| ORIG_SYS_REF | Y | N | VARCHAR2 (255) | Unique identification of a record in this table. Contains a foreign surrogate-key value matching CZ_ CONTACTS.ORIG_SYS_REF.<br>Disposition:<br>■ if found - M<br>■ if not found - I<br>■ if not unique - R-DUPL<br>■ if null - R-N17 |

*Table A–8 Description of Fields in CZ_IMP_CONTACT Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| SALUTATION | N | Y | VARCHAR2 (48) | Designates the appropriate salutation for the contact |
| FIRSTNAME | N | Y | VARCHAR2 (48) | Records the first name of the contact |
| MI | N | Y | CHAR (1) | Contains the middle initial of the contact |
| LASTNAME | N | Y | VARCHAR2 (48) | Contains the contact's last name |
| SUFFIX | N | Y | VARCHAR2 (24) | Contains any name suffix (Jr., III, Ph.D., etc.) that applies to this contact |
| TITLE | N | Y | VARCHAR2 (48) | Designates a title for referring to the contact |
| PHONE | N | Y | VARCHAR2 (48) | Contains the primary phone number for this contact |
| ALT_PHONE | N | Y | VARCHAR2 (48) | Records an alternate telephone number for the contact |
| FAX | N | Y | VARCHAR2 (48) | Records a fax number for the contact |
| PAGER | N | Y | VARCHAR2 (48) | Records a pager phone number for the contact |
| CELLULAR | N | Y | VARCHAR2 (48) | Contains the cellular / wireless telephone number for the contact |
| EMAIL_ADDR | N | Y | VARCHAR2 (150) | Records an e-mail address for the contact |
| NOTE | N | Y | VARCHAR2 (255) | Contains free-form notes about the contact |
| REC_NBR | N | Y | NUMBER | Provides a one-up sequence number identifying the record in this run |
| RUN_ID | N | Y | NUMBER | Contains a number indicating the Import run in which this record was loaded / processed |

*Table A–8 Description of Fields in CZ_IMP_CONTACT Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| REC_STATUS | N | Y | VARCHAR2 (4) | Records a coded status describing the import results for this record; if null it indicates the record has not yet been completely processed |
| DISPOSITION | N | Y | CHAR (1) | Indicates whether the record was inserted, updated, unchanged, or rejected |
| DELETED_FLAG | N | Y | CHAR (1) | Indicates ('1' / '0') that this record has been deleted |
| EFF_FROM | N | Y | DATE | Indicates the beginning date for which this record is effective |
| EFF_TO | N | Y | DATE | Indicates the ending date through which this record is effective |
| EFF_MASK | N | Y | VARCHAR2 (40) | Reserved for future use |
| USER_STR01 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_STR02 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_STR03 | N | Y | VARCHAR2 (255) | Textual user expansion field; may be used as an alternate 'surrogate key' for the record |
| USER_STR04 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_NUM01 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM02 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM03 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM04 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| CHECKOUT_USER | N | Y | VARCHAR2 (40) | Reserved for future use in checkout user |

*Table A–8 Description of Fields in CZ_IMP_CONTACT Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| PRIMARY_ROLE | N | Y | VARCHAR2 (60) | Contains the primary role of this contact |
| CREATION_DATE | N | Y | DATE | Indicates the date on which this record was created |
| LAST_UPDATE_ DATE | N | Y | DATE | Contains the date on which the record was last modified |
| CREATED_BY | N | Y | NUMBER | Identifies the user that created this record |
| LAST_UPDATED_ BY | N | Y | NUMBER | Records the login name for the user that last modified this record |
| SECURITY_MASK | N | Y | VARCHAR2 (40) | Reserved for future use in record-level access control |
| CONTACT_ HANDLE | N | Y | VARCHAR2 (255) | Contains the implementer's unique identifier for a contact |
| FSK_CUSTOMER_ 1_1 | Y | N | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ CUSTOMERS.ORIG_SYS_REF.<br>Disposition:<br>■ if found - assign customer ID<br>■ if not found - R-F41<br>■ if not unique - R-DUPL<br>■ if null - R-N41. |
| FSK_CUSTOMER_ 1_EXT | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ IMP_CUSTOMER on USER_STR03. Note: It is recommended that the usage of this column be very limited as it will not be supported in the near future |
| FSK_ADDRESS_2_1 | Y | N | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ ADDRESSES.ORIG_SYS_REF.<br>Disposition:<br>■ if found - assign address ID<br>■ if not found - R-F44<br>■ if not unique - R-DUPL<br>■ if null - R-N44 |

*Table A–9 Description of Fields in CZ_IMP_DEVL_PROJECT Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| DEVL_PROJECT_ ID | N | Y | NUMBER | Designates an OC identifier for the development project |
| INTL_TEXT_ID | N | Y | NUMBER | Contains the OC international text ID for this project |
| NAME | Y | N | VARCHAR2 (255) | Contains name of the development project. Disposition if null - ERR |
| GSL_FILENAME | N | Y | VARCHAR2 (255) | Contains gsl filename of the project |
| VERSION | N | Y | NUMBER | Contains version number of the project |
| DESC_TEXT | N | Y | VARCHAR2 (255) | Description text for this project |
| ORIG_SYS_REF | Y | N | VARCHAR2 (255) | Unique identification of a record in this table. Contains a foreign surrogate-key value matching CZ_DEVL_ PROJECTS.ORIG_SYS_REF.<br>Disposition:<br>■ if found - M<br>■ if not found - I<br>■ if not unique - R-DUPL<br>■ if null - R-N7 |
| CREATION_DATE | N | Y | DATE | Indicates the date on which this record was created |
| LAST_UPDATE_ DATE | N | Y | DATE | Contains the date on which the record was last modified |
| DELETED_FLAG | N | Y | CHAR (1) | Indicates ('1' / '0') that this record has been deleted |
| EFF_FROM | N | Y | DATE | Indicates the beginning date for which this record is effective |
| EFF_TO | N | Y | DATE | Indicates the ending date through which this record is effective |
| CREATED_BY | N | Y | NUMBER | Identifies the user that created this record |
| LAST_UPDATED_ BY | N | Y | NUMBER | Records the login name for the user that last modified this record |

*Table A–9 Description of Fields in CZ_IMP_DEVL_PROJECT Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| SECURITY_MASK | N | Y | VARCHAR2 (40) | Reserved for future use in record-level access control |
| EFF_MASK | N | Y | VARCHAR2 (40) | Reserved for future use |
| CHECKOUT_USER | N | Y | VARCHAR2 (40) | Reserved for future use in checkout user |
| RUN_ID | N | Y | NUMBER | Contains a number indicating the Import run in which this record was loaded / processed |
| REC_STATUS | N | Y | VARCHAR2 (4) | Records a coded status describing the import results for this record; if null it indicates the record has not yet been completely processed |
| DISPOSITION | N | Y | CHAR (1) | Indicates whether the record was inserted, updated, unchanged, or rejected |
| FSK_INTLTEXT_1_ 1 | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ IMP_INTL_TEXT on TEXT_STR |

*Table A–10 Description of Fields in CZ_IMP_END_USER Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| END_USER_ID | N | Y | NUMBER | Designates an OC identifier for the user |
| END_USER_ORG_ ID | N | Y | NUMBER | Indicates the organization to which this end user belongs |
| ORIG_SYS_REF | Y | N | VARCHAR2 (255) | Unique identification of a record in this table. Contains a foreign surrogate-key value matching CZ_END_ USERS.ORIG_SYS_REF.<br>Disposition:<br>■ if found - M<br>■ if not found - I<br>■ if not unique - R-DUPL<br>■ if null - R-N43 |
| TITLE | N | Y | VARCHAR2 (48) | Designates a title for referring to the end user |
| LOGIN_NAME | N | Y | VARCHAR2 (48) | Designates the name by which the end user logs in to the SellingPoint backbone data warehouse |
| LASTNAME | N | Y | VARCHAR2 (48) | Contains the end user’s last name |
| FIRSTNAME | N | Y | VARCHAR2 (48) | Records the first name of the user |
| MI | N | Y | CHAR (1) | Contains the middle initial of the end user |
| ALLOWABLE_ DISCOUNT | N | Y | NUMBER (16,9) | Percentage of discount can be allowed to this user |
| DESC_TEXT | N | Y | VARCHAR2 (255) | Describes this end user |
| ADDR_LINE1 | N | Y | VARCHAR2 (150) | Contains end user’s first line address |
| ADDR_LINE2 | N | Y | VARCHAR2 (150) | Contains end user’s second line address |
| CITY | N | Y | VARCHAR2 (56) | Contains end user’s city name |

*Table A–10 Description of Fields in CZ_IMP_END_USER Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| STATE | N | Y | VARCHAR2 (8) | Contains end user's state name |
| PROVINCE | N | Y | VARCHAR2 (56) | Contains end user's province name |
| COUNTY | N | Y | VARCHAR2 (56) | Contains end user's county name |
| ZIP | N | Y | VARCHAR2 (24) | Contains end user's zip code |
| COUNTRY | N | Y | VARCHAR2 (56) | Contains end user's country name |
| PHONE | N | Y | VARCHAR2 (48) | Contains end user's contact phone number |
| FAX | N | Y | VARCHAR2 (48) | Contains end user's FAX number |
| PAGER | N | Y | VARCHAR2 (48) | Contains end user's pager number |
| CELLULAR | N | Y | VARCHAR2 (48) | Contains end user's cellular phone number |
| EMAIL_ADDR | N | Y | VARCHAR2 (150) | Contains end user's e-mail ID |
| REC_NBR | N | Y | NUMBER | Provides a one-up sequence number identifying the record in this run |
| RUN_ID | N | Y | NUMBER | Contains a number indicating the Import run in which this record was loaded / processed |
| REC_STATUS | N | Y | VARCHAR2 (4) | Records a coded status describing the import results for this record; if null it indicates the record has not yet been completely processed |
| DISPOSITION | N | Y | CHAR (1) | Indicates whether the record was inserted, updated, unchanged, or rejected |
| DELETED_FLAG | N | Y | CHAR (1) | Indicates ('1' / '0') that this record has been deleted |

*Table A–10 Description of Fields in CZ_IMP_END_USER Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| EFF_FROM | N | Y | DATE | Indicates the beginning date for which this record is effective |
| EFF_TO | N | Y | DATE | Indicates the ending date through which this record is effective |
| EFF_MASK | N | Y | VARCHAR2 (40) | Reserved for future use |
| USER_STR01 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_STR02 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_STR03 | N | Y | VARCHAR2 (255) | Textual user expansion field; may be used as an alternate ’surrogate key’ for the record |
| USER_STR04 | N | Y | VARCHAR (255) | Textual user expansion field |
| USER_NUM01 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM02 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM03 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM04 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| CHECKOUT_USER | N | Y | VARCHAR2 (40) | Reserved for future use in checkout user |
| CREATION_DATE | N | Y | DATE | Indicates the date on which this record was created |
| LAST_UPDATE_ DATE | N | Y | DATE | Contains the date on which the record was last modified |
| CREATED_BY | N | Y | NUMBER | Identifies the user that created this record |
| LAST_UPDATED_ BY | N | Y | NUMBER | Records the login name for the user that last modified this record |

*Table A–10 Description of Fields in CZ_IMP_END_USER Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| SECURITY_MASK | N | Y | VARCHAR2 (40) | Reserved for future use in record-level access control |
| FSK_ENDUSER_1_ 1 | N | Y | VARCHAR2 (255) | Not currently used |
| FSK_ENDUSER_1_ EXT | N | Y | VARCHAR2 (255) | Not currently used |
| FSK_USERGROUP_ 1_1 | N | Y | VARCHAR2 (255) | Not currently used |
| FSK_USERGROUP_ 1_EXT | N | Y | VARCHAR2 (255) | Not currently used |
| NAME | N | Y | VARCHAR2 (255) | Contains name of the end user |

*Table A–11 Description of Fields in CZ_IMP_END_USER_GROUP Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| END_USER_ID | N | Y | NUMBER | Contains the OC ID of the associated user |
| USER_GROUP_ID | N | Y | NUMBER | Contains the OC Group ID that applies to this record |
| DATE_ADDED_ USER | N | Y | DATE | Indicates the date on which this user was added to the group |
| USER_ADDEDBY | N | Y | VARCHAR2 (48) | Records the user that added this user-group record |
| GROUP_PRIORITY | N | Y | CHAR (1) | Priority level of the group user |
| CHECKOUT_USER | N | Y | VARCHAR2 (40) | Reserved for future use in checkout user |

*Table A–11 Description of Fields in CZ_IMP_END_USER_GROUP Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| REC_NBR | N | Y | NUMBER | Provides a one-up sequence number identifying the record in this run |
| RUN_ID | N | Y | NUMBER | Contains a number indicating the Import run in which this record was loaded / processed |
| REC_STATUS | N | Y | VARCHAR2 (4) | Records a coded status describing the import results for this record; if null it indicates the record has not yet been completely processed |
| DISPOSITION | N | Y | CHAR (1) | Indicates whether the record was inserted, updated, unchanged, or rejected |
| DELETED_FLAG | N | Y | CHAR (1) | Indicates ('1' / '0') that this record has been deleted |
| EFF_FROM | N | Y | DATE | Indicates the beginning date for which this record is effective |
| EFF_TO | N | Y | DATE | Indicates the ending date through which this record is effective |
| EFF_MASK | N | Y | VARCHAR2 (40) | Reserved for future use |
| CREATION_DATE | N | Y | DATE | Indicates the date on which this record was created |
| LAST_UPDATE_ DATE | N | Y | DATE | Contains the date on which the record was last modified |
| CREATED_BY | N | Y | NUMBER | Identifies the user that created this record |
| LAST_UPDATED_ BY | N | Y | NUMBER | Records the login name for the user that last modified this record |
| SECURITY_MASK | N | Y | VARCHAR2 (40) | Reserved for future use in record-level access control |
| FSK_ENDUSER_1_ 1 | Y | N | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ END_USERS.ORIG_SYS_REF.<br>Disposition:<br>■ if found - assign end user ID<br>■ if not found - R-F20<br>■ if not unique - R-DUPL<br>■ if null - R-N20 |

*Table A–11 Description of Fields in CZ_IMP_END_USER_GROUP Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| FSK_ENDUSER_1_ EXT | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ IMP_END_USER on USER_STR03. Note: It is recommended that the usage of this column be very limited as it will not be supported in the near future |
| FSK_USERGROUP_ 2_1 | Y | N | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ USER_GROUPS.GROUP_NAME.<br>Disposition:<br>■ if found - assign user group ID<br>■ if not found - R-F22<br>■ if not unique - R-DUPL<br>■ if null - R-N22 |
| FSK_USERGROUP_ 2_EXT | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ IMP_END_USER_GROUP on USER_STR03. Note: It is recommended that the usage of this column be very limited as it will not be supported in the near future |
| Assigned end user Id and user group ID | Y | | | Additional columns required in source table. Queried against CZ__END_USER_GROUPS.CUSTOMER_ID and CZ_CUSTOMER_END_USER_GROUPS.END_USER_ID.<br>Disposition:<br>■ if found - M<br>■ if not found - I |

*Table A–12 Description of Fields in CZ_IMP_INTL_TEXT Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| INTL_TEXT_ID | N | Y | NUMBER | Designates an OC identifier for international text |
| TEXT_STR | Y | Y | VARCHAR2 (255) | String describes the international text. Contains a foreign surrogate-key value matching CZ_INTL_TEXTS.TEXT_STR.<br>Disposition:<br>■ if found - M<br>■ if not found - I<br>■ if not unique - R-DUPL |
| CREATION_DATE | N | Y | DATE | Indicates the date on which this record was created |
| LAST_UPDATE_DATE | N | Y | DATE | Contains the date on which the record was last modified |
| DELETED_FLAG | N | Y | CHAR (1) | Indicates ('1'/'0') that this record has been deleted |
| EFF_FROM | N | Y | DATE | Indicates the beginning date for which this record is effective |
| EFF_TO | N | Y | DATE | Indicates the ending date through which this record is effective |
| CREATED_BY | N | Y | NUMBER | Identifies the user that created this record |
| LAST_UPDATED_BY | N | Y | NUMBER | Records the login name for the user that last modified this record |
| SECURITY_MASK | N | Y | VARCHAR2 (40) | Reserved for future use in record-level access control |
| EFF_MASK | N | Y | VARCHAR2 (40) | Reserved for future use |
| CHECKOUT_USER | N | Y | VARCHAR2 (40) | Reserved for future use in checkout user |

*Table A–12 Description of Fields in CZ_IMP_INTL_TEXT Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| RUN_ID | N | Y | NUMBER | Contains a number indicating the Import run in which this record was loaded / processed |
| REC_STATUS | N | Y | VARCHAR2 (4) | Records a coded status describing the import results for this record; if null it indicates the record has not yet been completely processed |
| DISPOSITION | N | Y | CHAR (1) | Indicates whether the record was inserted, updated, unchanged, or rejected |

*Table A–13 Description of Fields in CZ_IMP_ITEM_MASTER Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| ITEM_ID | N | Y | NUMBER | Designates an OC identifier for this record |
| ITEM_TYPE_ID | N | Y | NUMBER | Contains the OC Item-type ID for this Item |
| ORIG_SYS_REF | Y | N | VARCHAR2 (255) | Unique identification of a record in this table. Contains a foreign surrogate-key value matching CZ_ITEM_ MASTERS.ORIG_SYS_REF.<br>Disposition:<br>■ if found - M<br>■ if not found - I<br>■ if not unique - R-DUPL<br>■ if null - R-N9 |
| DESC_TEXT | N | Y | VARCHAR2 (255) | Describes this item-master part entry |
| REF_PART_NBR | Y | N | VARCHAR2 (255) | Contains a part number for the item described in this record. Disposition if null - ERR |
| QUOTEABLE_ FLAG | N | Y | CHAR (1) | Indicates that this Item can be separately quoted |

*Table A–13 Description of Fields in CZ_IMP_ITEM_MASTER Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| PRIMARY_UOM_ CODE | N | Y | VARCHAR2 (3) | Contains code for primary unit of measure |
| LEAD_TIME | N | Y | NUMBER | Indicates the ordering / manufacturing lead time |
| ITEM_STATUS | N | Y | NUMBER | Holds a status code number for the item |
| REC_NBR | N | Y | NUMBER | Provides a one-up sequence number identifying the record in this run |
| RUN_ID | N | Y | NUMBER | Contains a number indicating the Import run in which this record was loaded / processed |
| REC_STATUS | N | Y | VARCHAR2 (4) | Records a coded status describing the import results for this record; if null it indicates the record has not yet been completely processed |
| DISPOSITION | N | Y | CHAR (1) | Indicates whether the record was inserted, updated, unchanged, or rejected |
| DELETED_FLAG | N | Y | CHAR (1) | Indicates ('1' / '0') that this record has been deleted |
| EFF_FROM | N | Y | DATE | Indicates the beginning date for which this record is effective |
| EFF_TO | N | Y | DATE | Indicates the ending date through which this record is effective |
| EFF_MASK | N | Y | VARCHAR2 (40) | Reserved for future use |
| USER_STR01 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_STR02 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_STR03 | N | Y | VARCHAR2 (255) | Textual user expansion field; may be used as an alternate 'surrogate key' for the record |
| USER_STR04 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_NUM01 | N | Y | NUMBER (16,9) | Numeric user expansion field |

*Table A–13    Description of Fields in CZ_IMP_ITEM_MASTER Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| USER_NUM02 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM03 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM04 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| CHECKOUT_USER | N | Y | VARCHAR2 (40) | Reserved for future use in checkout user |
| CREATION_DATE | N | Y | DATE | Indicates the date on which this record was created |
| LAST_UPDATE_ DATE | N | Y | DATE | Contains the date on which the record was last modified |
| CREATED_BY | N | Y | NUMBER | Identifies the user that created this record |
| LAST_UPDATED_ BY | N | Y | NUMBER | Records the login name for the user that last modified this record |
| SECURITY_MASK | N | Y | VARCHAR2 (40) | Reserved for future use in record-level access control |
| FSK_ITEMTYPE_1_ 1 | Y | N | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ IITEM_TYPES.ORIG_SYS_REF.<br>Disposition:<br>■ if found - assign item type ID<br>■ if not found - assign default item type ID, if defined, R-F27<br>■ if null - assign default item type ID, if defined, R-F27 |
| FSK_ITEMTYPE_1_ EXT | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ IMP_ITEM_TYPE on USER_STR03. Note: It is recommended that the usage of this column be very limited as it will not be supported in the near future |

*Table A–14 Description of Fields in CZ_IMP_ITEM_PARENT Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| PARENT_ITEM_ID | N | Y | NUMBER | Designates an OC identifier for the parent item |
| ITEM_ID | N | Y | NUMBER | Contains the OC Item ID for the parent item |
| REL_TYPE_ID | N | Y | NUMBER | |
| SEQ_NBR | N | Y | NUMBER | Contains sequence for the item number |
| REC_NBR | N | Y | NUMBER | Provides a one-up sequence number identifying the record in this run |
| RUN_ID | N | Y | NUMBER | Contains a number indicating the Import run in which this record was loaded / processed |
| REC_STATUS | N | Y | VARCHAR2 (4) | Records a coded status describing the import results for this record; if null it indicates the record has not yet been completely processed |
| DISPOSITION | N | Y | CHAR (1) | Indicates whether the record was inserted, updated, unchanged, or rejected |
| DELETED_FLAG | N | Y | CHAR (1) | Indicates ('1' / '0') that this record has been deleted |
| EFF_FROM | N | Y | DATE | Indicates the beginning date for which this record is effective |
| EFF_TO | N | Y | DATE | Indicates the ending date through which this record is effective |
| EFF_MASK | N | Y | VARCHAR2 (40) | Reserved for future use. |
| USER_STR01 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_STR04 | N | Y | VARCHAR2 (255) | Textual user expansion field |

*Table A–14 Description of Fields in CZ_IMP_ITEM_PARENT Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| USER_STR03 | N | Y | VARCHAR2 (255) | Textual user expansion field; may be used as an alternate 'surrogate key' for the record |
| USER_STR04 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_NUM01 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM02 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM03 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM04 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| CHECKOUT_ USER | N | Y | VARCHAR2 (40) | Reserved for future use in checkout user |
| CREATION_DATE | N | Y | DATE | Indicates the date on which this record was created |
| LAST_UPDATE_ DATE | N | Y | DATE | Contains the date on which the record was last modified |
| CREATED_BY | N | Y | NUMBER | Identifies the user that created this record |
| LAST_UPDATED_ BY | N | Y | NUMBER | Records the login name for the user that last modified this record |
| SECURITY_MASK | N | Y | VARCHAR2 (40) | Reserved for future use in record-level access control. |
| FSK_ ITEMMASTER_1_1 | N | N | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ IMP_ITEM_MASTER on REF_PART_NBR |

*Table A–14 Description of Fields in CZ_IMP_ITEM_PARENT Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| FSK_ ITEMMASTER_1_ EXT | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ IMP_ITEM_MASTER on USER_STR03 |
| FSK_ ITEMMASTER_2_1 | N | N | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ IMP_ITEM_MASTER on REF_PART_NBR |
| FSK_ ITEMMASTER_2_ EXT | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ IMP_ITEM_MASTER on USER_STR03 |

*Table A–15 Description of Fields in CZ_IMP_ITEM_PROPERTY_VALUE Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| PROPERTY_ID | N | Y | NUMBER | Contains the OC ID for the referenced property |
| ITEM_ID | N | Y | NUMBER | Contains the OC Item-Master ID to which this property value applies |
| PROPERTY_ VALUE | N | Y | VARCHAR2 (255) | Contains the value that is assigned to the property for this item |
| REC_NBR | N | Y | NUMBER | Provides a one-up sequence number identifying the record in this run |
| RUN_ID | N | Y | NUMBER | Contains a number indicating the Import run in which this record was loaded / processed |
| REC_STATUS | N | Y | VARCHAR2 (4) | Records a coded status describing the import results for this record; if null it indicates the record has not yet been completely processed |
| DISPOSITION | N | Y | CHAR | Indicates whether the record was inserted, updated, unchanged, or rejected |
| DELETED_FLAG | N | Y | CHAR | Indicates ('1' / '0') that this record has been deleted |

*Table A–15 Description of Fields in CZ_IMP_ITEM_PROPERTY_VALUE Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| EFF_FROM | N | Y | DATE | Indicates the beginning date for which this record is effective |
| EFF_TO | N | Y | DATE | Indicates the ending date through which this record is effective |
| EFF_MASK | N | Y | VARCHAR2 (40) | Reserved for future use |
| USER_STR01 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_STR02 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_STR03 | N | Y | VARCHAR2 (255) | Textual user expansion field; may be used as an alternate ’surrogate key’ for the record |
| USER_STR04 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_NUM01 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM02 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM03 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM04 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| CHECKOUT_USER | N | Y | VARCHAR2 (40) | Reserved for future use in checkout user |
| CREATION_DATE | N | Y | DATE | Indicates the date on which this record was created |
| LAST_UPDATE_ DATE | N | Y | DATE | Contains the date on which the record was last modified |
| CREATED_BY | N | Y | NUMBER | Identifies the user that created this record |
| LAST_UPDATED_ BY | N | Y | NUMBER | Records the login name for the user that last modified this record |

*Table A–15 Description of Fields in CZ_IMP_ITEM_PROPERTY_VALUE Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| SECURITY_MASK | N | Y | VARCHAR2 (40) | Reserved for future use in record-level access control |
| FSK_PROPERTY_1_1 | Y | N | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_PROPERTIES.ORIG_SYS_REF.<br>Disposition:<br>■ if found - assign property ID<br>■ if not found - R-F23<br>■ if not unique - R-DUPL<br>■ if null - R-N23 |
| FSK_PROPERTY_1_EXT | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_IMP_PROPERTY on USER_STR03. Note: It is recommended that the usage of this column be very limited as it will not be supported in the near future |
| FSK_ITEMMASTER_2_1 | Y | N | VARCHAR2 (255) | Contains foreign surrogate-key values matching CZ_ITEM_MASTERS.ORIG_SYS_REF.<br>Disposition:<br>■ if found - assign item ID<br>■ if not found - R-F25<br>■ if not unique - R-DUPL<br>■ if null - R-N25 |
| FSK_ITEMMASTER_2_EXT | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_IMP_ITEM_MASTER on USER_STR03. Note: It is recommended that the usage of this column be very limited as it will not be supported in the near future |
| Assigned property ID and item ID | Y | | | Additional columns required in source table. Queried against CZ_ITEM_PROPERTY_VALUES.PROPERTY_ID and CZ_ITEM_PROPERTY_VALUES.ITEM_ID.<br>Disposition:<br>■ if found - M<br>■ if not found - I |

*Table A–16 Description of Fields in CZ_IMP_ITEM_TYPE Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| ITEM_TYPE_ID | N | Y | NUMBER | Designates an OC Identifier for the item type |
| DESC_TEXT | N | Y | VARCHAR2 (255) | Describes this item type |
| NAME | N | Y | VARCHAR2 (255) | Contains the name of the Item Type |
| REC_NBR | N | Y | NUMBER | Provides a one-up sequence number identifying the record in this run |
| RUN_ID | N | Y | NUMBER | Contains a number indicating the Import run in which this record was loaded / processed |
| REC_STATUS | N | Y | VARCHAR2 (4) | Records a coded status describing the import results for this record; if null it indicates the record has not yet been completely processed |
| DISPOSITION | N | Y | CHAR (1) | Indicates whether the record was inserted, updated, unchanged, or rejected |
| DELETED_FLAG | N | Y | CHAR (1) | Indicates ('1' / '0') that this record has been deleted |
| ORIG_SYS_REF | Y | N | VARCHAR2 (255) | Unique identification of a record in this table. Contains a foreign surrogate-key value matchingCZ_ITEM_ TYPES.ORIG_SYS_REF.<br>Disposition:<br>■ if found - M<br>■ if not found - I<br>■ if not unique - R-DUPL<br>■ if null - R-N11 |
| EFF_FROM | N | Y | DATE | Indicates the beginning date for which this record is effective |
| EFF_TO | N | Y | DATE | Indicates the ending date through which this record is effective |
| EFF_MASK | N | Y | VARCHAR2 (40) | Reserved for future use |
| USER_STR01 | N | Y | VARCHAR2 (255) | Textual user expansion field |

*Table A–16 Description of Fields in CZ_IMP_ITEM_TYPE Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| USER_STR02 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_STR03 | N | Y | VARCHAR2 (255) | Textual user expansion field; may be used as an alternate ’surrogate key’ for the record |
| USER_STR04 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_NUM01 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM02 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM03 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM04 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| CHECKOUT_USER | N | Y | VARCHAR2 (40) | Reserved for future use in checkout user |
| CREATION_DATE | N | Y | DATE | Indicates the date on which this record was created |
| LAST_UPDATE_ DATE | N | Y | DATE | Contains the date on which the record was last modified |
| CREATED_BY | N | Y | NUMBER | Identifies the user that created this record |
| LAST_UPDATED_ BY | N | Y | NUMBER | Records the login name for the user that last modified this record |
| SECURITY_MASK | N | Y | VARCHAR2 (40) | Reserved for future use in record-level access control |

*Table A–17 Description of Fields in CZ_IMP_ITEM_TYPE_PROPERTY Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| ITEM_TYPE_ID | N | Y | NUMBER | Designates an OC identifier for an item type |
| PROPERTY_ID | N | Y | NUMBER | Contains the OC property ID for an item type |
| REC_NBR | N | Y | NUMBER | Provides a one-up sequence number identifying the record in this run |
| RUN_ID | N | Y | NUMBER | Contains a number indicating the Import run in which this record was loaded / processed |
| REC_STATUS | N | Y | VARCHAR2 (4) | Records a coded status describing the import results for this record; if null it indicates the record has not yet been completely processed |
| DISPOSITION | N | Y | CHAR (1) | Indicates whether the record was inserted, updated, unchanged, or rejected |
| DELETED_FLAG | N | Y | CHAR (1) | Indicates ('1' / '0') that this record has been deleted |
| EFF_FROM | N | Y | DATE | Indicates the beginning date for which this record is effective |
| EFF_TO | N | Y | DATE | Indicates the ending date through which this record is effective |
| EFF_MASK | N | Y | VARCHAR2 (40) | Reserved for future use. |
| USER_STR01 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_STR02 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_STR03 | N | Y | VARCHAR2 (255) | Textual user expansion field; may be used as an alternate 'surrogate key' for the record |
| USER_STR04 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_NUM01 | N | Y | NUMBER (16,9) | Numeric user expansion field |

*Table A–17 Description of Fields in CZ_IMP_ITEM_TYPE_PROPERTY Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| USER_NUM02 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM03 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM04 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| CHECKOUT_USER | N | Y | VARCHAR2 (40) | Reserved for future use in checkout user |
| CREATION_DATE | N | Y | DATE | Indicates the date on which this record was created |
| LAST_UPDATE_DATE | N | Y | DATE | Contains the date on which the record was last modified |
| CREATED_BY | N | Y | NUMBER | Identifies the user that created this record |
| LAST_UPDATED_BY | N | Y | NUMBER | Records the login name for the user that last modified this record |
| SECURITY_MASK | N | Y | VARCHAR2 (40) | Reserved for future use in record-level access control |
| FSK_ITEMTYPE_1_1 | Y | N | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ITEM_TYPES.ORIG_SYS_REF.<br>Disposition:<br>■ if found - assign item type ID<br>■ if not found - R-F22<br>■ if not unique - R-DUPL<br>■ if null - R-N22 |
| FSK_ITEMTYPE_1_EXT | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_IMP_ITEM_TYPE on USER_STR03 |

*Table A–17 Description of Fields in CZ_IMP_ITEM_TYPE_PROPERTY Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| FSK_PROPERTY_2_1 | Y | N | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_PROPERTIES.ORIG_SYS_REF.<br>Disposition:<br>■ if found - assign property ID<br>■ if not found - R-F24<br>■ if not unique - R-DUPL<br>■ if null - R-N24 |
| FSK_PROPERTY_2_EXT | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_IMP_PROPERTY on USER_STR03. Note: It is recommended that the usage of this column be very limited as it will not be supported in the near future |
| Assigned property ID and item type ID | Y | | | Additional columns required in source table. Queried against CZ_ITEM_TYPE_PROPERTIES.PROPERTY_ID and CZ_ITEM_TYPE_PROPERTIES.ITEM_TYPE_ID.<br>Disposition:<br>■ if found - M<br>■ if not found - I |

*Table A–18 Description of Fields in CZ_IMP_PRICE Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| ITEM_ID | N | Y | NUMBER | Contains the OC Item-Master ID for this record |
| PRICE_GROUP_ID | N | Y | NUMBER | Contains the OC ID for the price group of which this price is a part |
| ITEM_PRICE | N | Y | NUMBER (16,9) | Contains the price |
| EFF_FROM_DATE | N | Y | DATE | Indicates the beginning date for which this record is effective |

*Table A–18 Description of Fields in CZ_IMP_PRICE Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| EFF_TO_DATE | N | Y | DATE | Indicates the ending date through which this record is effective |
| REC_NBR | N | Y | NUMBER | Provides a one-up sequence number identifying the record in this run |
| RUN_ID | N | Y | NUMBER | Contains a number indicating the Import run in which this record was loaded / processed |
| REC_STATUS | N | Y | VARCHAR2 (4) | Records a coded status describing the import results for this record; if null it indicates the record has not yet been completely processed |
| DISPOSITION | N | Y | CHAR (1) | Indicates whether the record was inserted, updated, unchanged, or rejected |
| DELETED_FLAG | N | Y | CHAR (1) | Indicates ('1' / '0') that this record has been deleted |
| EFF_FROM | N | Y | DATE | Indicates the beginning date for which this record is effective |
| EFF_TO | N | Y | DATE | Indicates the ending date through which this record is effective |
| EFF_MASK | N | Y | VARCHAR2 (40) | Reserved for future use |
| CHECKOUT_USER | N | Y | VARCHAR2 (40) | Reserved for future use in checkout user |
| CREATION_DATE | N | Y | DATE | Indicates the date on which this record was created |
| LAST_UPDATE_ DATE | N | Y | DATE | Contains the date on which the record was last modified |
| CREATED_BY | N | Y | NUMBER | Identifies the user that created this record |
| LAST_UPDATED_ BY | N | Y | NUMBER | Records the login name for the user that last modified this record |
| SECURITY_MASK | N | Y | VARCHAR2 (40) | Reserved for future use in record-level access control. |

*Table A–18 Description of Fields in CZ_IMP_PRICE Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| FSK_ ITEMMASTER_1_1 | Y | N | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ ITEM_MASTERS.ORIG_SYS_REF.<br>Disposition:<br>■ if found - assign item ID<br>■ if not found - R-F28<br>■ if not unique - R-DUPL<br>■ if null - R-N28 |
| FSK_ ITEMMASTER_1_ EXT | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ IMP_ITEM_MASTER on USER_STR03 |
| FSK_ PRICEGROUP_2_1 | Y | N | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ PRICE_GROUPS.ORIG_SYS_REF.<br>Disposition:<br>■ if found - assign price group ID<br>■ if not found - R-F30<br>■ if not unique - R-DUPL<br>■ if null - R-N30 |
| FSK_ PRICEGROUP_2_ EXT | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ IMP_PRICE_GROUP on USER_STR03. Note: It is recommended that the usage of this column be very limited as it will not be supported in the near future |
| Assigned item ID and price group ID | Y | | | Additional columns required in source table. Queried against CZ_PRICES.ITEM_ID and CZ_PRICES.PRICE_ GROUP_ID.<br>Disposition:<br>■ if found - M<br>■ if not found - I |

*Table A–19 Description of Fields in CZ_IMP_PRICE_GROUP Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| PRICE_GROUP_ID | N | Y | NUMBER | Designates an OC identifier for this price group |
| DESC_TEXT | N | Y | VARCHAR2 (255) | Describes this price group |
| NAME | N | Y | VARCHAR2 (255) | Contains the name of the Price Group |
| ORIG_SYS_REF | Y | N | VARCHAR2 (255) | Unique identification of a record in this table. Contains a foreign surrogate-key value matching CZ_PRICE_GROUPSS.ORIG_SYS_REF.<br>Disposition:<br>■ if found -M<br>■ if not found - I<br>■ if not unique - R-DUPL<br>■ if null - R-N5 |
| REC_NBR | N | Y | NUMBER | Provides a one-up sequence number identifying the record in this run |
| RUN_ID | N | Y | NUMBER | Contains a number indicating the Import run in which this record was loaded / processed |
| REC_STATUS | N | Y | VARCHAR2 (4) | Records a coded status describing the import results for this record; if null it indicates the record has not yet been completely processed |
| DISPOSITION | N | Y | CHAR (1) | Indicates whether the record was inserted, updated, unchanged, or rejected |
| DELETED_FLAG | N | Y | CHAR (1) | Indicates ('1' / '0') that this record has been deleted |
| CURRENCY | N | Y | VARCHAR2 (20) | Contains the type of currency used |
| EFF_FROM | N | Y | DATE | Indicates the beginning date for which this record is effective |
| EFF_TO | N | Y | DATE | Indicates the ending date through which this record is effective |
| EFF_MASK | N | Y | VARCHAR2 (40) | Reserved for future use |

*Table A–19 Description of Fields in CZ_IMP_PRICE_GROUP Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| USER_STR01 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_STR02 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_STR03 | N | Y | VARCHAR2 (255) | Textual user expansion field; may be used as an alternate ’surrogate key’ for the record |
| USER_STR04 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_NUM01 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM02 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM03 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM04 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| CHECKOUT_USER | N | Y | VARCHAR2 (40) | Reserved for future use in checkout user |
| CREATION_DATE | N | Y | DATE | Indicates the date on which this record was created |
| LAST_UPDATE_ DATE | N | Y | DATE | Contains the date on which the record was last modified |
| CREATED_BY | N | Y | NUMBER | Identifies the user that created this record |
| LAST_UPDATED_ BY | N | Y | NUMBER | Records the login name for the user that last modified this record |
| SECURITY_MASK | N | Y | VARCHAR2 (40) | Reserved for future use in record-level access control |

*Table A–20 Description of Fields in CZ_IMP_PROPERTY Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| PROPERTY_ID | N | Y | NUMBER | Designates an OC Identifier for the property |
| PROPERTY_UNIT | N | Y | VARCHAR2 (8) | Indicates the units in which this property is measured or allocated |
| DESC_TEXT | N | Y | VARCHAR2 (255) | Describes this property |
| NAME | Y | N | VARCHAR2 (255) | Contains the name of the Property. Disposition if null - ERR |
| DATA_TYPE | Y | N | NUMBER | Indicates the data type that this property bears. Disposition if null - ERR |
| DEF_VALUE | N | Y | VARCHAR2 (255) | Records a default value for the property |
| REC_NBR | N | Y | NUMBER | Provides a one-up sequence number identifying the record in this run |
| RUN_ID | N | Y | NUMBER | Contains a number indicating the Import run in which this record was loaded / processed |
| REC_STATUS | N | Y | VARCHAR2 (4) | Records a coded status describing the import results for this record; if null it indicates the record has not yet been completely processed |
| DISPOSITION | N | Y | CHAR (1) | Indicates whether the record was inserted, updated, unchanged, or rejected |
| DELETED_FLAG | N | Y | CHAR (1) | Indicates ('1' / '0') that this record has been deleted |
| ORIG_SYS_REF | Y | N | VARCHAR2 (255) | Unique identification of a record in this table. Contains a foreign surrogate-key value matchingCZ_ PROPERTIES.ORIG_SYS_REF.<br>Disposition:<br>■ if found - M<br>■ if not found - I<br>■ if not unique - R-DUPL<br>■ if null - R-N17 |
| EFF_FROM | N | Y | DATE | Indicates the beginning date for which this record is effective |

*Table A–20 Description of Fields in CZ_IMP_PROPERTY Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| EFF_TO | N | Y | DATE | Indicates the ending date through which this record is effective |
| EFF_MASK | N | Y | VARCHAR2 (40) | Reserved for future use |
| USER_STR01 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_STR02 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_STR03 | N | Y | VARCHAR2 (255) | Textual user expansion field; may be used as an alternate ’surrogate key’ for the record |
| USER_STR04 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_NUM01 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM02 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM03 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM04 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| CHECKOUT_USER | N | Y | VARCHAR2 (40) | Reserved for future use in checkout user |
| CREATION_DATE | N | Y | DATE | Indicates the date on which this record was created |
| LAST_UPDATE_DATE | N | Y | DATE | Contains the date on which the record was last modified |
| CREATED_BY | N | Y | NUMBER | Identifies the user that created this record |
| LAST_UPDATED_BY | N | Y | NUMBER | Records the login name for the user that last modified this record |
| SECURITY_MASK | N | Y | VARCHAR2 (40) | Reserved for future use in record-level access control |

*Table A–21 Description of Fields in CZ_IMP_PS_NODE Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| PS_NODE_ID | N | Y | NUMBER | Designates an OC Identifier for the model node |
| DEVL_PROJECT_ ID | N | Y | NUMBER | Contains the OC development project ID for this record |
| FROM_ POPULATOR_ID | N | Y | NUMBER | Set when the node is created by a populator; designates the OC identifier for the populator that created this record |
| PROPERTY_ BACKPTR | N | Y | NUMBER | Set when the node is created by a populator; identifies the property from which this record was created |
| ITEM_TYPE_ BACKPTR | N | Y | NUMBER | Set when the node is created by a populator; identifies the item type of the populator that created this record |
| INTL_TEXT_ID | N | Y | NUMBER | Contains the OC international text ID for this record |
| SUB_CONS_ID | N | Y | NUMBER | Currently not used |
| ITEM_ID | N | Y | NUMBER | Contains the OC item ID for this record |
| NAME | N | Y | VARCHAR2 (255) | Contains the OC name for the model node (i.e. component, feature, etc.) |
| RESOURCE_FLAG | N | Y | CHAR (1) | Indicates that this node is a Total or Resource |
| INITIAL_VALUE | N | Y | VARCHAR2 (255) | Records the initial value for this node when instantiated in a configuration |
| PARENT_ID | N | Y | NUMBER | Contains the OC identifier for the parent node |
| MINIMUM | N | Y | NUMBER (16,9) | Contains the minimum selection requirement |
| MAXIMUM | N | Y | NUMBER (16,9) | Contains the maximum selection requirement |
| PS_NODE_TYPE | Y | N | NUMBER | Contains a numeric identification of the node’s type such as component, feature, etc. Disposition if null - ERR |
| FEATURE_TYPE | N | Y | NUMBER | Contains the data type of the feature node |
| PRODUCT_FLAG | N | Y | CHAR (1) | Contains a flag indicating that the node is a parent node |

*Table A–21 Description of Fields in CZ_IMP_PS_NODE Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| REFERENCE_ID | N | Y | NUMBER | Contains a reference to a project structure subtree for which this PS node is a surrogate. Note: Referencing is not currently supported |
| MULTI_CONFIG_ FLAG | N | Y | CHAR (1) | Indicates node children can be configured separately |
| ORDER_SEQ_ FLAG | N | Y | CHAR (1) | Indicates the component is a ordered sequence |
| SYSTEM_NODE_ FLAG | N | Y | CHAR (1) | Indicates this record is a system node, i.e. either root node or template for roots |
| TREE_SEQ | Y | N | NUMBER | Contains the order of this child node within the parent. Disposition if null - ERR |
| COUNTED_ OPTIONS_FLAG | N | Y | CHAR (1) | Indicates this feature has counted options |
| UI_OMIT | N | Y | CHAR (1) | Indicates whether or not the node is visible in the UI |
| UI_SECTION | N | Y | NUMBER | Indicates in which section of the UI the node is visible |
| BOM_ TREATMENT | N | Y | NUMBER | Indicates how BOM should be enumerated during configuration order generation |
| ORIG_SYS_REF | Y | N | VARCHAR2 (1500) | Unique identification of a record in this table. Contains a foreign surrogate-key value matching CZ_PS_ NODES.ORIG_SYS_REF.<br>Disposition:<br>■ if found - M<br>■ if not found - I<br>■ if not unique - R-DUPL<br>■ if null - R-N9 |
| RUN_ID | N | Y | NUMBER | Contains a number indicating the Import run in which this record was loaded / processed |
| REC_STATUS | N | Y | VARCHAR2 (4) | Records a coded status describing the import results for this record; if null it indicates the record has not yet been completely processed |
| DISPOSITION | N | Y | CHAR (1) | Indicates whether the record was inserted, updated, unchanged, or rejected |

*Table A–21 Description of Fields in CZ_IMP_PS_NODE Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| DELETED_FLAG | N | Y | CHAR (1) | Indicates ('1'/'0') that this record has been deleted |
| EFF_FROM | N | Y | DATE | Indicates the beginning date for which this record is effective |
| EFF_TO | N | Y | DATE | Indicates the ending date through which this record is effective |
| EFF_MASK | N | Y | VARCHAR2 (40) | Reserved for future use |
| USER_STR01 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_STR02 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_STR03 | N | Y | VARCHAR2 (255) | Textual user expansion field; may be used as an alternate 'surrogate key' for the record |
| USER_STR04 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| USER_NUM01 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM02 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM03 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| USER_NUM04 | N | Y | NUMBER (16,9) | Numeric user expansion field |
| CHECKOUT_USER | N | Y | VARCHAR2 (40) | Reserved for future use in checkout user |
| CREATION_DATE | N | Y | DATE | Indicates the date on which this record was created |
| LAST_UPDATE_ DATE | N | Y | DATE | Contains the date on which the record was last modified |
| CREATED_BY | N | Y | NUMBER | Identifies the user that created this record |

*Table A–21 Description of Fields in CZ_IMP_PS_NODE Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| LAST_UPDATED_BY | N | Y | NUMBER | Records the login name for the user that last modified this record |
| SECURITY_MASK | N | Y | VARCHAR2 (40) | Reserved for future use in record-level access control |
| FSK_INTLTEXT_1_1 | Y | N (see description) | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_INTL_TEXTS.TEXT_STR.<br>Disposition:<br>■ if found - assign international text ID<br>■ if not found - R-F44 (except model and project structure nodes)<br>■ if null - R-N44 (Only Model and Project Structure nodes are nullable.) |
| FSK_INTLTEXT_1_EXT | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_IMP_INTL_TEXT on USER_STR03. Note: It is recommended that the usage of this column be very limited as it will not be supported in the near future |
| FSK_ITEMMASTER_2_1 | Y | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_ITEM_MASTERS.ORIG_SYS_REF.<br>Disposition:<br>■ if found - assign item ID<br>■ if not found - R-F46 (except model and project structure nodes) |
| FSK_ITEMMASTER_2_EXT | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_IMP_ITEM_MASTER on ORIG_SYS_REF. Note: It is recommended that the usage of this column be very limited as it will not be supported in the near future |
| FSK_PSNODE_3_1 | Y | N (see description) | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_PS_NODES.ORIG_SYS_REF.<br>Disposition:<br>■ if found - assign parent Project Structure node ID<br>■ if not found - R-F48 (except model nodes)<br>■ if null - R-N48.(Only Model nodes are nullable.) |
| FSK_PSNODE_3_EXT | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_IMP_PS_NODE on USER_STR03. Note: It is recommended that the usage of this column be very limited as it will not be supported in the near future |

*Table A–21 Description of Fields in CZ_IMP_PS_NODE Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| FSK_PSNODE_4_1 | N | Y | VARCHAR2 (255) | Currently not used |
| FSK_PSNODE_4_EXT | N | Y | VARCHAR2 (255) | Currently not used |
| FSK_DEVLPROJECT_5_1 | Y | N | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_DEVL_PROJECTS.ORIG_SYS_REF.<br>Disposition:<br>■ if found - assign development project ID<br>■ if not found - R-F50<br>■ if null - R-N50 |
| FSK_DEVLPROJECT_5_EXT | N | Y | VARCHAR2 (255) | Contains a foreign surrogate-key value matching CZ_IMP_DEVL_PROJECT on USER_STR03. Note: It is recommended that the usage of this column be very limited as it will not be supported in the near future |
| COMPONENT_SEQUENCE_ID | N | Y | NUMBER | Component sequence identifier from BOM explosions |
| COMPONENT_CODE | N | Y | VARCHAR2 (1000) | Contains the path from the root BOM node |
| PLAN_LEVEL | Y | N | NUMBER | Indicates the depth of this node in the BOM structure. Disposition if null - ERR |
| BOM_ITEM_TYPE | N | Y | NUMBER | Indicates whether this node is a Model, Standard, or OptionClass BOM node |
| SO_ITEM_TYPE_CODE | N | Y | VARCHAR2 | Describes the application’s item type for ordering: model, class, kit, standard |
| MINIMUM_SELECTED | N | Y | NUMBER | For OptionClass nodes, indicates the minimum quantity selection for its children |
| MAXIMUM_SELECTED | N | Y | NUMBER | For OptionClass nodes, indicates the maximum quantity selection for its children |
| BOM_REQUIRED | N | Y | CHAR (1) | Contains a flag indicating that this node is required for BOM |

*Table A–22 Description of Fields in CZ_IMP_USER_GROUP Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| USER_GROUP_ID | N | Y | NUMBER | Designates an OC Identifier for this record |
| DESC_TEXT | N | Y | VARCHAR2 (255) | Describes this user group |
| GROUP_NAME | Y | N | VARCHAR2 (20) | Contains the name of the User Group. Contains a foreign surrogate-key value matchingCZ_USER_ GROUPS.GROUP_NAME.<br>Disposition:<br>■ if found - M<br>■ if not found - I<br>■ if not unique - R-DUPL<br>■ if null - R-N2 |
| GROUP_DESC | N | Y | VARCHAR2 (20) | Descriptions of the user group |
| READ_AUTH | N | Y | CHAR (1) | Flag for read authorization to the user group |
| CREATE_AUTH | N | Y | CHAR (1) | Flag for create authorization to the user group |
| DELETE_AUTH | N | Y | CHAR (1) | Flag for delete authorization to the user group |
| UPDATE_AUTH | N | Y | CHAR (1) | Flag for update authorization to the user group |
| USER_GROUP_ DISC_LIMIT | N | Y | NUMBER | Indicates the group discount limit |
| ALLOW_CONFIG_ CHANGES | N | Y | VARCHAR2 (20) | Indicates whether or not users of the group are allowed to change configurations |
| CHECKOUT_USER | N | Y | VARCHAR2 (40) | Reserved for future use in checkout user |
| REC_NBR | N | Y | NUMBER | Provides a one-up sequence number identifying the record in this run |
| RUN_ID | N | Y | NUMBER | Contains a number indicating the Import run in which this record was loaded / processed |
| REC_STATUS | N | Y | VARCHAR2 (4) | Records a coded status describing the import results for this record; if null it indicates the record has not yet been completely processed |

*Table A–22 Description of Fields in CZ_IMP_USER_GROUP Import Table*

| COLUMN_NAME | Required | Nullable | DATA_TYPE | Description |
|---|---|---|---|---|
| DISPOSITION | N | Y | CHAR (1) | Indicates whether the record was inserted, updated, unchanged, or rejected |
| DELETED_FLAG | N | Y | CHAR (1) | Indicates ('1' / '0') that this record has been deleted |
| EFF_FROM | N | Y | DATE | Indicates the beginning date for which this record is effective |
| EFF_TO | N | Y | DATE | Indicates the ending date through which this record is effective |
| EFF_MASK | N | Y | VARCHAR2 (40) | Reserved for future use |
| USER_STR03 | N | Y | VARCHAR2 (255) | Textual user expansion field |
| CREATION_DATE | N | Y | DATE | Indicates the date on which this record was created |
| LAST_UPDATE_ DATE | N | Y | DATE | Contains the date on which the record was last modified |
| CREATED_BY | N | Y | NUMBER | Identifies the user that created this record |
| LAST_UPDATED_ BY | N | Y | NUMBER | Records the login name for the user that last modified this record |
| SECURITY_MASK | N | Y | VARCHAR2 (40) | Reserved for future use in record-level access control |

# B
# Export Tables

This appendix presents which data in the Oracle Configurator schema are transferred to which tables and fields in the Oracle Applications database.

## B.1 CZ Schema Source and Oracle Applications Destination Tables

Table B–1 presents the data in the Oracle Configurator schema that are transferred to populate the interface tables and fields in the Oracle Applications database. Oracle Applications concurrent programs extract the data from the interface tables to populate fields in the Oracle Applications tables so that orders and new customer information submitted by the Oracle runtime configurator are complete.

*Table B–1 Data Export to Oracle Applications Source and Interface Tables*

| Oracle Configurator schema Source | -> | Oracle Applications Interface Table(s) |
|---|---|---|
| CZ_QUOTE_HDRS.QUOTE_HDR_ID<br>CZ_QUOTE_HDRS.QUOTE_REV_NBR<br>Integer for standard User ID | --> | SO_HEADERS_INTERFACE_ALL.ORIGINAL_SYSTEM_REFERENCE<br>SO_LINES_INTERFACE_ALL.ORIGINAL_SYSTEM_REFERENCE<br>SO_PRICE_ADJUSTMENT_INTERFACE.ORIGINAL_SYSTEM_REFERENCE |
| CZ_QUOTE_HDRS.ORDER_SUBMITTED_DATE | --> | SO_LINES_INTERFACE_ALL.CREATION_DATE<br>SO_PRICE_ADJUSTMENT_INTERFACE.CREATION_DATE |
| CZ_QUOTE_HDRS.USER_ID_CREATED | --> | SO_HEADERS_INTERFACE_ALL.CREATED_BY<br>SO_LINES_INTERFACE_ALL.CREATED_BY<br>SO_PRICE_ADJUSTMENT_INTERFACE.CREATED_BY |
| CZ_QUOTE_HDRS.ORDER_SUBMITTED_DATE | --> | SO_HEADERS_INTERFACE_ALL.LAST_UPDATE_DATE<br>SO_LINES_INTERFACE_ALL.LAST_UPDATE_DATE<br>SO_PRICE_ADJUSTMENT_INTERFACE.LAST_UPDATE_DATE |

*Table B–1 Data Export to Oracle Applications Source and Interface Tables*

| Oracle Configurator schema Source | -> | Oracle Applications Interface Table(s) |
|---|---|---|
| CZ_QUOTE_HDRS.USER_ID_CREATED | --> | SO_HEADERS_INTERFACE_ALL.LAST_UPDATED_BY<br>SO_LINES_INTERFACE_ALL.LAST_UPDATED_BY<br>SO_PRICE_ADJUSTMENT_INTERFACE.LAST_UPDATED_BY |
| CZ_QUOTE_HDRS.ORDER_SUBMITTED_ DATE | --> | SO_HEADERS_INTERFACE_ALL.DATE_ORDERED |
| CZ_QUOTE_HDRS<br>or imported SO_PRICE_LIST_ID for default price list CZ_DB_SETTINGS.SETTING_ ID='DefaultPriceId' | --> | SO_HEADERS_INTERFACE_ALL.PRICE_LIST_ID |
| CZ_QUOTE_HDRS.DISC_PERCENTAGE | --> | SO_PRICE_ADJUSTMENT_INTERFACE.PERCENT |
| CZ_QUOTE_HDRS<br>Parameter(NO COLUMNS)={'INSERT','DELETE'} | --> | SO_HEADERS_INTERFACE_ALL.OPERATION_CODE |
| CZ_DB_SETTINGS.SETTING_ ID='OrderTypeId' | --> | SO_HEADERS_INTERFACE_ALL.ORDER_TYPE_ID |
| CZ_DB_SETTINGS.SETTING_ ID='OrderSourceId' | --> | SO_HEADERS_INTERFACE_ALL.ORDER_SOURCE_ID<br>SO_LINES_INTERFACE_ALL.ORDER_SOURCE_ID<br>SO_PRICE_ADJUSTMENT_INTERFACE.ORDER_SOURCE_ID |
| value 'R' | --> | SO_HEADERS_INTERFACE_ALL.ORDER_CATEGORY |
| calculated value | --> | SO_LINES_INTERFACE_ALL.ORIGINAL_SYSTEM_LINE_ REFERENCE |
| calculated value | --> | SO_LINES_INTERFACE_ALL.LINE_NUMBER |
| value 'REGULAR' | --> | SO_LINES_INTERFACE_ALL.LINE_TYPE |
| SYSDATE | --> | SO_LINES_INTERFACE_ALL.DATE_REQUESTED_CURRENT |
| calculated value | --> | SO_LINES_INTERFACE_ALL.LINK_TO_LINE_REF |
| calculated value | --> | SO_LINES_INTERFACE_ALL.PARENT_LINE_REF |
| value 'N' or 'Y' | --> | SO_LINES_INTERFACE_ALL.CALCULATE_PRICE |
| value 'N' or 'Y' | --> | SO_LINES_INTERFACE_ALL.OPTION_FLAG |
| CZ_DB_SETTINGS.SETTING_ ID='DiscontId' | --> | SO_PRICE_ADJUSTMENT_INTERFACE.DISCOUNT_ID |
| CZ_ITEM_MASTERS.PRIMARY_UOM_ CODE | --> | SO_LINES_INTERFACE_ALL.UNIT_CODE |

*Table B–1 Data Export to Oracle Applications Source and Interface Tables*

| Oracle Configurator schema Source | -> | Oracle Applications Interface Table(s) |
|---|---|---|
| Function of tCZ_ITEM_MASTERS.ORIG_SYS_REF<br>and CZ_ITEM_MASTERS.REF_PART_NUMBER (inventory_id) | --> | SO_LINES_INTERFACE_ALL.INVENTORY_ITEM_ID |
| Function of tCZ_ITEM_MASTERS.ORIG_SYS_REF and CZ_ITEM_MASTERS.REF_PART_NUMBER (organization_id) | --> | SO_LINES_INTERFACE_ALL.WAREHOUSE_ID |
| Function of CZ_QUOTE_MAIN_ITEMS.ITEM_QTY | --> | SO_LINES_INTERFACE_ALL.ORDERED_QUANTITY |
| Function of CZ_DRILL_DOWN_ITEMS.CONFIG_ITEM_QTY | --> | SO_LINES_INTERFACE_ALL.ORDERED_QUANTITY |
| QUOTED_LIST_PRICE *((100_VAR1.DISC_PERCENTAGE) / 100) | --> | SO_LINES_INTERFACE_ALL.LIST_PRICE |
| QUOTED_LIST_PRICE *((100_VAR1.DISC_PERCENTAGE) / 100) | --> | SO_LINES_INTERFACE_ALL.SELLING_PRICE |
| Function of CZ_END_USERS.ORIG_SYS_REF and CZ_END_USERS.LOGIN_NAME | --> | SO_HEADERS_INTERFACE_ALL.SALES_REP_ID |
| Function of CZ_END_USERS.ORIG_SYS_REF and CZ_END_USERS.END_USER_ORG_ID | --> | SO_HEADERS_INTERFACE_ALL.ORG_ID<br>SO_LINES_INTERFACE_ALL.ORG_ID |
| CZ_CUSTOMERS.CUSTOMER_ID | --> | RA_CUSTOMER_PROFILES_INT_ALL.ORIG_SYSTEM_CUSTOMER_REF<br>RA_CUSTOMERS_INTERFACE_ALL.ORIG_SYSTEM_CUSTOMER_REF |
| CZ_CUSTOMERS.CUSTOMER_NAME | --> | RA_CUSTOMERS_INTERFACE_ALL.CUSTOMER_NAME |
| Function of CZ_CUSTOMERS.ORIG_SYS_REF derived from QUOTE_HDR.OPPORTUNITY_HDR_ID --> OPPORTUNITY_HDR.CUSTOMER_ID | --> | SO_HEADERS_INTERFACE_ALL.CUSTOMER_ID |
| Function of CZ_CUSTOMERS.ORIG_SYS_REF derived from QUOTE_HDR.BILL_CUSTOMER_ID | --> | SO_HEADERS_INTERFACE_ALL.INVOICE_CUSTOMER_ID |
| Function of CZ_CUSTOMERS.ORIG_SYS_REF derived from QUOTE_HDR.SHIPTO_CUSTOMER_ID | --> | SO_HEADERS_INTERFACE_ALL.SHIP_TO_CUSTOMER_ID |

*Table B–1 Data Export to Oracle Applications Source and Interface Tables*

| Oracle Configurator schema Source | -> | Oracle Applications Interface Table(s) |
|---|---|---|
| NULL value | --> | RA_CUSTOMERS_INTERFACE_ALL.CUSTOMER_NUMBER |
| value of "A" | --> | RA_CUSTOMERS_INTERFACE_ALL.CUSTOMER_STATUS |
| value of "Y" | --> | RA_CUSTOMERS_INTERFACE_ALL.PRIMARY_SITE_USE_FLAG |
| value of "I" | --> | RA_CUSTOMERS_INTERFACE_ALL.INSERT_UPDATE_FLAG<br>RA_CUSTOMER_PROFILES_INT_ALL.INSERT_UPDATE_FLAG |
| value of "CUSTOMER" | --> | RA_CUSTOMERS_INTERFACE_ALL.CUSTOMER_PROSPECT_CODE |
| Integer for standard User ID | --> | RA_CUSTOMERS_INTERFACE_ALL.LAST_UPDATED_BY |
| SYSDATE | --> | RA_CUSTOMERS_INTERFACE_ALL.LAST_UPDATE_DATE<br>RA_CUSTOMER_PROFILES_INT_ALL.LAST_UPDATE_DATE |
| Integer for standard User ID | --> | RA_CUSTOMERS_INTERFACE_ALL.CREATED_BY<br>RA_CUSTOMER_PROFILES_INT_ALL.CREATED_BY |
| SYSDATE | --> | RA_CUSTOMERS_INTERFACE_ALL.CREATION_DATE<br>RA_CUSTOMER_PROFILES_INT_ALL.CREATION_DATE |
| Integer for standard User ID | --> | RA_CUSTOMERS_INTERFACE_ALL.LAST_UPDATE_LOGIN<br>RA_CUSTOMER_PROFILES_INT_ALL.LAST_UPDATE_LOGIN |
| CZ_DB_SETTINGS.SETTING_ID='CustomerProfileClassName' | --> | RA_CUSTOMER_PROFILES_INT_ALL.CUSTOMER_PROFILE_CLASS_NAME |
| value of "Y" | --> | RA_CUSTOMER_PROFILES_INT_ALL.CREDIT_HOLD |
| CZ_ADDRESSES.ADDRESS_ID | --> | RA_CUSTOMERS_INTERFACE_ALL.ORIG_SYSTEM_ADDRESS_REF |
| CZ_ADDRESSES.ADDR_LINE1 | --> | RA_CUSTOMERS_INTERFACE_ALL.ADDRESS1 |
| CZ_ADDRESSES.ADDR_LINE2 | --> | RA_CUSTOMERS_INTERFACE_ALL.ADDRESS2 |
| CZ_ADDRESSES.CITY | --> | RA_CUSTOMERS_INTERFACE_ALL.CITY |
| CZ_ADDRESSES.STATE | --> | RA_CUSTOMERS_INTERFACE_ALL.STATE |
| CZ_ADDRESSES.PROVINCE | --> | RA_CUSTOMERS_INTERFACE_ALL.PROVINCE |
| CZ_ADDRESSES.COUNTY | --> | RA_CUSTOMERS_INTERFACE_ALL.COUNTY |
| CZ_ADDRESSES.POSTAL_CODE | --> | RA_CUSTOMERS_INTERFACE_ALL.POSTAL_CODE |
| CZ_ADDRESSES.COUNTRY | --> | RA_CUSTOMERS_INTERFACE_ALL.COUNTRY |
| Function of CZ_ADDRESSES.ORIG_SYS_REF derived from QUOTE_HDR.BILLTO_ADDRESS_ID | --> | SO_HEADERS_INTERFACE_ALL.INVOICE_ADDRESS_ID |

*Table B–1 Data Export to Oracle Applications Source and Interface Tables*

| Oracle Configurator schema Source | -> | Oracle Applications Interface Table(s) |
|---|---|---|
| Function of CZ_ADDRESSES.ORIG_SYS_REF derived from QUOTE_HDR.SHIPTO_ADDRESS_ID | --> | SO_HEADERS_INTERFACE_ALL.SHIP_TO_ADDRESS_ID |
| CZ_ADDRESS_USES.SITE_USER_CODE | --> | RA_CUSTOMERS_INTERFACE_ALL.SITE_USE_CODE |
| CZ_PRICE_GROUPS.CURRENCY | --> | SO_HEADERS_INTERFACE_ALL.CURRENCY_CODE |

# C

# OC SQL*Plus Scripts and Procedures

This Appendix is organized as follows:

## C.1 The DBAdmin Folder

Your Oracle Configurator Developer CD contains a folder, DBAdmin, containing the OC SQL*Plus scripts to perform basic administrative procedures, and the packages that support those scripts. The folder is organized as follows:

***Table C–1 DBAdmin Folder Organization***

| Path | Contents |
|---|---|
| DBAdmin/ | Scripts used when integrating with Oracle Applications |
| DBAdmin/EndUsers/ | Scripts for end user administration. |
| DBAdmin/Lite/ | Scripts for creating/upgrading an Oracle8*i* Lite version of the Oracle Configurator schema. Oracle Applications-unaware. |
| DBAdmin/Server/ | Scripts for creating/upgrading the server version of the Oracle Configurator schema. Oracle Applications-unaware. |

## C.2 Scripts

Oracle Configurator provides SQL*Plus scripts to perform basic administrative procedures. All files are included in the DBAdmin folder on the Oracle Configurator Developer CD.

**Note:** The OC SQL*PLUS scripts provided in the DBAdmin folder on the Oracle Configurator Developer CD are not guaranteed to work for all environments or situations! They can instead be used as a starting point or guide. Oracle advises that you evaluate the appropriateness of each script before running it at your site.

All scripts are run with SQL*Plus in the <OC-scripts> directory, or start SQL*Plus, go to File **-->** Open, navigate to the directory or folder where the OC scripts are located (<OC-scripts>), and click Cancel.

**Caution:** Run OC SQL*Plus scripts from SQL*Plus. **Do not run** OC SQL*Plus scripts from SQL Worksheet.

All arguments for the scripts are listed and described below in Appendix C.2.1, "Script Arguments". The results of these scripts are created in the System space of your Oracle8*i* Enterprise Edition RDBMS unless otherwise specified through script arguments <defaultspace>, <tempspace>, <impdefaultspace>, <imptempspace>, and <indxspace>.

**Caution:** Outstrips output while the script executes can be difficult to interpret as either spurious or indicative of actual failure. Oracle recommends setting echo on and enabling spooling while running these scripts.

*Table C–2 OC SQL*Plus Scripts*

| Script Name | What it does |
|---|---|
| ADD_CONTROL_RECORDS.sql | Inserts standard records into import/export control tables when integrated with Oracle Applications |
| ADD_CONTROL_RECORDS_ONLY.sql | Inserts standard records into import/export control tables when not integrated with Oracle Applications |
| create_colgroups.sql | Creates column groups for replication |
| CREATE_EXP_VIEWS.sql | Creates the 'extraction views' used to extract data for export from the Oracle Configurator schema into the Oracle Applications database |
| CREATE_EXV_VIEWS.sql | Uses the system date as the effective date to create the 'extraction views' used to extract Oracle Applications data for import into the Oracle Configurator schema. If the effective date is to be something other than the system date, this script must be customized to do so |
| CREATE_IMPORT_SCHEMA.sql | Creates the tables used for import |
| create_master_group.sql | Creates a master group for replication, with the tables that are replicated |
| create_mlog_indexes.sql | Applies indices to Oracle8 replication control tables as a performance improvement for replication |
| create_resolvers.sql | Creates packages for custom conflict resolution functions |
| create_snapshot_logs.sql | Creates snapshot logs during creation of the replica |

*Table C–2 OC SQL*Plus Scripts*

| Script Name | What it does |
|---|---|
| CTRA_ADMIN_B.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CTRA_ADMIN_S.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CTRA_ORAAPPS_INTEGRATE_B.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CTRA_ORAAPPS_INTEGRATE_S.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CTRA_UTILS_B.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CTRA_UTILS_S.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_BASE_MGR.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_CF_API_B_80.sql | When creating Oracle Configurator schema in Oracle 8.0.x, CZ_CF_API package compiles with errors and you must load CZ_CF_API_B_80.sql after creating the Oracle Configurator schema. |

*Table C–2 OC SQL*Plus Scripts*

| Script Name | What it does |
|---|---|
| CZ_CF_API_B.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script.<br>Not available in an Oracle 7.3 installation. In Oracle 8.0.x, CZ_CF_API_B.sql commits any open transactions in the session from which you call it and you must load CZ_CF_API_B_80.sql after creating the Oracle Configurator schema. |
| CZ_CF_API_S.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_DEFAULTS.sql | (Lite version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_DEFAULTS.sql | (Server version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_EXPORT.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_GN_MGR.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_IM_MGR.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |

*Table C–2 OC SQL*Plus Scripts*

| Script Name | What it does |
|---|---|
| cz_imgs_upd.sql | Updates the CZ_UI_NODE_PROPS table replacing mixed-case bmp files from 4.1.1 or earlier versions with new lowercase gif files. This should be run by each schema owner |
| CZ_INDEXES.sql | (Lite version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_INDEXES.sql | (Server version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_LC_MGR.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_LITE_TRIGGERS.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| cz_list_price.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_LIST_PRICE_B.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| cz_list_price_package.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |

***Table C–2 OC SQL*Plus Scripts***

| Script Name | What it does |
|---|---|
| CZ_LIST_PRICE_S.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_LITE.sql | Creates standalone Oracle Configurator schema on Oracle8*i* Lite. Used for creating demo databases, only. For creating an Oracle8*i* Lite Oracle Configurator Mobile Database, use replication |
| CZ_MANAGER.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_MGR_INSTALL.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_OM_MGR.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_PACKAGES.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_PK_CONSTRAINTS.sql | (Lite version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_PK_CONSTRAINTS.sql | (Server version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |

*Table C–2 OC SQL*Plus Scripts*

| Script Name | What it does |
|---|---|
| CZ_PR_MGR.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| cz_prc_util_package.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_PRC_UTIL_S.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_PS_COPY.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_PS_MGR.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_QC_MGR.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_REBUILD_INDEXES.sql | Rebuilds tablespace indexes. |
| CZ_REFRESH.sql | Refreshes all configuration models in a production Oracle Configurator schema with changes made in the development schema. |
| CZ_SEQUENCES.sql | (Lite version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |

*Table C–2 OC SQL*Plus Scripts*

| Script Name | What it does |
|---|---|
| CZ_SEQUENCES.sql | (Server version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_SERVER.sql | Creates an Oracle Configurator schema on an Oracle8 server. This script does not require parameters. |
| cz_standalone.sql | (Lite version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| cz_standalone.sql | (Server version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_TABLES.sql | (Lite version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_TABLES.sql | (Server version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_TRIGGERS.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_UI_MGR.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |

***Table C–2 OC SQL*Plus Scripts***

| Script Name | What it does |
|---|---|
| czuindprps.sql | Updates the CZ_UI_NODE_PROPS table replacing <none> values with NULL This should be run once when all schemas are updated |
| CZ_VIEWS.sql | (Lite version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_VIEWS.sql | (Server version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| cz_xf_mgr.sql | (Integration version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| CZ_XF_MGR.sql | (Server version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| czclnup.sql | Displays a list of invalid database objects, attempts to re-compile them, and displays a list of objects that failed the re-compilation attempt |
| DropAppsIntegrate.sql | Drops links and objects from a previous integrated database |
| EndUsers.sql | Interactively implements different end user administration tasks. See also Appendix C.2.7, "Using EndUsers.sql" |
| export_conc_prog.sql | Registers the Oracle Configurator Order Export software for periodic execution by Concurrent Manager |

*Table C–2 OC SQL*Plus Scripts*

| Script Name | What it does |
|---|---|
| FND_STATS.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| generate_support.sql | Generates replication support. |
| GO_IMPORT.sql | PL/SQL script installs Oracle Configurator schema, creates the OC tables and import schema and all Oracle Applications integration, and performs initial import See also Appendix C.2.2, "Using GO_IMPORT.sql" |
| GO_IMPORT_ONLY.sql | PL/SQL script installs Oracle Configurator schema, import tables, and import packages for non-Oracle Applications installations. See also Appendix C.2.4, "Using GO_IMPORT_ONLY.sql" |
| GRANT_TO_ROLE.sql | PL/SQL script grants access privileges to the default SPX_USER role (or to whatever customized role is stored in CZ_DB_SETTINGS) |
| IMP_AC_KRS_B.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_AC_KRS_S.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_AC_MAIN_B.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_AC_MAIN_S.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |

*Table C–2 OC SQL*Plus Scripts*

| Script Name | What it does |
|---|---|
| IMP_AC_XFR_B.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_AC_XFR_S.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_ALL_B.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_ALL_ONLY_B.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_ALL_ONLY_S.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_ALL_S.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_EXTRACT_B.sq | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_EXTRACT_S.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |

*Table C–2 OC SQL*Plus Scripts*

| Script Name | What it does |
|---|---|
| IMP_IM_KRS_B.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_IM_KRS_S.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_IM_MAIN_B.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_IM_MAIN_S.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_IM_XFR_B.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_IM_XFR_S.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_PR_KRS_B.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_PR_KRS_S.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |

*Table C–2 OC SQL*Plus Scripts*

| Script Name | What it does |
|---|---|
| IMP_PR_MAIN_B.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_PR_MAIN_S.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_PR_XFR_B.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_PR_XFR_S.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_PS_NODE_B.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_PS_NODE_ONLY_B.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_PS_NODE_ONLY_S.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| IMP_PS_NODE_S.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |

*Table C–2 OC SQL*Plus Scripts*

| Script Name | What it does |
|---|---|
| import_conc_prog.sql | Registers the SellingPoint Import software for periodic execution by Concurrent Manager |
| ImportSingleBill.sql | Interactively imports a single bill of material by ORGANIZATION_ID and TOP_ITEM_ID |
| InstAppsIntegrate.sql | This script creates the import schema, creates views and grants for import, installs all the integration packages and populates the extraction/import order control table |
| InstAppsIntegrateViaLink.sql | Like InstAppsIntegrate, this script creates the import schema, creates views and grants for import, installs all the integration packages and populates the extraction/import order control table. The difference is that it uses database links to connect to an Oracle Applications database that is not in the same instance. See also Appendix C.2.3, "Using InstAppsIntegrateViaLink.sql" |
| LoadAllBills.sql | Finds all current entries in the Oracle Applications tables BOM_EXPLOSIONS and SO_PRICE_LISTS and identifies them in the SellingPoint tables CZ_XFR_PROJECT_BILLS and CZ_XFR_PRICE_GROUP so that their data will be loaded in the next Import run. See also Appendix C.2.5, "Using LoadAllBills.sql" |
| ospc_messages_lite.sql | (Lite version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| ospc_messages.sql | (Server version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| order_status_conc_prog.sql | Registers the Oracle runtime configurator order-status-retrieval software for periodic execution by Concurrent Manager |
| register_resolvers.sql | Registers custom conflict resolution functions |
| rep_admin.sql | Sets permissions for each replication end user |

***Table C–2 OC SQL*Plus Scripts***

| Script Name | What it does |
|---|---|
| rep_grants.sql | Grants necessary replication privileges to the role of end users |
| rep_prop_setup.sql | Creates a replication admin user and registers that user as the propagator |
| rep_setup.sql | Creates a replication group to configure the replication of Oracle Configurator schema tables with Oracle Lite. Adds snapshot logs, changes triggers, and creates package for replication utilities |
| rep_triggers.sql | Recreates table triggers with logic to support replication by end users |
| RunGenImport.sql | PL/SQL script executes the software for a generic import and displays the results |
| RunImport.sql | PL/SQL script executes the Import software so that it emits a report on its results |
| TimeStampTriggers.class | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| TimeStampTriggers.java | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| UPGRADE_CONFIGS.sql | Script for upgrading configuration tables from 4.1.1 (12f) to 4.2 (14c) |
| UPGRADE_DEFAULTS.sql | (Lite version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| UPGRADE_DEFAULTS.sql | (Server version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |

*Table C–2 OC SQL*Plus Scripts*

| Script Name | What it does |
|---|---|
| UPGRADE_DROP_OBJECTS.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| UPGRADE_EXPRESSIONS.sql | (Lite version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| UPGRADE_EXPRESSIONS.sql | (Server version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| UPGRADE_LITE.sql | Script for upgrading an Oracle Configurator Mobile Database (Oracle8*i* Lite) from 4.1.1 (12f) to 4.2 (14c) |
| UPGRADE_SEQUENCES.sql | (Lite version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| UPGRADE_SEQUENCES.sql | (Server version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| UPGRADE_SERVER.sql | Script for upgrading an Oracle Configurator schema (Oracle8*i* Enterprise Edition) from 4.1.1 (12f) to 4.2 (14c) |
| UPGRADE_TABLES.sql | (Lite version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |

*Table C–2 OC SQL*Plus Scripts*

| Script Name | What it does |
|---|---|
| UPGRADE_TABLES.sql | (Server version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| UPGRADE_TABLES_PREPARE.sql | (Lite version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| UPGRADE_TABLES_PREPARE.sql | (Server version) This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| UpgradeExpressions.class | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| UpgradeExpressions.java | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| UpgradeSequences.class | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| UpgradeSequences.java | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| UpgradeTables.class | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |

*Table C–2 OC SQL*Plus Scripts*

| Script Name | What it does |
|---|---|
| UpgradeTables.java | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| user1.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| user2.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| user3.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| user4.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| user5.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |
| user6.sql | This file is delivered as part of Oracle Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script |

## C.2.1 Script Arguments

This book refers to the Oracle Configurator schema and its DBOwner as <oc>. This manual refers to the instance hosting the Oracle Configurator schema that contains

<oc> as <ocsid>. This manual refers to the machine where <ocsid> is established as <ocdbhost>. For Release 11*i*, several parameters that refer to the Oracle Configurator schema are actually the same value as the Oracle Applications database: <oc>=<apps>, <ocsid>=<appssid>, and ocdbhost>=<appsdbhost>. These arguments for the Oracle Configurator schema are presented differently for better differentiation in discussions involving generic import scripts.

Wherever scripts take arguments, the following conventions are used:

***Table C–3 Argument Tokens for OC SQL*Plus Scripts***

| Token | Description |
|---|---|
| <appspass> | Password for <apps> |
| <apps> | Owner of the Oracle Applications database, usually ’apps’ |
| <appsdbhost> | Hostname for the system on which the Oracle Applications Oracle8i instance is running. |
| <appssid> | Instance name for the Oracle8i instance on which the Oracle Applications database is installed |
| <dba> | Username of DBA with DBA privileges. |
| <dbapass> | Password for <dba>. |
| <DBAUser> | Username of the Oracle Configurator administrator who has DBA privileges. |
| <DBAUserpass> | Password for <DBAUser> |
| <defaultspace> | Default permanent tablespace to be used by <oc>, <ocdev>, and end users for permanent tables. The value of <defaultspace> is the tablespace specified for user data when Oracle8*i* Enterprise Edition was installed. The Oracle8*i* Enterprise Edition installation default value for this tablespace is USERS. |
| <expdump> | Owner of the schema from which a dump (.dmp) file is exported. |
| <imp> | Owner of the Configurator Import/Integration source database |
| <impdefaultspace> | Default tablespace to be used by <imp> for permanent tables |
| <impdump> | Owner of the empty schema into which a dump (.dmp) file is imported. |
| <impdumppass> | Password for <impdump> |

*Table C–3 Argument Tokens for OC SQL*Plus Scripts*

| Token | Description |
|---|---|
| <imppass> | Password for <imp> |
| <imptempspace | Temporary tablespace to be used by <imp> for temporary tables |
| <indxspace> | Tablespace to be used by <oc>, <ocdev>, and end users for indexes |
| <oc> | Owner of the Oracle Configurator schema |
| <ocdbhost> | Hostname for the system on which the Oracle Configurator schema Oracle8*i* Enterprise Edition instance is running. |
| <ocdev> | Oracle Configurator Developer user |
| <ocdevpass> | Password for <ocdev> |
| <ocpass> | Password for <oc> |
| <ocsid> | Name for the Oracle8*i* Enterprise Edition instance on which the Oracle Configurator schema is installed. If on the same installation of Oracle8*i* Enterprise Edition as where the Oracle Applications database is installed, <ocsid>=<appssid>. |
| <OC-scripts> | Directory where SQL*Plus scripts used to set up the Oracle Configurator schema are located |
| <SPX_USER> | Name of database role for Oracle SellingPoint application end users for grants. The default value in CZ_DB_SETTINGS table in Oracle Configurator schema is "SPX_USER". |
| <oe> | Owner of Oracle Applications order-entry tables |
| <oepass> | Password for <oe> |
| <port> | Listener port number (1521 in most locations) |
| <repadmin> | Name of the Replication Administration user creating the replication objects in the Oracle Configurator schema. Has access to the packages dbms_repcat, etc. |
| <repadminpass> | Password for <repadmin> |
| <repgroup> | Name of the Replication Group in the Oracle Configurator schema |
| <tempspace> | Temporary tablespace to be used by <oc>, <ocdev>, and end users for temporary tables. The value of <tempspace> is the tablespace specified for temporary data when Oracle8*i* Enterprise Edition was installed. The Oracle8*i* Enterprise Edition installation default value for this tablespace is TEMP. |

*Table C–3 Argument Tokens for OC SQL*Plus Scripts*

| Token | Description |
|---|---|
| <TNSalias> | The TNS alias is the service name you enter in Net8 Easy Config that represents the full service name to which you connect for database access. |

## C.2.2 Using GO_IMPORT.sql

### C.2.2.1 Parameters

The parameters for GO_IMPORT.sql are specified as arguments. See Appendix C.2.1, "Script Arguments" for descriptions of the argument tokens. The parameters are listed in the order in which they must be specified when executing GO_IMPORT.sql.

- &1 - Name of Oracle Configurator schema user (<oc>, <ocdev>, etc.)
- &2 - Password for &1 (<ocpass>, <ocdevpass>, etc.)
- &3 - Owner of the SellingPoint Import/Integration database <imp>
- &4 - Password for <imp> <imppass>
- &5 - Owner of the Oracle Applications database, usually 'apps' <apps>
- &6 - Password for <apps>: <apppass>
- &7 - TNS entry name for all databases <ocsid>

### C.2.2.2 Restrictions

The restrictions on GO_IMPORT.sql are that the script:

**a.** assumes that 2 empty users (<oc> and <imp>) are available with sufficient privileges (connect, resource)

**b.** assumes all schemas should share the same database instance

**c.** should be run from <oc>

### C.2.2.3 Actions

When you execute GO_IMPORT.sql, it does the following:

**a.** creates SellingPoint online schema <oc>

**b.** connects &3/&4@&7

**c.** creates import schemas

**d.** connect &5/&6@&7

**e.** grants selects on the required tables and execute on the 'bom_exploder' procedure from Oracle Applications for &1

**f.** connect &1/&2@&7

**g.** creates 'extraction views' &5

**h.** populates CZ_XFR_PROJECT_BILLS with all top-level bills from EXV_PS_NODE (our view for bom_explosions) with explosion_type='OPTIONAL'

**i.** populates CZ_XFR_PRICE_LISTS with all price_list_ids from EXV_PRICE_LISTS (our view for SO_PRICE_LISTS) defaulting not to import prices

**j.** creates synonym for the 'bom_exploder' procedure

**k.** installs (compiles) all the import packages

**l.** installs (compiles) all the export packages

**m.** adds extraction/import order control records to CZ_XFR_TABLE

**n.** adds several records to CZ_DB_SETTINGS

| setting_id | section_name | data_type | value | desc_text |
|---|---|---|---|---|
| 17 | DATABASE_OWNERS | 4 | &5 | Name of the Oracle Applications owner <apps> |
| 18 | DATABASE_OWNERS | 4 | &5 | Name of the Oracle Applications Order Entry owner <apps> |

**o.** runs import (RunImport.sql)

**p.** displays the import results from CZ_XFR_RUN_RESULT table

## C.2.3 Using InstAppsIntegrateViaLink.sql

### C.2.3.1 Parameters

The parameters for InstAppsIntegrateViaLink.sql are specified as arguments. See Appendix C.2.1, "Script Arguments" for descriptions of the argument tokens. The parameters are listed in the order in which they must be specified when executing InstAppsIntegrateViaLink.sql.

- &1 - Owner of the Oracle Configurator schema <oc>
- &2 - Password for <oc> <ocpass>
- &3 - Owner of the SellingPoint Import/Integration database <imp>
- &4 - Password for <imp> <imppass>
- &5 - TNS entry for the Oracle Configurator schema <ocsid>
- &6 - Owner of the Oracle Applications database, usually ’apps’ <apps>
- &7 - Password for <apps> <appspass>
- &8 - Name of database link
- &9 - Instance name for the Oracle8 instance on which the Oracle Applications database is installed <appssid>
- &10 - Hostname for the system on which the Oracle Applications Oracle8 instance is running <appsdbhost>
- &11 Listener port number (1521 in most locations) <port>

### C.2.3.2 Restrictions

The restrictions on InstAppsIntegrateViaLink.sql are that the script:

**a.** assumes that the Oracle Configurator schema has already been created in <oc> and user <imp> is available with sufficient privileges

**b.** requires the GLOBAL_NAMES be set to ‘FALSE’ while running InstAppsIntegrateViaLink.sql so that the script can create the synonyms it needs. If this is set to ’FALSE’, the name of the database link can be any arbitrary name. If this must be set to ’TRUE’, the name of the database link must be the same as the remote database.

**c.** should be run from <oc>

### C.2.3.3 Actions

When you execute InstAppsIntegrateViaLink.sql, it does the following:

a. creates database link &8 pointing to <apps> and <oe>

b. connect to &6 identified by &7

c. creates synonyms to reference Oracle Applications tables via database links synonym for the ’bom_exploder’ procedure

d. connect &3/&4@&5

e. creates import schema

f. connect &1/&2@&5

g. creates ’extraction views’

h. installs (compiles) all the import packages

i. installs (compiles) the export package

j. adds extraction/import order control records to CZ_XFR_TABLE

k. adds several records to CZ_DB_SETTINGS

| setting_id | section_name | data_type | value | desc_text |
|---|---|---|---|---|
| 17 | DATABASE_OWNERS | 4 | &6 | Name of the Oracle Applications owner <apps> |
| 18 | DATABASE_OWNERS | 4 | &6 | Name of the Oracle Applications Order Entry owner <apps> |
| APPSLINK | DATABASE_OWNERS | 4 | &8 | Link used for connecting to a remote database server for Oracle Applications <appssid> |

## C.2.4 Using GO_IMPORT_ONLY.sql

### C.2.4.1 Parameters

The parameters for GO_IMPORT_ONLY.sql are specified as arguments. See Appendix C.2.1, "Script Arguments" for descriptions of the argument tokens. The parameters are listed in the order in which they must be specified when executing GO_IMPORT_ONLY.sql. See also Section 4.3.2, "Run Generic Import" on page 4-27.

- &1 - Owner of the Oracle Configurator schema <oc>
- &2 - Password for <oc> <ocpass>
- &3 - Owner of the SellingPoint Import/Integration database <imp>
- &4 - Password for <imp> <imppass>
- &5 - Instance name for the Oracle8 instance on which Oracle Configurator schema is installed <ocsid>

### C.2.4.2 Restrictions

The restrictions on GO_IMPORT_ONLY.sql are that the script:

a. assumes that 2 empty users (<oc> and <imp>) are available with sufficient privileges (connect, resource)

b. both schemas should share the same database instance

c. should be run from <oc>

### C.2.4.3 Actions

When you execute GO_IMPORT_ONLY.sql, it does the following:

a. creates SellingPoint online schema

b. creates import schema

c. installs (compiles) all the import packages

d. adds extraction/import order control records to CZ_XFR_TABLE

e. adds several records to CZ_DB_SETTINGS which are not included yet into the SPX_server.sql script

## C.2.5 Using LoadAllBills.sql

### C.2.5.1 Parameters

LoadAllBill.sql has no parameters.

### C.2.5.2 Restrictions

The restrictions on LoadAllBills.sql are that the script:

a. assumes that InstallAppsIntegrate.sql has already been run or at least that the 'extraction views' have somehow been created

b. should be run from <oc>

### C.2.5.3 Actions

When you execute LoadAllBills.sql, it does the following:

a. populates CZ_XFR_PROJECT_BILLS with all top-level bills from EXV_PS_NODE (our view for bom_explosions) with explosion_type='OPTIONAL'

b. populates CZ_XFR_PRICE_LISTS with all price_list_ids from EXV_PRICE_LISTS (our view for SO_PRICE_LISTS) defaulting not to import prices

## C.2.6 Using GRANT_SELECT_FOR.sql

### C.2.6.1 Parameters

The parameters for GRANT_SELECT_FOR.sql are specified as arguments. See Appendix C.2.1, "Script Arguments" on page C-19 for descriptions of the argument tokens. The parameters are listed in the order in which they must be specified when executing GRANT_SELECT_FOR.sql. See also Section 2.2.2, "Create Users and Responsibilities" on page 2-3.

- &1 -- Owner of the Oracle Configurator schema <oc>

### C.2.6.2 Actions

When you execute GRANT_SELECT_FOR.sql, it does the following:

a. Grants 'select' access to <oc> for importing data from Oracle Applications schema.

### C.2.6.3 Restrictions

The restrictions on GRANT_SELECT_FOR.sql are that the script:

a. Must be run by <apps> owner.

## C.2.7 Using EndUsers.sql

### C.2.7.1 Parameters

There are no parameters for EndUsers.sql. Instead, EndUsers.sql is an interactive script prompting you to select from a list of choices, and then prompting you for parameters.

#### C.2.7.1.1 (1) Display All Available End User Login Names

The script returns the end_user_id, end_user_org_id, and login_name of every end users listed in the CZ_END_USERS table of the Oracle Configurator schema.

#### C.2.7.1.2 (2) Add End User

The script prompts you for an end user login name and allowable discount (number), which are then inserted in the CZ_END_USERS table along with an end_user_id incremented from the highest value in the table. Any added end user is automatically assigned the user_group_id END_USER, which is granted the <SPX_USER> role.

Unless explicitly specified (see Appendix C.2.7.1.3, "(3) Enable End User as a Database User"), the user added does not have database access. This is equivalent to setting the CZ_DB_SETTING AUTOCREATE_IMPORTED_USERS to ‘NO’.

#### C.2.7.1.3 (3) Enable End User as a Database User

The script prompts you for an end user login name and password.

Display available end users and then run this against an end user in the CZ_END_USERS table to enable any of them as a database user.

Any end user of the Oracle runtime configurator needing access to the Oracle Configurator schema, must be enabled as a database user. This choice is used to manipulate which end users in the CZ_END_USERS table have access to the Oracle Configurator schema if the imported users weren’t automatically enabled (i.e., DB_SETTING AUTOCREATE_IMPORTED_USERS=‘NO’).

#### C.2.7.1.4 (4) Enable All End Users (not recommended)

The script enables all end users listed in the CZ_END_USERS table as database users. After an import, the number of records in this table could be very large with all Oracle Applications users included, many of whom should not have access to the DBMS running the Oracle Configurator schema.

#### C.2.7.1.5 (5) Change the Default Tablespace Name

The script displays the current default tablespace name, which is the value of SpxDefaultTablespace in the CZ_DB_SETTINGS table. The script then prompts you to enter a new default tablespace name. This new name is then inserted as the value of SpxDefaultTablespace in the CZ_DB_SETTINGS table.

#### C.2.7.1.6 (6) Change the Temporary Tablespace Name

The script displays the current temporary tablespace name, which is the value of SpxTemporaryTablespace in the CZ_DB_SETTINGS table. The script then prompts you to enter a new temporary tablespace name. This new name is then inserted as the value of SpxTemporaryTablespace in the CZ_DB_SETTINGS table.

#### C.2.7.1.7 (CTRL-c) Return to Previous Screen

Returns you to the previous prompt as you step through EndUsers.sql.

## C.2.8 Summary of Actions by OC SQL*Plus Scripts

*Table C–4 Summary of Actions by OC SQL*Plus Scripts*

| Action | GO_IMPORT.sql | GO_IMPORT_ONLY.sql | InstAppsIntegrateViaLink.sql | InstAppsIntegrate.sql | LoadAllBills.sql | RunImport.sql | GRANT_TO_ROLE.sql | CZ_SERVER.sql |
|---|---|---|---|---|---|---|---|---|
| Create Oracle Configurator schema (<oc>) | X | X | | | | | | X |
| Create integration schema (<imp>) | X | | X | X | | | | |
| Grant privileges for Oracle Applications | X | | | X | | | | |
| Create Database Links | | | X | | | | | |
| Create Extraction Views | X | | X | X | | | | |

*Table C–4 Summary of Actions by OC SQL*Plus Scripts*

| Action | GO_IMPORT.sql | GO_IMPORT_ONLY.sql | InstAppsIntegrateViaLink.sql | InstAppsIntegrate.sql | LoadAllBills.sql | RunImport.sql | GRANT_TO_ROLE.sql | CZ_SERVER.sql |
|---|---|---|---|---|---|---|---|---|
| Install Import Tables | X | X | X | X | | | | |
| Install Export Tables | X | | X | X | | | | |
| Add records to CZ_DB_SETTINGS | X | X | X | X | | | | |
| Populate CZ_XFR_ tables for direct import from Oracle Applications | X | | X | X | X | | | |
| Populate CZ_XFR_ tables for generic import | X | X | X | X | | | | |
| Grant privileges to role | X | X | X | X | | | X | |
| Run data import | X | | | | | X | | |

## C.3 Procedures

The procedures are defined in packages. The scripts execute procedures by calling these packages. The packages listed below are located in the DBAdmin folder on the Oracle Configurator Developer CD, unless otherwise indicated as being in one of the subdirectories in DBAdmin/.

These procedures report problems or information using the CZ_UTILS.REPORT function, which routes informational messages to the CZ_DB_LOGS table and to the SQL*Plus display.

***Table C–5 Packages Used in OC Administrative Tasks***

| Package Name | Description |
|---|---|
| create_resolvers.sql | Package of custom conflict resolution functions |
| CTRA_ADMIN_B.sql | Body of administration package CTRA_ADMIN |
| CTRA_ADMIN_S.sql | Specification of administration package CTRA_ ADMIN |
| CTRA_ORAAPPS_INTEGRATE_B.sql | Body of Oracle Applications Integration package CTRA_ORAAPPS_INTEGRATE |
| CTRA_ORAAPPS_INTEGRATE_S.sql | Specification of Oracle Applications Integration package CTRA_ORAAPPS_INTEGRATE |
| CTRA_UTILS_B.sql | Body of utility package CTRA_UTILS |
| CTRA_UTILS_S.sql | Specification of utility package CTRA_UTILS |
| CZ_ATP_UTIL_B.sql | Body of Oracle Applications Integration package CZ_ATP_UTIL which uses functions from other modules in Oracle Applications in order to determine the availability of items configured and quoted |
| CZ_ATP_UTIL_S.sql | Specification of utility package CZ_ATP_UTIL |
| CZ_DEFAULTS.sql | Inserts initial data into Oracle Configurator schema tables (in DBAdmin/Server/) |
| CZ_DEFAULTS.sql | Inserts initial data into the Oracle Configurator schema tables (in DBAdmin/Lite/) |
| CZ_EXPORT.sql | Package for Oracle Applications Order Export from the Oracle Configurator schema |
| CZ_GN_MGR.sql | Manager package (purge, clear, validate) for 'general' tables (i.e., everything not handled by later MGR packages) (in DBAdmin/Server/) |
| CZ_IM_MGR.sql | Manager package for Item-Master subschema (in DBAdmin/Server/) |
| CZ_INDEXES.sql | Creates indexes on Oracle Configurator schema tables (in DBAdmin/Server/) |
| CZ_INDEXES.sql | Creates indexes in Oracle Configurator schema (in DBAdmin/Lite/) |
| CZ_LC_MGR.sql | Manager package for LCE (compiled logic) subschema (in DBAdmin/Server/) |

*Table C–5 Packages Used in OC Administrative Tasks*

| Package Name | Description |
|---|---|
| cz_list_price.sql | Pricing implementation, body of package providing list prices. |
| CZ_LIST_PRICE_B.sql | Body of Oracle Applications Integration package CZ_LIST_PRICE which is used to retrieve list prices from other modules in Oracle Applications for configured or quoted items |
| cz_list_price_package.sql | Pricing implementation, specification for package providing list prices. |
| CZ_LIST_PRICE_S.sql | Specification of utility package CZ_LIST_PRICE |
| CZ_LITE_TRIGGERS.sql | Attaches timestamp/user triggers to Oracle Configurator schema tables (in DBAdmin/Lite/) |
| CZ_MANAGER.sql | Umbrella manager package; its methods call all of the sub-package methods (in DBAdmin/Server/) |
| CZ_MGR_INSTALL.sql | Master script for installing the MGR packages (in DBAdmin/Server/) |
| CZ_OM_MGR.sql | Manager package for opportunity management (in DBAdmin/Server/) |
| CZ_PACKAGES.sql | Master script for installing all Oracle Configurator schema packages (in DBAdmin/Server/) |
| CZ_PK_CONSTRAINTS.sql | Defines primary keys on Oracle Configurator schema tables (in DBAdmin/Lite/) |
| CZ_PK_CONSTRAINTS.sql | Script creates primary key constraints on Oracle Configurator schema tables (in DBAdmin/Server/) |
| CZ_PR_MGR.sql | Manager package for pricing subschema (in DBAdmin/Server/) |
| cz_prc_util.sql | Utility package for pricing (body) |
| CZ_PRC_UTIL_B.sql | Body of Oracle Applications Integration package CZ_PRC_UTIL which contains utility routines for supporting price interrogation and discounting |
| cz_prc_util_package.sql | Utility package for pricing (specification) |
| CZ_PRC_UTIL_S.sql | Specification of utility package CZ_PRC_UTIL |
| CZ_PS_COPY.sql | Package with support functions for project copy, copy with rules, rule-folder copy, etc. (in DBAdmin/Server/) |

*Table C–5 Packages Used in OC Administrative Tasks*

| Package Name | Description |
|---|---|
| CZ_PS_MGR.sql | Manager package for product-structure subschema (in DBAdmin/Server/) |
| CZ_QC_MGR.sql | Manager package for quote/config subschema (in DBAdmin/Server/) |
| CZ_SEQUENCES.sql | Creates SEQUENCE objects (in DBAdmin/Lite/) |
| CZ_SEQUENCES.sql | Script installs SEQUENCE objects into Oracle Configurator schema (in DBAdmin/Server/) |
| cz_standalone.sql | Oracle8*i* Enterprise Edition server script creating database objects strictly for non-Oracle Applications integrated work, i.e., creates a version of the Oracle Applications table FND_NEW_MESSAGES (in DBAdmin/Server/) |
| CZ_STANDALONE.sql | Oracle8*i* Lite script creating database objects strictly for non-Oracle Applications integrated work, i.e., creates a version of the Oracle Applications table FND_NEW_MESSAGES (in DBAdmin/Lite) |
| CZ_TABLES.sql | Creates Oracle Configurator schema tables (in DBAdmin/Lite/) |
| CZ_TABLES.sql | Script installs tables into Oracle Configurator schema database (in DBAdmin/Server/) |
| CZ_TRIGGERS.sql | Script attaches trigger procedures to tables in Oracle Configurator schema (in DBAdmin/Server/) |
| CZ_UI_MGR.sql | Manager package for User-interface subschema (in DBAdmin/Server/) |
| CZ_VIEWS.sql | Creates Oracle Configurator schema views (in DBAdmin/Lite/) |
| CZ_VIEWS.sql | Script installs views into Oracle Configurator schema (in DBAdmin/Server/) |
| CZ_XF_MGR.sql | Manager package for transfer (Import, export, integration) subschema (in DBAdmin/Server/) |
| FND_STATS.sql | Standalone routine used for performance tuning. This routine is a replacement for the Oracle Applications package FND_STATS. |

*Table C–5 Packages Used in OC Administrative Tasks*

| Package Name | Description |
|---|---|
| IMP_AC_KRS_B.sql | Body of import package for resolving keys in the Account/Address subschema. |
| IMP_AC_KRS_S.sql | Specification of import package for resolving keys in the Account/Address subschema |
| IMP_AC_MAIN_B.sql | Body of main import package for the Account/Address subschema. |
| IMP_AC_MAIN_S.sql | Specification of main import package for the Account/Address subschema |
| IMP_AC_XFR_B.sql | Body of import data-transfer package for the Account/ Address subschema |
| IMP_AC_XFR_S.sql | Specification of import data-transfer package for the Account/Address subschema |
| IMP_ALL_B.sql | Body of master import package for Oracle Applications integration. This package is a script that must be invoked with one argument: the name of the import schema owner <imp>. |
| IMP_ALL_S.sql | Specification of master import package for Oracle Applications integration |
| IMP_ALL_ONLY_B.sql | Body of master import package for generic import - not with Oracle Applications. This package is a script that must be invoked with one argument: the name of the import schema owner <imp>. |
| IMP_ALL_ONLY_S.sql | Specification of master import package for generic import - not with Oracle Applications |
| IMP_EXTRACT_B.sql | Body of package for extracting data from Oracle Applications. This package is a script that must be invoked with one argument: the name of the import schema owner <imp>. |
| IMP_EXTRACT_S.sql | Specification of package for extracting data from Oracle Applications |
| IMP_IM_KRS_B.sql | Body of import package for resolving keys in the Item-Master subschema. This package is a script that must be invoked with one argument: the name of the import schema owner <imp>. |
| IMP_IM_KRS_S.sql | Specification of import package for resolving keys in the Item-Master subschema |

*Table C–5 Packages Used in OC Administrative Tasks*

| Package Name | Description |
|---|---|
| IMP_IM_MAIN_B.sql | Body of main import package for the Item-Master subschema. This package is a script that must be invoked with one argument: the name of the import schema owner <imp>. |
| IMP_IM_MAIN_S.sql | Specification of main import package for the Item-Master subschema |
| IMP_IM_XFR_B.sql | Body of import data-transfer package for the Item-Master subschema. This package is a script that must be invoked with one argument: the name of the import schema owner <imp>. |
| IMP_IM_XFR_S.sql | Specification of import data-transfer package for the Item-Master subschema |
| IMP_PR_KRS_B.sql | Body of import package for resolving keys in the Pricing subschema. This package is a script that must be invoked with one argument: the name of the import schema owner <imp>. |
| IMP_PR_KRS_S.sql | Specification of import package for resolving keys in the Pricing subschema |
| IMP_PR_MAIN_B.sql | Body of main import package for the Pricing subschema. This package is a script that must be invoked with one argument: the name of the import schema owner <imp>. |
| IMP_PR_MAIN_S.sql | Specification of main import package for the Pricing subschema |
| IMP_PR_XFR_B.sql | Body of import data-transfer package for the Pricing subschema. This package is a script that must be invoked with one argument: the name of the import schema owner <imp>. |
| IMP_PR_XFR_S.sql | Specification of import data-transfer package for the Pricing subschema |
| IMP_PS_NODE_B.sql | Body of import package for transferring Apps BOM data into the Oracle Configurator schema. This package is a script that must be invoked with one argument: the name of the import schema owner <imp>. |
| IMP_PS_NODE_S.sql | Specification of import package for transferring Apps BOM data into the Oracle Configurator schema |

***Table C–5 Packages Used in OC Administrative Tasks***

| Package Name | Description |
|---|---|
| IMP_PS_NODE_ONLY_B.sql | Body of import package for transferring imported product structure related data into the Oracle Configurator schema without Oracle Applications. This package is a script that must be invoked with one argument: the name of the import schema owner <imp>. |
| IMP_PS_NODE_ONLY_S.sql | Specification of import package for transferring imported product structure related data into the Oracle Configurator schema without Oracle Applications |
| ospc_messages.sql | Inserts OC messages into FND_NEW_MESSAGES (in DBAdmin/Server/) |
| ospc_messages_lite.sql | Inserts OC messages into FND_NEW_MESSAGES (in DBAdmin/Lite) |
| TimeStampTriggers.class | Compiled class file for Java timestamp/user triggers (in DBAdmin/Lite/) |
| TimeStampTriggers.java | Source text for Java timestamp/user triggers (in DBAdmin/Lite/) |
| UPGRADE_DEFAULTS.sql | In upgrade, alters the initial data to current rev (in DBAdmin/Lite/) |
| UPGRADE_DEFAULTS.sql | Upgrades initial data from 4.1.1 to 4.2 values (in DBAdmin/Server/) |
| UPGRADE_DROP_OBJECTS.sql | Drops objects as part of 4.1.1 to 4.2 upgrade (in DBAdmin/Server/) |
| UPGRADE_EXPRESSIONS.sql | Upgrades expressions from 4.1/4.1.1 to the ’advanced expressions’ of 4.2. (in DBAdmin/Server/) |
| UPGRADE_EXPRESSIONS.sql | Restructures expressions for 4.1.1 to 4.2 upgrade (in DBAdmin/Lite/) |
| UPGRADE_SEQUENCES.sql | Creates and alters sequences for upgrade (uses Java files) (in DBAdmin/Lite/) |
| UPGRADE_SEQUENCES.sql | Upgrades sequences for 4.1.1 to 4.2 (in DBAdmin/Server/) |
| UPGRADE_TABLES.sql | Upgrades 4.1/4.1.1 tables to 4.2 (in DBAdmin/Lite/) |

*Table C–5 Packages Used in OC Administrative Tasks*

| Package Name | Description |
|---|---|
| UPGRADE_TABLES.sql | Redoes tables for 4.1.1 to 4.2 upgrade (in DBAdmin/Server/) |
| UPGRADE_TABLES_PREPARE.sql | Ancillary script for tables upgrade (in DBAdmin/Lite/) |
| UPGRADE_TABLES_PREPARE.sql | Script performs preparation for upgrading tables from 4.1.1 to 4.2 (in DBAdmin/Server/) |
| UpgradeExpressions.class | Compiled Java code to upgrade Expression tables to 4.2 (in DBAdmin/Lite/) |
| UpgradeExpressions.java | Java source text for upgrading Expression tables (in DBAdmin/Lite/) |
| UpgradeSequences.class | Compiled Java code for upgrading SEQUENCES (in DBAdmin/Lite/) |
| UpgradeSequences.java | Java source for upgradesequences.class (in DBAdmin/Lite/) |
| UpgradeTables.class | Compile Java code for upgrading tables (in DBAdmin/Lite/) |
| UpgradeTables.java | Java source for UpgradeTables.class (in DBAdmin/Lite/) |

# Glossary of Terms

This glossary for Oracle Configurator is followed by a Glossary of Acronyms

### Active Model

The part of Oracle Configurator runtime architecture that processes model structure and rules to create configurations. Interfaces dynamically with the end user Active UI and data.

### Active User Interface

The part of Oracle Configurator runtime architecture that provides the graphical views necessary to create configurations interactively. Interfaces with the Active Model and data to give users access to customer requirements gathering, product selection, and customer-centric extensions.

### Application Architecture

The software structure of an application at runtime. Architecture affects how an application is used, maintained, extended, and changed.

### Architecture

The software structure of a system. Architecture affects how a system is used, maintained, extended, and changed. See also Application Architecture.

### Beta

An external release, delivered as an installable application, and subject to system, validation, and acceptance testing. Specially selected and prepared end users may participate in beta testing.

### Bill of Material

A list of component items associated with a parent item (assembly) and information about how each item relates to the parent item.

### BOM

See Bill of Material.

### BOM Item

The nodes imported into the Oracle Configurator Developer Model that correspond to an Oracle BOM.

### BOM Model

The imported Model node in the Oracle Configurator Developer that corresponds to Standard Model in an Oracle BOM.

### BOM OptionClass

The imported Model node in the Oracle Configurator Developer that corresponds to Option Class in an Oracle BOM.

### BOM StandardItem

The imported Model node in the Oracle Configurator Developer that corresponds to Standard Item in an Oracle BOM.

### Boolean Expression

An element of a component in the Model that has two options: true or false.

### Bug

See Defect.

### Build

A specific instance of an application during its construction. A build must have an install early in the project so that application implementers can unit test their latest work in the context of the entire available application.

### CIO

See Oracle Configuration Interface Object.

### Client

A runtime program using a server to access functionality shared with other clients.

### Comparison Rule

An Oracle Configurator Developer rule type to establish a relationship that determines the selection state of a logical item (option, boolean feature, or list-of-options feature) based on a comparison of two numeric values (numeric features, totals, resources, option counts, or numeric constants). The numeric values being compared can be computed or they can be discrete intervals in a continuous numeric input.

### Compatibility Rule

An Oracle Configurator Developer rule type to establish a relationship among features in the Model that specifies the allowable combinations of options. See also, Property-based Compatibility Rule.

### Compatibility Table

A type of compatibility relationship where the allowable combination of options are explicitly enumerated.

### Component

Represents a configurable element in a product. An element of the Model structure, typically containing features. May correspond to one screen of selections in an Oracle runtime configurator.

### Component Set

An element of the Model that contains a number of components of the same type, where each component of the set is independently configured.

### Configuration

A specific set of specifications for a product, resulting from selections made in an Oracle runtime configurator.

### Configuration Model

The model structure and rules-based content of an Oracle runtime configurator. The configuration model is constructed and maintained using Oracle Configurator Developer, and is interpreted at runtime by the Active Model.

### Configuration Rules

The Oracle Configurator Developer logic rules and numeric rules available for defining configurations.

### Configurator

The part of an application that provides custom configuration capabilities.

### Constraint Rule

An Oracle Configurator Developer rule type to create a logical relationship among features and options. See also Rules.

### Contributes to

An Oracle Configurator Developer numeric rule type for accumulating a total value.

### Consumes from

An Oracle Configurator Developer numeric rule type for specifying the quantity of a resource used.

### CRM

Customer Relationship Management. The aspect of the enterprise that involves contact with customers, from lead generation to support services.

### Customer

The person or persons for whom products are configured by end users of the Oracle Configurator or other ERP and CRM applications.

### Customer-centric Extensions

See Customer-centric Views.

### Customer-centric Views

Optional extensions to core functionality that supplement product selection with rules for pre-selection, validation, and intelligent views. View capabilities include generative geometry, drawings, sketches and schematics, charts, performance analyses, and ROI calculations.

### Customer Requirements

The needs of the customer that serve as the basis for determining the configuration of products, systems, and/or services. Also called Needs Assessment.

### Data Import

Populating the Oracle Configurator schema with enterprise data from ERP or legacy systems via import tables.

### Data Integration Object

Data Integration Object. A server in the runtime application that creates and manages the interface between the client (usually a user interface like the Active User Interface) and the Oracle Configurator schema.

### Data Maintenance Environment

The environment in which the Oracle runtime configurator data is maintained.

### Data Replication

The activity of downloading and uploading configuration, quote, and order data between the Oracle Configurator schema on the enterprise server and Oracle Configurator Mobile Database on end-user mobile laptop PCs. See also Data Synchronization.

### Datasource

A programmatic reference to a database. Referred to by a datasource name, or DSN.

### Data Synchronization

A process for matching the data in the Oracle Configurator schema and the data available to client processes such as the Oracle SellingPoint application. See also Data Replication.

### Default

The automatic selection of an option based on the pre-selection rules or the selection of another option.

### Defaults

An Oracle Configurator Developer logic rule to determine the logic state of features or options in a default relation to other features and options. For instance, if you set A to True by selecting it, B becomes true (selected) if it is available (not false) and can be set to True without contradicting a non-default rule or a previous default setting for B.

### Defect

A failure in a product to satisfy the users' requirements. Defects are prioritized as critical, major, or minor, and fixes range from corrections or workarounds to enhancements. Also known as a “bug”.

### Defect Tracking

A system of identifying defects for managing additional tests, testing, and approval for release to users.

### Deliverable

A work product that is specified for review and delivery.

### Demonstration

A presentation of the tested application, showing a particular usage scenario.

### Design Chart

An Oracle Configurator Developer rule type for defining advanced Explicit Compatibilities interactively in a chart view.

### Design Review

A technical review that focuses on application or system design.

### Developer

The tool (Oracle Configurator Developer) used to create configuration models. The person who uses Oracle Configurator Developer to create a configurator. See also Implementer

### DIO

See Data Integration Object.

### End User

The ultimate user of the Oracle runtime configurator. The types of end users vary by project but may include salespeople or distributors, administrative office staff, marketing personnel, order entry personnel, product engineers, or customers directly accessing the application via web or kiosk.

### Enterprise

The systems and resources of a business.

### Environment

The arena in which software tools are used, such as operating system, applications, and server processes.

### ERP

Enterprise Resource Planning. A software system and process that provides automation for the customer’s back-room operations, including order processing.

### Excludes

An Oracle Configurator Developer rule type for determining the logic state of features or options in an excluding relation to other features and options. For instance, if you set A to True, B becomes false, since it is not allowed when A is true. If you set A to False, there is no effect on B, meaning it could be true, false, or unknown.

### Extended Functionality

A release after delivery of core functionality that extends that core functionality with customer-centric views, more complex proposal generation, discounting, quoting, and expanded integration with ERP, CRM, and third-party software.

### Feature

An element of the Model structure. A configurable parameter of a component. Features can either have a value (numeric or boolean) or enumerated options.

### Functional Companion

An object associated with a component that supplies methods that can be used to initialize, validate and generate customer-centric views and outputs for the configuration.

### Functional Specification

Document describing the functionality of the application based on user requirements.

### Incremental Construction

The process of organizing the construction of the application into builds, where each build is designed to meet a specified portion of the overall requirements and is unit tested.

### Implementation

The stage in a project between defining the problem by selecting a configuration technology vendor, such as Oracle, and deploying the completed sales configuration application. The Implementation stage includes gathering requirements, defining test cases, designing the application, constructing and testing the application, and delivering it to users.

### Implementer

The person who uses Oracle Configurator Developer to build the model structure, rules, and UI customizations that make up an Oracle runtime configurator.

### Implies

An Oracle Configurator Developer logic rule type that determines the logic state of features or options in an implied relation to other features and options. For instance, if you set A to True by selecting it, B becomes true, since selecting A implies that B is also selected. If you set A to False by deselecting it, there is no effect on B, meaning it could be true false or unknown based on other relations B participates in. And if you set B to True by selecting it, there is no effect on A, meaning it could be true false or unknown based on other relations A participates in. But if you set B to False by deselecting it, the relation of A implies B is preserved only by having A be false (deselected) as well.

### Import Tables

Tables mirroring the Oracle Configurator schema Item Master structure, but without integrity constraints. Import Tables allow batch population of the Oracle Configurator schema Item Master. Import Tables are used in conjunction with extractions from Oracle Applications or legacy data to create, update, or delete records in the Oracle Configurator schema Item Master.

### Install

A program that sets up the local machine and installs the application for testing and use.

### Integration

The process of combining multiple software components and making them work together.

### Integration Testing

Testing the interaction among software programs that have been integrated into an application or system.

### Intelligent Views

Configuration output, such as reports, graphs, schematics, and diagrams, that help to illustrate the value proposition of what is being sold.

### Item Master

A table in the Oracle Configurator schema containing data used to structure the product. Data in the item master is either entered manually or imported from Oracle Applications or legacy data.

### Item Type

A table in the Oracle Configurator schema containing data used to classify the product data in the item master table.

### Log File

A file containing errors, warnings and other information output by the running application.

### Logic Rules

Logic rules directly or indirectly set the logical state (true, false, or unknown) of features and options in the Model.

There are four (4) primary logic rules: Implies, Requires, Excludes, and Negates. Each of these rules takes a list of features or options as operands. See also Logic, Implies, Requires, Excludes, and Negates.

### Maintainability

The characteristic of a product or process to allow straightforward maintenance, alteration, and extension. Maintainability must be built into the product or process from inception.

### Maintenance

The effort of keeping a system running once it has been deployed, through bug fixes, procedure changes, infrastructure adjustments, data replication schedules, etc.

### Maintenance Guide

A guide for maintaining a specific application or system. The maintenance guide covers all aspects of maintenance described in the generic Maintenance Plan.

### Maintenance Plan

A document that outlines what is required for successful maintenance, and who is responsible for all the actions and deliverables of carrying out maintenance on a system.

### MDUI

See Model-driven UI.

### Mobile Database

See Oracle Configurator Mobile Database.

### Model

The entire hierarchical “tree” view of all the data required for configurations, including model structure, variables such as resources and totals, and elements in support of intermediary rules. May consist of BOM Items.

### Model-driven UI

The graphical views of the model structure and rules generated by the Active UI to present end users with interactive product selection based on configuration models.

### Model Structure

Hierarchical, “tree” view of data in terms of product elements (Models, Products Components, Features, Options, BOM Models, BOM OptionClasses, BOM StandardItems, Resources, and Totals). May include reusable components.

### MRP

Manufacturing Resource Planning. A software system and process for monitoring and maintaining the customer's manufacturing systems.

### Negates

An Oracle Configurator Developer logic rule type that determines the logic state of features or options in a negating relation to other features and options. For instance, if you set one item in the relationship to True, the other item must be false. And if you set one item to False, the other item must be true.

### Node

The place in a Model occupied by a component, feature, option or variable, BOM Model, BOM OptionClass, or BOM StandardItem.

### Numeric Rules

Rules that are used to set the global parameters specified in product structuring. See also, Contributes to and Consumes from.

### OC

See Oracle Configurator.

### Opportunity

The workspace in the Oracle SellingPoint application and Oracle Sales Online in which products, systems, and/or services are configured, quotes and proposals are generated, and orders are submitted.

### Option

An element of the Model. A choice for the value of an enumerated feature.

A logical selection made by the end user when configuring a component.

### Oracle Configurator

The product family consisting of development tools and runtime applications such as Oracle Configurator schema, Oracle Configurator Developer, Oracle Configurator window, and Oracle SellingPoint application. Also the Oracle runtime configurator variously packaged for use in networked, mobile, or web deployments.

### Oracle Configurator Architecture

The application runtime architecture consists of the Active User Interface, the Active Model, and the Oracle Configurator schema or Oracle Configurator Mobile Database. The application development architecture consists of Oracle Configurator Developer and the Oracle Configurator schema, with test instances of an Oracle runtime configurator.

### Oracle Configurator Developer

The suite of tools in the Oracle Configurator product family for constructing and maintaining configurators.

### Oracle Configuration Interface Object (CIO)

A server in the runtime application that creates and manages the interface between the client (usually a user interface like the Active User Interface) and the underlying representation of model structure and rules in the Active Model.

CIO protocols support creating and navigating the Model, querying and modifying selection states, and saving and restoring configurations.

### Oracle Configurator Mobile Database

The runtime version of the standard Oracle Configurator schema that manages data for the configuration model in a mobile deployment. The runtime schema includes customer, product, and pricing data as well as data created during operation of an Oracle Configurator.

### Oracle Configurator Schema

The implementation version of the standard Oracle runtime configurator data-warehousing schema that manages data for the configuration model. The implementation schema includes all the data required for the runtime system as well as specific tables used during the construction of the configurator.

### Oracle SellingPoint Application

The test application generated by Oracle Configurator Developer. Also a full configuration environment with opportunity management, quotes, and proposals for networked or mobile deployments.

### Output

The output generated by a configurator, such as quotes, proposals, bills of material (BOM), and customer-centric views.

### PDM

Product Data Management. A software system that manages the version control of product data.

### Populator

An entity in the Oracle Configurator Developer that defines how to create a Model from information in the item master.

### Pre-selection

The default state in a configurator that defines an initial selection of components, features, and options for configuration.

A process that is implemented to select the initial element(s) of the configuration.

### Principal Design Consultant

Member of the project team responsible for architecting the design of the application.

### Product

Whatever is subjected to configuration and is the output of the application.

The root element of the Model.

### Product Family

A collection of products or product lines, which are organized as a group by a provider or manufacturer.

### Project

The workspace in Oracle Configurator Developer in which configurators are constructed

### Project Manager

A member of the project team who is responsible for directing the project during implementation.

### Project Plan

A document that outlines the logistics of successfully implementing the project, including the schedule.

### Property

A named value associated with an object in the Model or the item master. A set of properties may be associated with an item type.

### Property-based Compatibility Rule

A kind of compatibility relationship where the allowable combinations of options are specified implicitly by relationships among property values of the options.

### Prototype

A construction technique in which a preliminary version of the application, or part of the application, is built to facilitate user feedback, to prove feasibility or examine other implementation issues.

### Reference

The use of a reusable component within the Model. Not implemented in Release 11*i* or before.

### Regression Test

An automated test that ensures the newest build still meets previously tested requirements and functionality.

### Requires

An Oracle Configurator Developer logic rule type that determines the logic state of features or options in a requirement relation to other features and options. For instance, if you set one item in the relationship to True, the other item is required to be true as well. And if you set one item to False, the other item must be false as well.

### Resource

Staff or materials available or needed within an enterprise.

A variable in the Model used to maintain the balance of features not consuming more of a specific resource than has been provided by other features.

### Reusable Component

A component that is referenced from multiple locations in the Model. Not implemented in Release 11*i* or before.

### Reusability

The extent to and ease with which parts of a system can be put to use in other systems.

### Rules

Also called business rules or configuration rules. Constraints applied among elements of the product to ensure that defined relationships are preserved during configuration. Elements of the product are components, features, and options. Rules express logic, numeric parameters, implicit compatibility, or explicit compatibility. Rules are used to provide pre-selection and validation capability in an application.

See also Logic Rules and Numeric Rules.

### Runtime

The environment and context in which applications are run or used, rather than developed.

### Sales Configuration

A part of the sales process to which configuration technology has been applied in order to increase sales effectiveness and decrease order errors. Commonly identifies needs assessment and product configuration.

### Server

Centrally located software processes or hardware, shared by clients.

### Solution

The deployed system as a response to a problem or problems.

### System

The hardware and software components and infrastructure integrated to satisfy functional and performance requirements.

### Test Case

A description of inputs, execution instructions, and expected results, which are created for the purpose of determining whether a specific software feature works correctly or a specific requirement has been met.

### Total

A variable in the Model used to accumulate a numeric total, such as total price or total weight.

### Undetermined

The logic state that is neither true nor false, but unknown at the time a logic rule is executed. This logic state is also referred to as Available, especially when considered from the point of view of the Oracle runtime configurator end user.

### Unit Test

Execution of individual routines and modules by the application implementer or by an independent test consultant for the purposes of finding defects.

### Update

Moving a production configurator to a new version of configuration model.

### Upgrade

Moving the configurator to a new release of Oracle Configurator.

### User

The person using the Oracle Configurator Developer tools and methods to build an Oracle runtime configurator. See also end user.

### User Interface

The visible part of the application, including menus, dialog boxes, and other on-screen elements. The part of a system where the user interacts with the software.

### User Requirements

A description of what the Oracle Configurator or Oracle SellingPoint application is expected to do from the end user's perspective.

### User's Guide

Documentation on using the application or configurator to solve the intended problem.

### Validation

Tests that ensure that the configured components will meet specific performance or acceptance criteria.

A type of functional companion that is implemented to ensure that the configured components will meet specific performance or acceptance criteria.

### Variable

Parts of the Model that are represented by Totals, Resources, or numeric Features.

### Verification

Tests that check whether the result agrees with the specification.

# Glossary of Acronyms

**API**

Application Programming Interface

**ATP**

Available to Promise

**BOM**

Bill of Material

**CIO**

Configuration Interface Object

**CM**

Configuration Management

**COM**

Component Object Module

**CRM**

Customer Relationship Management

**DBMS**

Database Management System

**DCOM**

Distributed Component Object Modeling

**DHTML**

Dynamic Hypertext Markup Language

**DIO**

Data Integration Object

**DLL**

Dynamically Linked Library

**DXF**

Drawing Exchange Format (AutoCAD drawings)

**ECO**

Engineering Change Order

**ERM**

Enterprise Relationship Management

**ERP**

Enterprise Resource Planning

**ESD**

Electronic Software Distribution

**ESP**

External Service Provider

**ESS**

Enterprise Selling System

**HT**

High Tech

**HTML**

Hypertext Markup Language

**IP**

Industrial Products

**IS**

Information Services

**ISS**

Interactive Selling System

**ISV**

Independent Software Vendor

**LAN**

Local Area Network

**MAPI**

Messaging Application Programming Interface

**MC/S**

Mobile Client/Server System

**MDUI**

Model-Driven User Interface

**MES**

Marketing Encyclopedia System (Catalog)

**MIS**

Management Information Systems

**MRP**

Manufacturing Resource Planning

**MS**

Microsoft

**OC**

Oracle Configurator

**OCX**

Object Control File, OLE custom controls

**ODBC**

Open Database Connectivity

**OLE**

Object linking and embedding

**OMS**

Opportunity Management System

**OOD**

Object-Oriented Design

**ORB**

Object Request Broker

**PDM**

Product Data Management

**PIA**

Project Impact Assessment

**POS**

Point of Sale

**QA**

Quality Assurance

**RAD**

Rapid Application Development

**RDBMS**

Relational Database Management System

**RFQ**

Request for Quote

**ROI**

Return on Investment

**SAS**

Sales Analysis System

**SCM**

Supply Chain Management

**SCS**

Sales Configuration System

**SE**

Sales Engineer

**SFA**

Sales Force Automation

**SI**

System Integrator

**SOT**

Strategic Options Theory

**SQA**

Software Quality Assurance

**SQL**

Structured Query Language

**TERM**

Technology-Enabled Relationship Management

**TES**

Technology-Enabled Selling

**UI**

User Interface

**VAR**

Value-Added Reseller

**VB**

Microsoft Visual Basic

**WAN**

Wide Area Network

**WIP**

Work In Progress

# Index