Oracle® SellingPoint Configurator

Administration Guide

Release 4.2.2

March 2000

Part No. A74983-03

This document addresses integration with Oracle Applications as well as stand-alone installations of Oracle SellingPoint Configurator. The administration tasks presented here are required to set up and support developers and users of Oracle SellingPoint applications.

Oracle SellingPoint Configurator Administration Guide, Release 4.2.2

Part No. A74983-03

Primary Authors: Denise Boyer, Tina Brand

Contributing Authors: Jan Stetson, Andrew Wolfe

Contributors: Serge Kudravtsev, Sarit Manna, Helen Reznik, Carmela Sciandra, Alex Siaston,

Contents

2 Server Administration

3 Client Administration

4 Data Import

5 Production Deployment

6 Data Export

7 Mobile Deployment Administration

A OSPC SQL*Plus Scripts and Procedures

B Data Import Requirements

C Data Export Requirements

Glossary of Terms

Glossary of Acronyms

Index

List of Figures

List of Tables

Send Us Your Comments

Oracle SellingPoint Configurator Administration Guide, Release 4.2.2

Part No. A74983-03

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 your 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, and telephone number below.

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

Preface

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

Intended Audience

This manual is intended for system administrators and database administrators who are supporting Oracle SellingPoint Configurator users.

Anyone responsible for supporting use of Oracle SellingPoint Configurator (OSPC) should read this book. That includes supporting the application development environment (Oracle SellingPoint Developer) as well as the application run-time environment (Oracle SellingPoint).

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

Structure

This manual contains:

- Chapter 1, "Introduction"
- Chapter 2, "Server Administration"
- Chapter 3, "Client Administration"
- Chapter 4, "Data Import"
- Chapter 5, "Production Deployment"
- Chapter 6, "Data Export"

- Chapter 7, "Mobile Deployment Administration"
- Appendix A, "OSPC SQL*Plus Scripts and Procedures"
- Appendix B, "Data Import Requirements"
- "Glossary of Terms"
- "Glossary of Acronyms"

Related Documents

For more information, see the documentation for your release of Oracle Applications, Release 8 Oracle RDBMS documentation, and the product-specific *Release Notes* for releases supported to work with Oracle SellingPoint 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
. . .	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 names.
[ ]	Brackets enclose optional clauses from which you can choose one or none.
>	The left bracket alone represents the MS DOS prompt.

1 Introduction

Oracle SellingPoint Configurator (OSPC) consists of Oracle SellingPoint Developer, the Oracle SellingPoint application, and the Oracle Configurator Database. Oracle SellingPoint Configurator exists as a standalone Rapid Application Development (RAD) product, and may also be integrated with Oracle Applications for data import from Bills of Material and export to Order Entry. Whether integrated with Oracle Applications or not, Oracle SellingPoint Configurator is commonly set up to use data imported from Oracle or other legacy databases.

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

- Import of Oracle Applications data into OSPC.
- Import of non-Oracle or legacy data into OSPC.
- Export of orders and customers from OSPC to Oracle Order Entry.

In Release 4.2.2, documentation for Oracle SellingPoint Developer is titled *Oracle Configurator Developer User's Guide*. The Oracle SellingPoint application is only available in Oracle SellingPoint Configurator, Release 4.2 and 4.2.1.

1.1 Overview of OSPC Administration

Using Oracle SellingPoint Configurator (OSPC) requires administrative support. This document presents the following categories of administrative tasks:

- Settings in Oracle Applications for integration with OSPC, if applicable
- Preparing data for import into OSPC
- Oracle Configurator Database creation, preparations for use, database (DB) settings, and tuning

- Client/Server connectivity
- Importing Data
- Enabling user access to OSPC
- Data synchronization and replication
- Data export to Oracle Order Entry
- Deployment of an Oracle SellingPoint application

The administrative tasks needed to support using OSPC require both the knowledge of an experienced Database Administrator (DBA) and some knowledge of system administration. However, since DBA and System Administration knowledge is occasionally weak on OSPC projects, this manual tries to compensate with explicit explanations and directions wherever possible.

OSPC provides scripts for many of the necessary administrative tasks. These scripts are available on the OSPC 4.2.2 CD in the DBAdmin folder, and are explained in this book.

The Oracle SellingPoint application also provides an interface (Tools menu) for additional administrative tasks having to do with how and to whom the application is deployed.

If your site is using OSPC integrated with Oracle Applications, administrative tasks could 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.

Integration with Oracle Applications involves importing Bills of Material and customer data from the Oracle Applications database and submitting orders to Order Entry. You can import BOM, Oracle Pricing, customer, address, user, and contact data into the Oracle SellingPoint application. You export quotes from the Oracle SellingPoint application to Oracle Order Entry, and new customer information to Oracle Receivables.

1.2 Terms

Oracle and Oracle SellingPoint 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 SellingPoint Configurator, is installed, set up, and started as an Oracle **Server**.

An Oracle Server consists of one or more Oracle **database instances**. An instance consists of memory and processes that manage a single, self-contained collection of database 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).

Every defined **user** is its own schema. A user can have access to any number of other schemas depending on the user's security domain. The **security domain** of a user consists of the particular combination of **privileges** and resource allocations under which the user employs the database.

Groups of privileges are defined as **roles,** which can be conveniently granted to users instead of granting privileges explicitly.

In the Oracle SellingPoint Configurator generally, and this book in particular, users are distinguished from **end users**. Users are the implementers using Oracle SellingPoint Developer to create Oracle SellingPoint end-user applications. End users are the users of those end user applications.

See Section 1.5, "OSPC Architecture Overview" for additional terms and definitions.

1.3 Requirements

All requirements and prerequisites for installing and using OSPC are presented in the *Oracle SellingPoint Configurator ReadMe* (Part No. A74831-01).

Oracle SellingPoint Configurator users must be registered with (must be defined users in) Oracle8*i* Enterprise Edition. If OSPC is integrated with Oracle Applications, Oracle SellingPoint Configurator end users exporting orders must be defined users in Oracle Applications.

1.4 OSPC Installations

Oracle SellingPoint Configurator is a standalone product and not part of the Oracle Applications product suite. In Oracle Applications 10.7 and 11, the OSPC is not installed using the Oracle Applications Autoinstaller. In Oracle Applications

Release 11*i*, only Oracle SellingPoint Developer, available on the OSPC 4.2.2 CD, is not installed using the Oracle Applications Autoinstaller.

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

- Development
- Test
- Deployment
- Maintenance

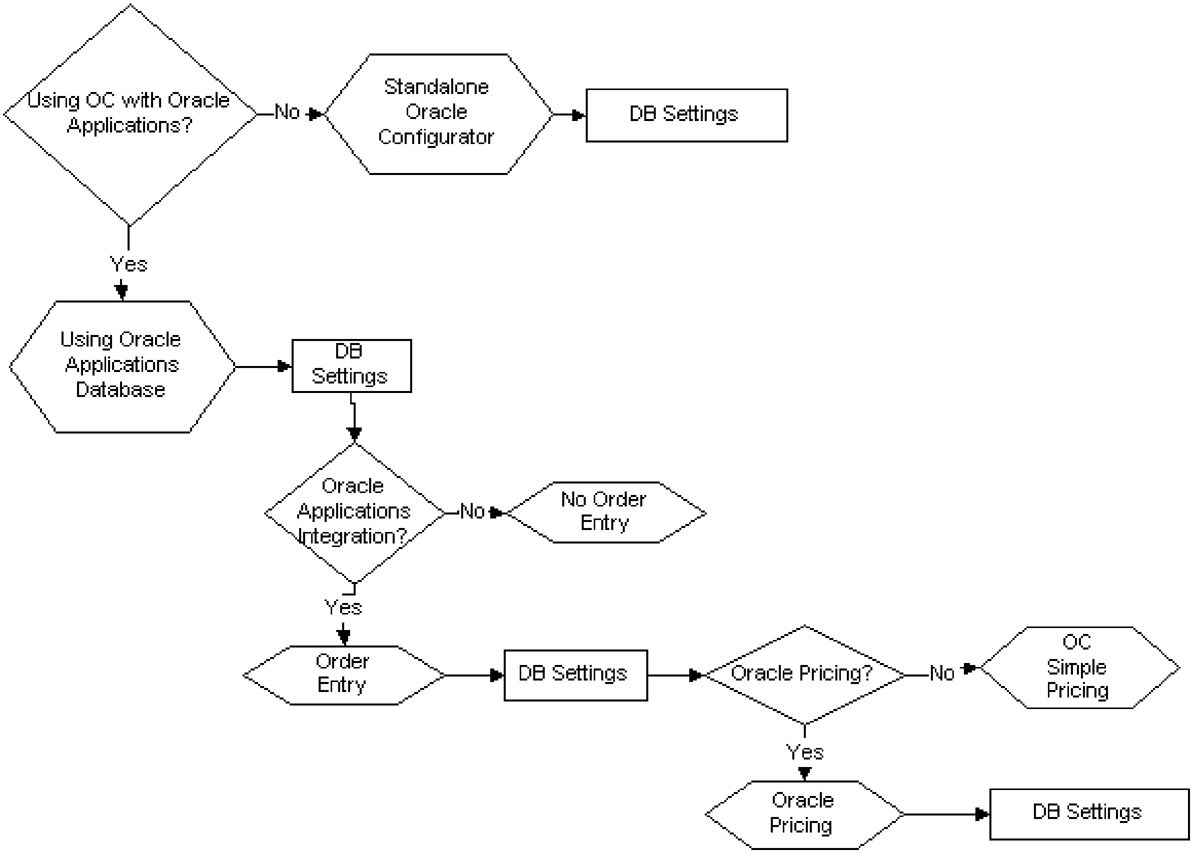
For each installation, there is likely to be a separate server database, or at least a separate schema (DB owner) within a database shared by more than one installation of the project. The schema of objects that the DBA creates for any installation of OSPC is called the Oracle Configurator Database.

In any installation, you will be running the Oracle SellingPoint Configurator in one of the following scenarios:

- without any other Oracle Applications, or
- with Oracle Applications (data import from BOM and Receivables) but not integrated with Order Entry, or
- integrated with Oracle Applications Order Entry, but not using Oracle Pricing, or
- integrated with Oracle Applications Order Entry, and using Oracle Pricing.

In Figure 1–1, below, a map of decisions shows the possible installation scenarios. Note that you establish parameters for these scenarios through settings in the CZ_DB_SETTINGS table in the Oracle Configurator Database.

Figure 1–1 Installation Scenarios for Oracle SellingPoint Configurator


Using OC with Oracle Applications?
No
Standalone Oracle Configurator
DB Settings
Yes
Using Oracle Applications Database
DB Settings
Oracle Applications Integration?
No
No Order Entry
Yes
Order Entry
DB Settings
Oracle Pricing?
No
OC Simple Pricing
Yes
Oracle Pricing
DB Settings

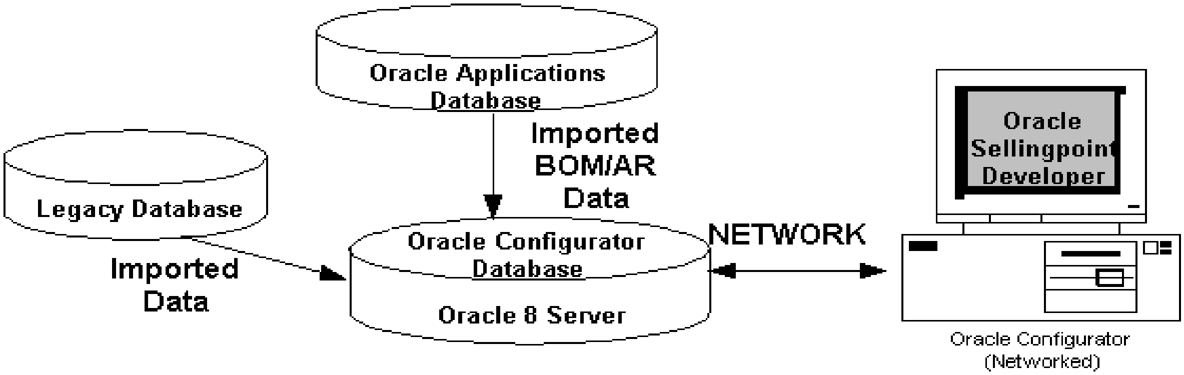
DB settings additionally vary depending on whether the installation is for development, test, deployment, or maintenance. There is a CZ_DB_SETTINGS table for each Oracle Configurator Database schema or owner that is in effect for all users of that schema.

For details about what parameters in CZ_DB_SETTINGS apply in each of the scenarios, above, see Section 2.4, "Set DB Settings".

1.4.1 Development

A development environment is one in which you implement your Oracle SellingPoint Configurator project by constructing an end-user application referred to in this manual as the Oracle SellingPoint application. Development is most commonly a networked installation with Oracle SellingPoint Developer on the client machine and the Oracle Configurator Database on a server.

Figure 1–2 Development Installation


Oracle Applications Database
Imported BOM/AR Data
Legacy Database
Imported Data
Oracle Configurator Database
Oracle 8 Server
NETWORK
Oracle Sellingpoint Developer
Oracle Configurator (Networked)

The source database for data import, whether Oracle Applications or another database, may or may not be on the same machine as the Oracle Configurator Database. If not, the integration tables in the Oracle Configurator Database must be set up with links to the source database. The Oracle SellingPoint Configurator provides scripts for creating the Oracle Configurator Database with links to the Oracle Applications database on another machine.

Note: Oracle recommends installing both the Oracle Configurator Database and Oracle Applications database in one RDBMS hosted in one instance requiring no database links.

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 importing new data. Modeling data includes item master, item type, structure, and property data.

Before deployment, refresh the Oracle Configurator Database with changes to the legacy database or Oracle Applications database, including account and end user (sales representatives) data. In some cases this may require further development on the configuration model to bring it in line with changes in your business.

1.4.2 Test

A test environment is one in which you test your Oracle SellingPoint application 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. Depending on your deployment requirements, also test your Oracle SellingPoint application in a disconnected mode, making sure periodic import and replication work as required.

1.4.3 Deployment

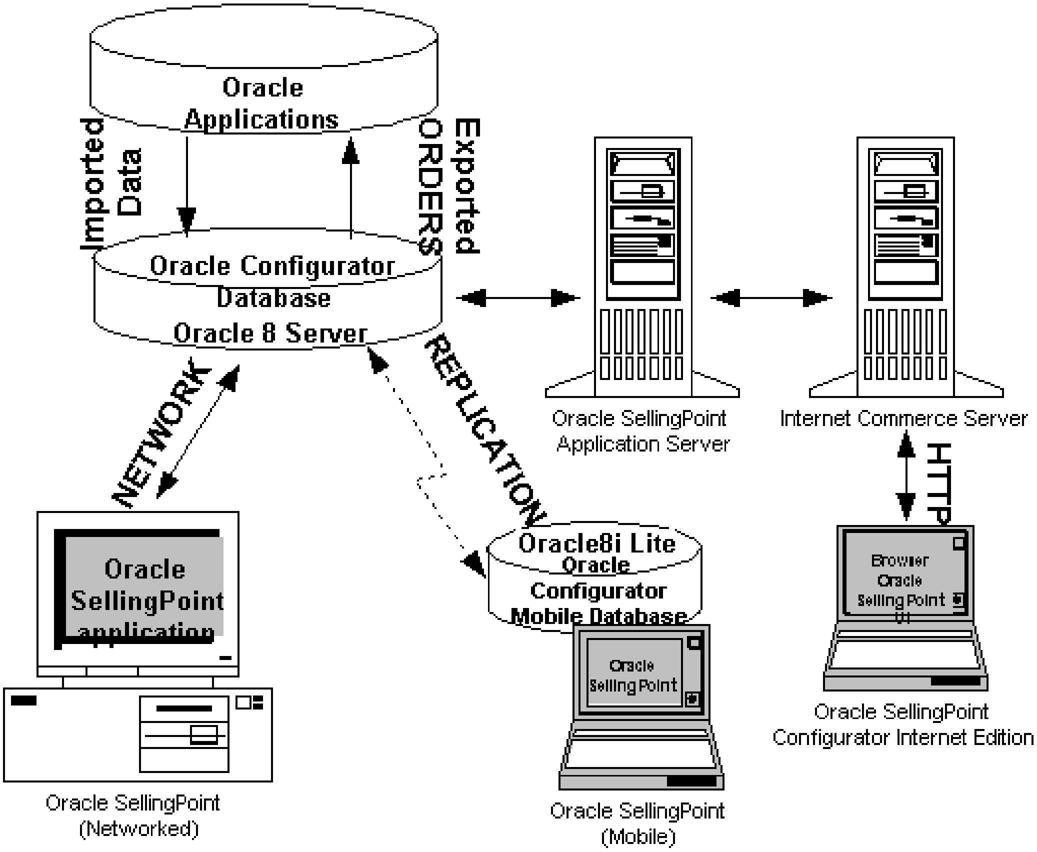
A deployed application is one in which end users of the Oracle SellingPoint application use the software in a production mode. To prepare for deployment, you should publish a production-ready version of the Oracle Configurator Database. If you have mobile users, deployment includes creating replicas of the Oracle Configurator Database (the Oracle Configurator Mobile Database) and setting up the laptop machines appropriately.

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

There are three possible deployment scenarios:

- networked
- mobile (requires data replication to synchronize enterprise data between the server and the laptop)
- web (Oracle SellingPoint Internet Edition)

Figure 1–3 Overview of Deployments Integrated with Oracle Applications


Oracle Applications
Imported Data
Exported ORDERS
Oracle Configurator Database
Oracle 8 Server
NETWORK
REPLICATION
Oracle SellingPoint application
Oracle SellingPoint (Networked)
Oracle8i Lite
Oracle Configurator Mobile Database
Oracle SellingPoint
Oracle SellingPoint (Mobile)
Oracle SellingPoint Application Server
Internet Commerce Server
HTTP
Browser Oracle Selling Point UI
Oracle SellingPoint Configurator Internet Edition

In Figure 1–3, if Oracle SellingPoint application were not integrated with Oracle Applications, there would be no order export to Order Entry. Imported data could come from any legacy database, updating the Oracle Configurator Database with fresh enterprise data.

The mobile deployment requires that the application and the Oracle Configurator Mobile Database are installed on a laptop in the field, synchronizing data with the Oracle Configurator Database on a server. Replication ensures that imported data refreshes the Oracle Configurator Mobile Database and quotes are submitted to Order Entry for processing.

Orders can also be submitted in a networked environment for non-field sales.

With an Internet Edition deployment, the Oracle SellingPoint application user interface is running in a browser. The application itself, and the Oracle Configurator Database, are running on the Oracle SellingPoint application server machine with the internet commerce server brokering the processes and http connection.

1.4.4 Maintenance

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

At the time you deploy your Oracle SellingPoint application and publish a production Oracle Configurator Database, you also create a maintenance Oracle SellingPoint application. In the course of a deployed release of your Oracle SellingPoint application, you may conduct periodic imports from your legacy data or Oracle Applications and redeploy the refreshed Oracle SellingPoint applications.

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 Database with the Oracle Configurator Database under development for the next release of your Oracle SellingPoint application.

When you upgrade the version of Oracle SellingPoint Configurator that your Oracle SellingPoint application runs against, you start by upgrading your Oracle Configurator Database. OSPC provides upgrade scripts for server databases. Oracle8*i* Lite databases are recreated by generating a replica of an upgraded server database.

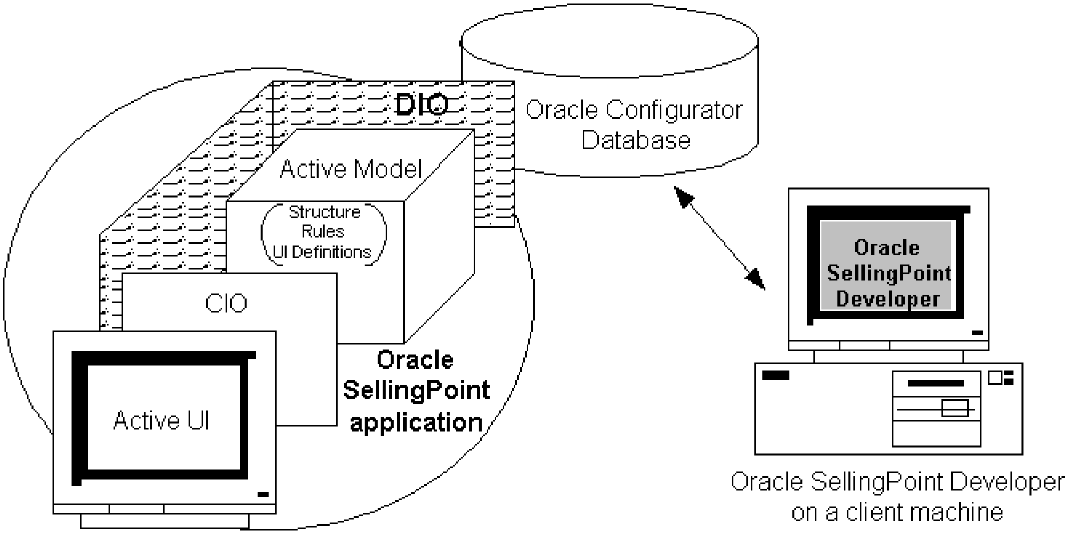
Before applying the structural changes that are inherent in upgrading your Oracle Configurator Database to run with a new release of OSPC, make a backup copy of the database. Once you have upgraded your Oracle Configurator Database for a new release, you must re-execute the Generate Active Model command in Oracle SellingPoint Developer.

1.5 OSPC Architecture Overview

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

The model-driven UI definitions in the Oracle SellingPoint application are called the **Active UI**. The Active UI and Active Model interact with the data in the Oracle Configurator Database to present the users of Oracle SellingPoint Configurator with the data they need to develop and run the Oracle SellingPoint application.

Figure 1–4 Overview of OSPC Architecture


DIO
Oracle Configurator Database
Active Model
Structure
Rules
UI Definitions
CIO
Active UI
Oracle SellingPoint application
Oracle SellingPoint Developer
Oracle SellingPoint 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.

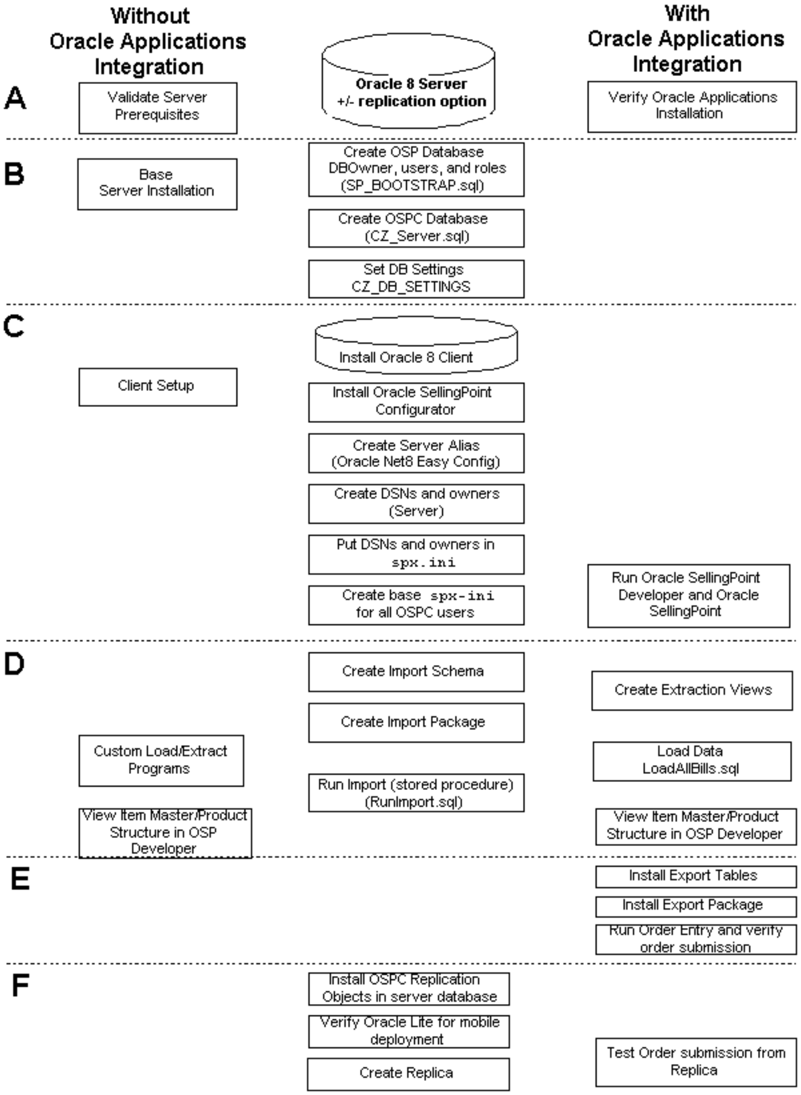
The DIO is the Data Integration Object, responsible for communications between the Active UI and the Oracle Configurator Database, and between the Active Model and the Oracle Configurator Database.

1.6 Flow of Administration Tasks

In order to support any installation of Oracle SellingPoint Configurator, you must perform administrative tasks in the order presented below. The tasks in the center column are required for supporting Oracle SellingPoint Configurator.

Many tasks presented in Figure 1–5 can be combined in running various scripts provided by the Oracle SellingPoint Configurator, but setting up an OSPC requires the activities in this diagram.

Figure 1–5 Administrative Task Flow



1.7 Frequently 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 SellingPoint application. Instructions for some frequently performed tasks are included in the following sections:

- Connect to a Database Instance
- Run SQL*Plus in the <OSPC-scripts> Directory

This book does *not* present instructions for

- establishing any privileges to any user, including DBA privileges
- establishing a database instance such as <spxsid> or <appssid>
- installing and setting up Oracle8*i* Enterprise Edition RDBMS or

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

1.7.1 Connect to a Database Instance

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

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 Database by connecting to the instance <spxsid> as a the user or schema you need to access.

   Example:

   ```
   SQL> connect <spx>/<spxpass>@<spxsid>
   ```

   where `<spx>` is the owner (DBOwner) of the Oracle Configurator Database, and `<spxsid>` is the name for the Oracle8*i* Enterprise Edition instance on which the Oracle Configurator Database is installed.

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

   Example:

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

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

Example:

```
SQL> conn <imp>/<imppass> @<spxsid>
```

1.7.2 Run SQL*Plus in the <OSPC-scripts> Directory

OSPC provides scripts for many of the necessary administrative tasks. These scripts are available on the OSPC 4.2.2 CD in the DBAdmin folder. Any OSPC 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 OSPC SQL*Plus scripts from SQL Worksheet.

Some scripts require be run while connected to a specific database instance (see Section 1.7.1, "Connect to a Database Instance").

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

4. Start SQL*Plus
5. Go to File --> Open
6. Navigate to your <OSPC-scripts> directory
7. Click Cancel

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

2

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 SellingPoint Configurator involves both.

2.1 Overview of Server Administration

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

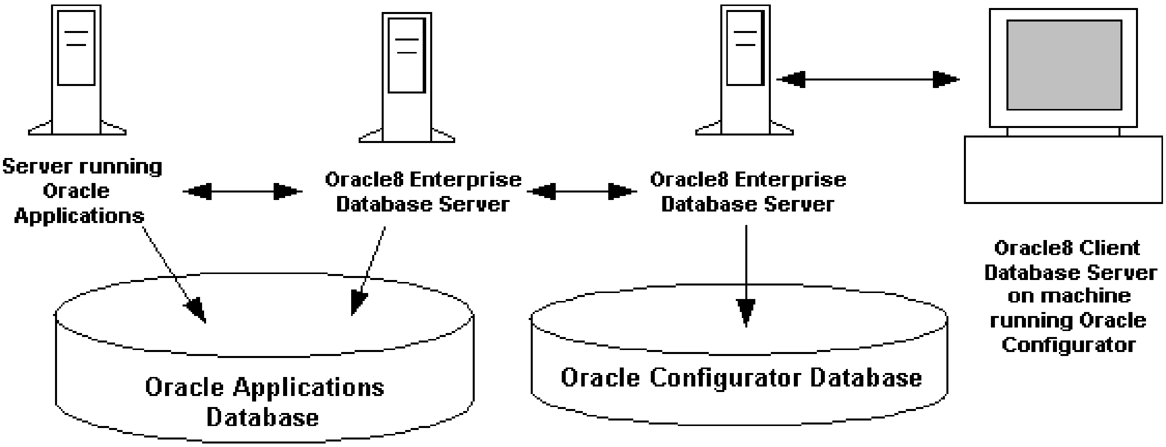
- Server (machine) running Oracle SellingPoint Configurator
- Server (machine) running Oracle Applications if OSPC is integrated with Oracle Applications
- Oracle Applications Server for web deployments
- Database Server where the Oracle Configurator Database is installed
- Database Server where the Oracle Applications database is installed

This chapter is primarily about creating the server database for running Oracle SellingPoint Configurator with or without integration to Oracle Applications.

If OSPC is integrated with Oracle Applications in any way, that server machine running the Oracle Configurator Database must also have access to the Oracle Applications database and Oracle Applications running either on the same machine or other machines.

In addition, Oracle SellingPoint Internet Edition requires Oracle Applications Server to broker the processes and http connection (see Figure 1–3, "Overview of Deployments Integrated with Oracle Applications").

Figure 2–1 System Configuration for OSPC Server


Server running Oracle Applications
Oracle8 Enterprise Database Server
Oracle8 Enterprise Database Server
Oracle8 Client Database Server on machine running Oracle Configurator
Oracle Applications Database
Oracle Configurator Database

Although Oracle Applications, the Oracle Applications database, and the Oracle Configurator Database can all be located on separate machines, Oracle recommends installing the Oracle Applications database and Oracle Configurator Database on the same machine in the same Oracle8*i* Enterprise Edition RDBMS instance so as to avoid problems caused by managing data links.

2.2 Oracle SellingPoint Configurator Server Machine

In general, the Oracle Configurator Database is installed on a machine that acts as the server for networked clients running OSPC. This would be the case for any installment of OSPC including mobile users creating or synchronizing a replica with the server. The server machine must have Oracle8*i* Enterprise Edition installed and the client machine(s) must have Oracle Client 8.0 installed. See Chapter 3, "Client Administration".

Whether or not your OSPC is integrated with Oracle Applications, you need to install Oracle8*i* Enterprise Edition on the machine where the Oracle Configurator Database will be installed.

Your tasks in getting the server database setup include the following:

- Validate Server Prerequisites
- Install Base Server Database (<spxsid>)

- Tune Base Server Database
- Install Oracle Applications database (<appssid> and <apps>), if applicable
- Verify Oracle Applications installation, if applicable
- Install and verify Oracle Applications Server access, if applicable
- Establish OSPC Server Instance (<spxsid>)
- Create Oracle Configurator Database schema (<spx>)
- Set DB Settings
- Create OSPC Oracle8*i* Lite Database on mobile client machines, if applicable

Once a server database is set up, additional tasks may include making copies of populated schemas and upgrading the Oracle Configurator Database to a new version of the schema in a new release of OSPC.

2.2.1 Validate Server Prerequisites

Even if your OSPC is not integrated with Oracle Applications, it is recommended that you install Oracle8*i* Enterprise Edition. If you are integrating with Oracle Applications, you must install Oracle8*i* Enterprise Edition first. If you intend to support replication, 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 Verify Oracle Applications Installation

The server machine where the Oracle Configurator Database is installed must have access (data connectivity) to the Oracle Applications database which may be on another server machine. The server machine where Oracle Applications are running must also have access to the Oracle Applications database. Oracle recommends that the Oracle Configurator Database and the Oracle Applications database be hosted in the same server instance.

If your OSPC is integrated with Oracle Applications, you should verify your Oracle Applications installation. OSPC integrates with Oracle Applications BOM, Pricing, and Order Entry. You verify your Oracle Applications installation by creating or altering a BOM, and submitting orders against that BOM, using Oracle Applications.

All Oracle SellingPoint application end users must be defined as application users for Oracle Applications. The CZ_END_USERS table in the Oracle Configurator Database is populated with Oracle Applications users as database users at the time of data import, provided the DB_SETTING AUTOCREATE_IMPORTED_USERS parameter is not Null and the CZ_XFR_TABLE is configured to allow import of end users.

2.2.3 Base Database Server Installation

A base installation of Oracle8*i* Enterprise Edition is sufficient for supporting Oracle SellingPoint Configurator.

Once Oracle8*i* Enterprise Edition is installed, you must create an instance (<spxsid>) within which the Oracle Configurator Database (<spx>) will exist.

Oracle recommends that you create your Oracle Configurator Database in the same RDBMS as your Oracle Applications database. Oracle also recommends that the Oracle Applications database instance (<appssid>) has the same instance ID as your Oracle Configurator Database (<spx>). This recommended arrangement means that <appssid> = <spxsid>.

2.3 Establish an Oracle Configurator Database

The DBA creates the Oracle Configurator Database on a server machine where Oracle8*i* Enterprise Edition is installed. If OSPC integrates with Oracle Applications, Oracle recommends creating the Oracle Configurator Database within the same Oracle8*i* Enterprise Edition RDBMS where the Oracle Applications database already exists. If the Oracle Configurator Database and Oracle Applications database are hosted in the same server instance, no data links are required for integrating the two databases.

The tasks for creating an Oracle Configurator Database are presented in the following sections. In general terms, the tasks are:

- Install Replication support for the Oracle8*i* Enterprise Edition RDBMS, if replication is required.
- Establish or identify the server instance (<spxsid>) needed to host the Oracle Configurator Database. This requires DBA privileges.
- Connect to <spxsid> with DBA privileges and create a DBOwner (<spx>) for the Oracle Configurator Database. This step is handled by the script SP_BOOTSTRAP.sql, which also creates <spxdev> and assigns the tablespace used by these schemas.

See Section 2.3.1, "Create Oracle Configurator DBOwner" on page 2-5.

- If the Oracle Configurator Database needs to support data import of any kind, create a DBOwner for the integration subschema, which includes the import tables (<imp>). This step is handled by the script SP_BOOTSTRAP.sql, which also assigns the tablespace used by these schemas.
- Grant connect and resource privileges to the DBOwner of the Oracle Configurator Database. This step is handled by the script SP_BOOTSTRAP.sql, which assigns the <SPX_USER> role to <spx>.

  See Section 2.3.2, "Create OSPC Users and Responsibilities" on page 2-7.
- Create the Oracle Configurator Database (populate the schema with tables and other objects). This step is handled by several available scripts. CZ_SERVER.sql creates the Oracle Configurator Database schema without integration objects. InstAppsIntegrate.sql and InstAppsIntegrateViaLink.sql create the integration objects.

  See Section 2.3.3, "Create the Oracle Configurator Database" on page 2-10.

For an overview of the client/server data communication architecture, see Figure 3–1, "Client/Server Data Communication Architecture" on page 3-5. For detailed information about the OSPC SQL*Plus scripts, see Appendix A, "OSPC SQL*Plus Scripts and Procedures".

2.3.1 Create Oracle Configurator DBOwner

You will have to specify an owner for the Oracle Configurator Database, as well as run scripts to define roles and grant permissions for users by role. These tasks are presented in the next sections and should be executed by running SP_BOOTSTRAP.sql (see Section 2.3.1.1, "Instructions for Running SP_BOOTSTRAP" and Section A.2.2, "Using SP_BOOTSTRAP.sql").

DBOwner <spx> is the owner of the database schema <spx>, referred to as the Oracle Configurator Database. This schema is created by someone with DBA privileges (<DBAUser>) connected to the instance <spxsid>, specifically for development, test, deployment or maintenance of OSPC. The value of <spx> is the DBOwner, which is equivalent to the schema that the DBOwner owns.

Establish the DBOwner for the implementation environment of your OSPC project. Create a separate user (<spxdev>) representing the implementation group (Developer users). Implementers using Oracle SellingPoint Developer then log on with that username, using the password you defined (<spxdev>/<spxdevpass>).

The DBOwner <spx> must be granted RESOURCE privileges. Resource is a predefined role in all Oracle databases (see Section 2.3.2, "Create OSPC Users and Responsibilities" on page 2-7).

When the schema is populated with Oracle Configurator Database objects using one of the OSPC SQL*Plus creation scripts, <spx> is entered in the CZ_END_USERS table. The user <spxdev> is not entered in the CZ_END_USERS table. The passwords (<spxpass> and <spxdevpass>) are not stored anywhere in the Oracle Configurator Database, but rather in the instance (<spxsid>).

Other users can be defined in the <spxsid> instance with access privileges to relevant parts of OSPC, such as both Oracle SellingPoint Developer and the Oracle SellingPoint application (true for <spx>), just Oracle SellingPoint Developer (true for <spxdev>), or just the Oracle SellingPoint application (usernames listed in the CZ_END_USERS table).

2.3.1.1 Instructions for Running SP_BOOTSTRAP

OSPC provides the SP_BOOTSTRAP.sql script for creating a DBOwner for the Oracle Configurator Database schema, an owner for development with Oracle SellingPoint Developer, and an owner for the integration tables schema of the Oracle Configurator Database. To run SP_BOOTSTRAP.sql, you must have DBA privileges. For details about the script, see Appendix A.2.2, "Using SP_BOOTSTRAP.sql" on page A-22.

1. On the machine where Oracle8*i* Enterprise Edition is installed, establish a database instance (<spxsid>) for your OSPC project.
2. You are running SQL*Plus in the <OSPC-scripts> directory. See Section 1.7.2, "Run SQL*Plus in the <OSPC-scripts> Directory" on page 1-14 for instructions on how to do this.
3. Run the script SP_BOOTSTRAP.sql.

   SP_BOOTSTRAP.sql connects to the database instance <spxsid> with DBA privileges.

   Example (entered all on one line, one space between each argument):

   ```
   SQL> @SP_BOOTSTRAP <spx> <spxpass> <spxdev> <spxdevpass> <defaultspace>
   <tempspace> <imp> <imppass> <impdefaultspace> <imptempspace>  <indxspace>
   <SPX_USER> <spxsid>
   ```

   The DBA gives <spx>, <spxpass>, <spxdev>, <spxdevpass>, <imp>, and <imppass> whatever values are desired for the username and password of the

Oracle Configurator Database, Oracle SellingPoint Developer, and the integration schema owner within the Oracle Configurator Database.

The value of <defaultspace> and <impdefaultspace> are the tablespaces specified for the user data in the Oracle Configurator Database, Oracle SellingPoint Developer, and the integration schema within the Oracle Configurator Database. The <tempspace> and <imptempspace> are the tablespaces specified for temporary data in the Oracle Configurator Database, Oracle SellingPoint Developer, and the integration schema within the Oracle Configurator Database. When Oracle8*i* Enterprise Edition is installed, the default values for the <defaultspace> and <tempspace> tablespaces are typically USERS and TEMP, respectively. See your Oracle Applications Administrator for the exact names of these tablespaces on your server.

The value of <indxspace> is the tablespace specified 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. SP_BOOTSTRAP.sql inserts this value in the SpxIndxTablespace setting in the CZ_DB_SETTINGS table.

<SPX_USER> indicates the value of DB_USER_ROLES in the CZ_DB_SETTINGS table. You can specify any role name for <SPX_USER>, which SP_BOOTSTRAP.sql inserts into the CZ_DB_SETTINGS table. Other scripts that expand an existing CZ_DB_SETTINGS table, e.g., CZ_SERVER.sql, preserve any value inserted by SP_BOOTSRAP.sql.

SP_BOOTSTRAP.sql creates the <SPX_USER> role in the Oracle Configurator Database (see Section 2.3.2, "Create OSPC Users and Responsibilities").

4. Make a note of the values of <spx>, <spxpass>, <spxdev>, <spxdevpass>, <imp>, <imppass>, and <spxsid> for the QUICK START IMPORT (see Section 4.2.3, "Setup for Direct Import") or a selective import (see Section 4.2.4, "Setup for Generic Import" on page 4-18).

If you don't use SP_BOOTSTRAP.sql, you should create both database user and role, then database objects, and then run GRANT_TO_ROLE.sql.

2.3.2 Create OSPC Users and Responsibilities

The DBOwner <spx> must be assigned the RESOURCE role to create users. To allow creation of users during import and in the Oracle SellingPoint application, the DBOwner of the Oracle Configurator Database schema must be granted DBA or preferably a more granular role such as 'create user'. When integrating 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 Database schema involved in export to Order Entry.

When you create an Oracle Configurator Database (populate the <spx>_schema with tables and objects), the following user tables are created:

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 Database DBOwner (<spx>) is automatically inserted in the CZ_END_USERS table and granted the role <SPX_USER>. <SPX_USER> is the only role created by the OSPC SQL*Plus scripts (SP_BOOTSTRAP.sql). The <SPX_USER> role is stored in the CZ_DB_SETTINGS table, it is needed for using OSPC, and it is supported by Oracle. The <SPX_USER> role is not sufficient for adding or managing end users.

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 Database tables:

- Select
- Insert
- Update
- Delete

In terms of using the Oracle SellingPoint application, any user that is database enabled and granted <SPX_USER> privileges can create customers, i.e., Opportunities, Configurations, etc.

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

The Oracle Configurator Database DBOwner (<spx>) can log in to both the Oracle SellingPoint Developer and the Oracle SellingPoint application. Although the OSPC SQL*Plus scripts automatically insert <spx> in the CZ_END_USERS table, a non-DBOwner user of Oracle SellingPoint Developer need only be an Oracle8*i* Enterprise Edition account with <SPX_USER> role privileges to log into Developer

(<spxdev>). To use the Oracle SellingPoint application, that non-DBOwner user (Oracle8*i* Enterprise Edition account) would also have to be inserted in the CZ_END_USERS table. SP_BOOTSTRAP.sql does not insert <spxdev> into the CZ_END_USERS table.

The Oracle Configurator Database DBOwner (<spx>) also has privileges that enable the 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.

Any end user assigned the <SPX_USER> role and listed in the CZ_END_USERS table can log in to the Oracle SellingPoint application. End users cannot, however, view each other's Opportunities. Users can only view their own Opportunities. End users are associated with an ORGANIZATION_ID for accessing Order Entry. See Section 5.3, "End User Administration" on page 5-4 for more information.

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 Database 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 SellingPoint Developer and the Oracle SellingPoint application.

Any OSPC user must also be defined as a database user in the Oracle8*i* Enterprise Edition server database.

If your OSPC is integrated with Oracle Applications, any OSPC 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.

GRANT_SELECT_FOR.sql grants access on Oracle Applications tables to the Oracle Configurator Database DBOwner. See also Appendix A.2.7, "Using GRANT_SELECT_FOR.sql".

During a direct import from Oracle Applications, the CZ_END_USERS table in the Oracle Configurator Database 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 2.4.0.6, "DB Settings for ORAAPPS_INTEGRATE" on page 2-23 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.

The `spx.ini` file assumes the following default users / passwords for both the Oracle8*i* Enterprise Edition server and Oracle8*i* Lite versions of the Oracle Configurator Database.

Table 2–1 Default spx.ini User Definitions

DBMS	Instance	Role	DBOwner	Username	Password
Oracle8*i* Enterprise Edition	Spx_oracle	SPX_USER	spx	spx14b	spx14b
Oracle8*i* Lite†	Spx_Lite	SPX_USER	spx	spx	spx

†The Oracle8*i* Lite database is only used with Oracle SellingPoint Developer for prototyping, demonstrations, and working through the *Oracle Configurator Developer Tutorial*.

2.3.3 Create the Oracle Configurator Database

Users cannot run Oracle SellingPoint Configurator without having access to an Oracle Configurator Database and having an ODBC Data Source Name (DSN) defined to point to that database. Any client machine running OSPC must have data connectivity service configured to the Oracle8*i* Enterprise Edition Oracle Configurator Database on the server machine (see Section 3.5, "Set Up Oracle SellingPoint Configurator" on page 3-4).

Oracle SellingPoint Configurator provides numerous SQL*Plus scripts for creating the Oracle Configurator Database you need. These OSPC scripts are all available in the DBAdmin folder on the Oracle SellingPoint Configurator CD-ROM. Be aware that they are not guaranteed to work for all environments or situations, but are intended as a starting point or guide. Oracle advises that you evaluate the appropriateness of each script before running it at your site. Copy this folder to your Oracle SellingPoint Configurator install directory for optimal script usage.

For a list of available scripts and information on how to run these scripts successfully and the arguments available for each, see Appendix A.2, "Scripts".

To determine which scripts to use, you need to know the requirements of your site and installation. For instance, if your OSPC is integrated with Oracle Applications, your Oracle Configurator Database must have the necessary integration tables and packages needed to transfer data. The script(s) you use to create import tables and the necessary import packages depends on the specifics of your site.

Several scripts are available for the quickest possible access to an Oracle Configurator Database.

> **Note:** The OSPC SQL*Plus scripts for creating an Oracle Configurator Database schema are supported with Oracle8*i* Enterprise Edition. When creating Oracle Configurator Database in Oracle 8.0.x, the CZ_CF_API package compiles with errors and you must replace it by loading CZ_CF_API_80.sql after creating the Oracle Configurator Database. CZ_CF_API package is not available at all with Oracle 7.3.

SP_BOOTSTRAP.sql creates a DBOwner for the Oracle Configurator Database schema (see Section 2.3.1, "Create Oracle Configurator DBOwner" on page 2-5).

GO_IMPORT.sql creates an Oracle Configurator Database with all the integration tables, and executes a complete direct data import from Oracle Applications (also see Section 4.2.3, "Setup for Direct Import").

2.3.3.1 Create the Oracle Configurator Database without Integration

Oracle SellingPoint Configurator provides a script for creating an empty base Oracle Configurator Database, CZ_SERVER.sql, that does not include integration tables.

CZ_SERVER.sql creates the primary database objects that constitute an Oracle8*i* Enterprise Edition Oracle Configurator Database. An Oracle Configurator Database created with CZ_SERVER.sql does not include import and export tables.

GO_IMPORT_ONLY.sql also creates the primary database objects that constitute an Oracle Configurator Database. However, in addition, this script creates the import tables and import packages needed for non-Oracle Applications data import.

2.3.3.2 Create the Oracle Configurator Database for Integration

Oracle SellingPoint Configurator provides scripts for creating an empty Oracle Configurator Database that includes integration tables. Integration tables are the import tables and Export Tables in the Oracle Configurator Database. The integration tables ensure that data populating the Oracle Configurator Database is in a format that allows import from and export to Oracle Applications to be successful.

To import and export data to and from the Oracle Configurator Database, the database must contain the Integration Tables.

Run **CREATE_IMPORT_SCHEMA.sql** to add import tables only to an already existing Oracle Configurator Database (created with CZ_SERVER.sql).

Run **InstAppsIntegrate.sql** to add all necessary integration tables to an already existing Oracle Configurator Database (created with CZ_SERVER.sql) that is hosted in the same instance that hosts the Oracle Applications database (<appssid>=<spxsid>).

> **Note:** Oracle recommends that you start with an empty Oracle Configurator Database when adding any integration subschema to the Oracle Configurator Database.

Run **InstAppsIntegrateViaLink.sql** to add all necessary integration tables to an already existing Oracle Configurator Database (created with CZ_SERVER.sql) that is NOT hosted in the same instance that hosts the Oracle Applications database (<appssid>≠<spxsid>). Be sure to observe any restrictions that apply to running InstAppsIntegrateViaLink.sql (see Appendix A.2.4, "Using InstAppsIntegrateViaLink.sql" on page A-27).

These, and the scripts and procedures they call, are described in detail in Appendix A, "OSPC SQL*Plus Scripts and Procedures".

See also Section 3.6, "Create Oracle8i Lite Versions of an Oracle Configurator Database" on page 3-18.

2.3.4 Characteristics of the Oracle Configurator Database

A key feature of the Oracle Configurator Database is the ability to run multiple schemas independently in the same server instance, using standard DB users and permissions. It is possible to run development, testing, pilot, and production databases all on the same server. The same user can use any of these schemas to run Oracle SellingPoint Configurator by identifying them in the `spx.ini` file.

The Oracle Configurator Database does not use public synonyms. The order export software uses non-public synonyms to refer to tables in the Oracle Applications database. See Appendix A.2.4, "Using InstAppsIntegrateViaLink.sql" for the implication of synonyms on GLOBAL_NAMES.

The Oracle Configurator Database consists of 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.

2.3.4.1 IM Item-Master

CZ_ITEM_MASTERS
CZ_ITEM_TYPES
CZ_PROPERTIES
CZ_ITEM_TYPE_PROPS
CZ_ITEM_PROPERTY_VALUES
CZ_REL_TYPES
CZ_ITEM_PARENTS

2.3.4.2 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_POPULATORS
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

2.3.4.3 UI User Interface (Active UI)

CZ_UI_DEFS
CZ_UI_NODES
CZ_UI_PROPERTIES
CZ_UI_NODE_PROPS

2.3.4.4 LC Logic for Configuration (Active Model)

CZ_LCE_HEADERS
CZ_LCE_LINES
CZ_LCE_OPERANDS

2.3.4.5 PR Pricing

CZ_PRICE_GROUPS
CZ_PRICES

2.3.4.6 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

2.3.4.7 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

2.3.4.8 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

2.3.4.9 GN General Use Tables

CZ_DB_LOGS
CZ_DB_SETTINGS

2.3.5 Tune Performance of the Oracle Configurator Database

For better performance, Oracle recommends considering the following tuning tasks:

- setting initialization parameters
- reducing chained and migrated rows
- I/O performance

2.3.5.1 Initialization Parameters

SHARED_POOL_SIZE is an initialization parameter. The default Oracle Configurator Database shared pool size is 60Mb. A shared pool size set to 90 Mb improves performance.

2.3.5.2 Reduce Chained and Migrated Rows

Certain Oracle Configurator Database tables tend to accumulate in size. For these tables, it may be desirable to revise the CZ_SERVER.sql script and provide INITIAL_SIZE values.

Records marked for deletion can proliferate during development. Purging deleted records can improve use of disk space. See Section 2.6.1, "Purge" on page 2-36 for more information about PURGE.

Another approach to keeping an Oracle Configurator Database performing quickly is with indexing and adjustments as performance problems arise.

2.4 Set DB Settings

When you create an Oracle Configurator Database with one of the available scripts, you establish installation-wide customizable settings that describe the structure and the content of the Oracle Configurator Database, 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 Database.

You must enter the values your installation requires in the DB settings 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 2–2, "CZ_DB_SETTINGS" briefly describes the DB settings. The DB settings for each section are described in more detail in the sections following the table.

Table 2–2 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 imported with GO_IMPORT.sql
1	DB_USER_ ROLES	<any_string>	SPX_ DEVELOPER	Role assigned to Oracle SellingPoint 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 Applications Order Entry owner <oe>
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/ Named only	NO	Indicates whether or not to enable imported users as database users or to enable only the end users with valid login names as database users
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 2.4.0.7, "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 2.4.0.7, "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*
CustomerExportEnabled	ORAPPS_ INTEGRATE	YES/NO	YES	Indicates whether Oracle Configurator Database customer information can be exported to Oracle Applications

Table 2–2 CZ_DB_SETTINGS

Setting_ID	Section_name	Value	Default Value	Relevance and Contribution
CustomerProfileClassName	ORAPPS_ INTEGRATE	'Default'	Oracle Receivables default profile	Indicates the customer profile
CommitSize	IMPORT	<integer>	500	Indicates the number of 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
DefaultPriceGroupID	ORAPPS_ INTEGRATE	Price_Group_ ID	Null	Price_Group in Oracle Configurator Database is applied
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 Database 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.
IMPORT_ITEM_TYPE	ORAPPS_ INTEGRATE	<integer>	0	The item type id assigned to import items by default
MAJOR_VERSION	SCHEMA	System setting	System setting	Indicates the major version label for the Oracle Configurator Database schema created by CZ_SERVER.sql or GO_IMPORT_ONLY.sql
MaximumErrors	IMPORT	<integer>	10000	Default error limit for import runs before aborting
MINOR_VERSION	SCHEMA	System setting	System setting	Indicates the minor version label for the Oracle Configurator Database schema created by CZ_SERVER.sql or GO_IMPORT_ONLY.sql

Table 2–2 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 quotes. If this is set to "NO", you must also set: 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. 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.
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 export 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 Database from which an order will be imported back into Oracle Applications.
OrderTypeID	ORAPPS_ INTEGRATE	Order_Type_ ID	Null	Specifies the type of order in Oracle Applications that is being imported from the Oracle Configurator Database.

Table 2–2 CZ_DB_SETTINGS

Setting_ID	Section_name	Value	Default Value	Relevance and Contribution
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 an imported item in the Oracle Configurator Database 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 2–2 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.

2.4.0.3 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 Database and the tablespace to which <spx> and <spxdev> 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 DB_USER_ROLE <SPX_USER>. If no value is given, <SPX_USER>=‘SPX_USER’. The value can be any string (data type VARCHAR2). The value of the Setting_ID **1** is used for the DB_USER_ROLE <SPX_DEVELOPER>. 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 imported with GO_IMPORT.sql or RunImport.sql.

When granted to a user, this role gives the user access to the Oracle Configurator Database with RESOURCE privileges. <SPX_USER> is granted to <spx> by SP_BOOTSTRAP.sql. See Section 2.3.2, "Create OSPC Users and Responsibilities" on page 2-7 for more information about the access privileges of the default <SPX_USER>.

2.4.0.4 DB Settings for DATABASE_OWNERS

The settings in the DATABASE_OWNER section list names of the Oracle Applications <apps> and Order Entry owner <oe>, and the database link to

<appsid> if it is remote. These CZ_DB_SETTINGS are inserted automatically by OSPC SQL*Plus scripts such as InstAppsIntegrateViaLink.sql script.

2.4.0.5 DB Settings for SCHEMA

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

MAJOR_VERSION is the major version label for the Oracle Configurator Database schema created by CZ_SERVER.sql or GO_IMPORT_ONLY.sql.

MINOR_VERSION is the minor version label for the Oracle Configurator Database schema created by CZ_SERVER.sql, GO_IMPORT.sql, or GO_IMPORT_ONLY.sql.

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 also Section 2.6.2, "Redo Sequences" on page 2-37.

Replication is a YES / NO flag (default=NO) that determines whether replication is enabled or not on the Oracle Configurator Database.

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 8iLite, 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 <spx>, <spxdev>, and end users for permanent tables. The value of the SCHEMA SpxDefaultTablespace is <defaultspace>.

SpxTemporaryTablespace holds the default temporary tablespace used by <spx>, <spxdev>, 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>

2.4.0.6 DB Settings for ORAAPPS_INTEGRATE

The settings in the ORAPPS_INTEGRATE section control how and what gets imported and exported to and from the Oracle SellingPoint application.

AUTOCREATE_IMPORTED_USERS is a YES / NO / NAMED_ONLY flag (default=NO) that indicates whether, at the end of an 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 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 OSPC SQL*Plus script EndUsers.sql.

If the setting of AUTOCREATE_IMPORTED_USERS is 'YES', all imported users will be created as database users. For users with no login names in the Oracle Applications database (LOGIN_NAME=null), the import assigns names (e.g., <spx_*n*>). To import only users with existing login names, use the value 'NAMED_ONLY'. Specifying 'NAMED_ONLY' means all imported users whose LOGIN_NAME is not null are added to the CZ_END_USERS table in the Oracle Configurator Database 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 created as database users will have a login of CZ_END_USERS.LOGIN_NAME and a password of CZ_END_USERS.LOGIN_NAME.

Note: The end user role created when SP_BOOTSTRAP.sql is run is stored in the CZ_DB_SETTINGS table (Section_name = DB_USER_ROLE) in the VALUE column (e.g., SPX_USER). Make sure this role is setup in the database and that it has all of the grants and privileges specified in the SP_BOOTSTRAP.sql.

Table 2–3, "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 2–3 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.

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

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

DefaultPriceGroupID is the <price_group_id> (default='-1' (null)) of a price group defined in the Oracle Configurator Database 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 Database.

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 Database are exported with the order into Oracle Applications Order Entry.

DefaultSOPriceID is the <Orig_Sys_Ref> (default='-1' (null)) of an empty price list defined in the Oracle Applications database that will be imported into the Oracle Configurator Database 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, so prices come from a user-selected price group in the Quote module of the Oracle SellingPoint application.

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 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 imported into the Oracle Configurator Database 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 Database have Oracle Pricing applied or not. If ’YES’, Oracle Applications Order Entry will calculate and price the order itself. If ’NO’, Oracle SellingPoint application prices will override prices in the Order Entry system.

OrderEntry is a YES/NO flag (default=NO) that determines whether integration with **Oracle Applications Order Entry** 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 imported 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 Database from which an order will be imported into Oracle Applications Order Entry. In the Oracle Configurator Database that is identified as the import source: use_ids=true.

OrderImportSourceID must be populated for Oracle Applications 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 imported from the Oracle Configurator Database. OrderTypeID must be populated for Oracle Applications Order Entry to work (OrderEntry='YES'). Set before 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 Database. 'Segment1' is used by default so that the name loaded into the model structure in Oracle SellingPoint 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 Database. 'Segment1' is the usual field that contains the name for an item, so this is the default for retrieving name to be displayed in the Item Master in Oracle SellingPoint 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 import in the CZ_XFR_PROJECT_BILLS table in the Oracle Configurator Database at the time of import.

The OSPC import scripts load bills and items based on top bills listed in the CZ_XFR_PROJECT_BILLS table in the Oracle Configurator Database. 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 import scripts will import the BOMs that are flagged for import in the CZ_XFR_PROJECT_BILLS table without running the BOM exploder first.

2.4.0.7 DB Settings for IMPORT

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

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 import runs before aborting. If you have a large amount of data to import or you aren’t concerned with import stopping once a certain number of errors is reached, set this to an extremely large number.

See Section 4.2.1, "DB Settings for Import" for more information about all DB settings that apply to data import.

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 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 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
’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

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 4.2.4, "Setup for Generic Import", on page 4-18 for more detailed information about all CZ_DB_Settings that apply to generic import.

2.5 Make a Copy of an Oracle Configurator DB

In order to make a copy of an Oracle Configurator Database, create a dump (.dmp) file. Oracle recommends using the exp80 command (logged in as the schema owner) to make a dump file of your Oracle Configurator Database before upgrading to a new release. A new release of OSPC 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 Database allows you to revert back to the existing database in the event of upgrade problems. See Section 2.6.5, "Upgrade the Oracle Configurator Database" for more information about upgrading.

The following sections explain how to import such a dump file to create a copy of your Oracle Configurator Database. This import procedure also allows you to create an Oracle Configurator Database from the sample database files provided on the Oracle SellingPoint Configurator CD. For information on how to create a dump file from an existing Oracle Configurator Database, see the Oracle documentation on the exp and exp80 commands.

2.5.1 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 in your Oracle Configurator Database 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 2.5.2, "Create an Empty Database".
3. The role SPX_USER must exist in your Oracle Configurator Database. This role is created by the script SP_BOOTSTRAP.SQL, which is run when creating a new Oracle Configurator Database DBOwner such as <spx> or <impdump>.
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.

2.5.2 Create an Empty Database

In order to import (or re-import) a dump file into your Oracle Configurator Database, you must create a schema that is empty of user tables and user objects.

The empty schema might be a previously used schema (account) or one you create from scratch.

1. **Create a Schema from Scratch**

To create a schema from scratch, use SP_BOOTSTRAP.sql. This script creates a CZ_DB_SETTINGS table containing the default DB_USER_ROLES setting SPX_USER.

   a. Create a DBOwner with SP_BOOTSTRAP.sql by following the steps in Section 2.3.1.1, "Instructions for Running SP_BOOTSTRAP". Instead of <spx>/<spxpass>, you create <impdump>/<impdumppass>.

      Example:

      SQL> @SP_BOOTSTRAP <impdump> <impdumppass> <impdump> <impdumppass> <defaultspace> <tempspace> <imp> <imppass> <impdefaultspace> <imptempspace> <SPX_USER> <spxsid>

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 Database schema such as <spx>:

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

      Example:

      ```
      SQL> conn <spx>/<spxpass>@<spxsid>
      ```

   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 2.5.3, "Import a Dump File".

2.5.3 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. If you are importing the OSPC sample dump files, you must give the following values for <expdump> and <dumpfile>:

Argument	Value
<expdump>	docmark (for TutorialServer.dmp) hometheater_demo (for hometheater_server.dmp)
<dumpfile>	TutorialServer.dmp hometheater_server.dmp

   Example of importing the TutorialServer.dmp in NT:

   ```
   C:\>imp80 impuser/myword@serv1 fromuser=docmark touser=impuser
   recordlength=4096 file=TutorialServer.dmp log=TutImp.log
   ```

5. 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.

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

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

7. 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 for TutorialServer.dmp (note case-sensitivity):

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

8. Commit the change to the database:

   ```
   SQL> COMMIT;
   ```

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

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

2.5.4 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 OSPC database 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 OSPC database, but any database requires similar verification after being imported from a dump file.

2.5.5 Using an Imported Dump File

Before you can log into Oracle SellingPoint 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 3.5.3, "Create DSNs and DBOwners (Oracle8 and Oracle8i Lite)").
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>
```

2.5.6 OSPC Sample Data Dump Files

The Oracle SellingPoint Configurator installation includes two populated sample databases:

Tutorial	Item master, configuration, and UI data that supports the *Oracle Configurator Developer Tutorial*.
HomeTheater	Item master, configuration, and UI data from a home electronics demo.

The sample database files provided are:

- TutorialLite.odb (Oracle8*i* Lite)
- HomeTheaterDemo.odb (Oracle8*i* Lite)
- TutorialServer.dmp (Oracle8*i* Enterprise Edition with Oracle Client 8.0)
- hometheater_server.dmp (Oracle8*i* Enterprise Edition with Oracle Client 8.0)

The .odb files are installed automatically by the Oracle SellingPoint Configurator installation. If you have Oracle8*i* Lite installed on your machine, and DSNs properly set to these files in your `spx.ini` file, you can use these sample database in Oracle SellingPoint Developer and the Oracle SellingPoint application.

The .dmp files are located in the `Samples` directory on the Oracle SellingPoint Configurator CD. These files were created with the Export utility (exp or exp80) of Oracle8*i* Enterprise Edition, Release 8.0.5.1.0. The exported record length is 4096. Use imp (UNIX) or imp80 (NT) to import these .dmp files into your Oracle Configurator Database (see Section 2.5.3, "Import a Dump File", in particular Step 4 on page 2-32).

2.6 Oracle Configurator Database Maintenance

Oracle SellingPoint Configurator provides SQL*Plus scripts for managing an Oracle Configurator Database and its various subschemas.

Table 2–4 Oracle Configurator Database Maintenance Scripts

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

These scripts 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 scripts may invoke additional administrative packages, such as PURGE, REDO_SEQUENCES, TRIGGERS_ENABLED, or CONSTRAINTS_ENABLED to perform specific functions.

2.6.1 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 file of your database using Oracle8 exp80. See Section 2.5, "Make a Copy of an Oracle Configurator DB" for information about creating a dump (.dmp) file.
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 SellingPoint Developer.

2.6.2 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.

See Section 2.5.5, "Using an Imported Dump File", Step 1 on page 2-34 for an example of using REDO_SEQUENCES.

2.6.3 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.

2.6.4 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.

2.6.5 Upgrade the Oracle Configurator Database

If you are upgrading the Oracle Configurator Database from Release 4.1 or 4.1.1 to Release 4.2, use the following procedure:

1. Optionally, make a copy of your Release 4.1 or 4.1.1 Oracle Configurator Database, in case you need to revert.
2. To upgrade, connect to your Oracle Configurator Database

   For example:

   ```
   SQL> conn <spx>/<spx>@<spxsid>
   ```

   where `<spx>` is the owner (DBOwner) of the Oracle Configurator Database, and `<spxsid>` is the name for the Oracle8*i* Enterprise Edition instance on which the Oracle Configurator Database is installed.
3. You are running SQL*Plus in the <OSPC-scripts> directory. See Section 1.7.2, "Run SQL*Plus in the <OSPC-scripts> Directory" on page 1-14 for instructions on how to do this.
4. Run the script UPGRADE_SERVER.sql

For example:

```
SQL> @UPGRADE_SERVER
```

5. Let the script run. UPGRADE_SERVER.sql makes the necessary schema changes, commits them, and spools the output to a log file in the chosen directory (<OSPC-scripts> in Step 3, above).

6. To verify that the script incremented to the Release 4.2 schema (14b), select the version settings from the CZ_DB_SETTINGS table of your upgraded Oracle Configurator Database.

   For example:

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

   The result should be MAJOR_VERSION = 14, MINOR_VERSION = b.

3

Client Administration

Oracle SellingPoint Developer and the runtime Oracle SellingPoint application are client processes. These products require administration for all installations of OSPC, but especially development and deployment.

3.1 Overview of Client Administration

Client administration involves setting up a client machine with Oracle Client 8.0 and installing Oracle SellingPoint Configurator. Installing Oracle SellingPoint Configurator for an implementation, test, or maintenance environment consists of installing Oracle SellingPoint Developer (which includes Oracle SellingPoint application).

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

3.2 Oracle SellingPoint Developer Client Requirements

Oracle SellingPoint Developer runs on a client machine for:

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

The usual setup is Oracle SellingPoint Developer running on a client machine networked to a server where the Oracle Configurator Database is installed. The requirements for a networked setup are:

- The Oracle Configurator Database is running on the server. See Chapter 2, "Server Administration".

- Oracle Client 8.0 is installed on the client machine running Oracle SellingPoint Developer. See Section 3.4, "Oracle Client 8.0 Installation", on page 3-3.
- The client machine is configured to connect to the Oracle Configurator Database on the server machine (Oracle Net8 Easy Config). See Section 3.5.2, "Enable Client for Database Connectivity", on page 3-6.
- A datasource name for the Oracle Configurator Database on the server machine is defined in ODBC Administrator on the client machine. See Section 3.5.3, "Create DSNs and DBOwners (Oracle8 and Oracle8i Lite)", on page 3-7.
- The user logged into the client machine is a user defined in Oracle8*i* Enterprise Edition running the Oracle Configurator Database on the server.
- If Oracle SellingPoint Developer is being integrated with Oracle Applications for export to Order Entry, 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 Database on the server. See Section 3.5.4, "Set Parameters in spx.ini", on page 3-9.

3.3 Oracle SellingPoint Application Client Requirements

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 Oracle SellingPoint application.
- End-users using a deployed Oracle SellingPoint application in a networked environment.

Setup for running the Oracle SellingPoint application on a networked client during development, testing, maintenance, support, and client/server deployments is essentially the same as that for Oracle SellingPoint Developer on a machine networked to a server. The requirements for a networked setup of the Oracle SellingPoint application are:

- The Oracle Configurator Database is running on the server.
- Oracle Client 8.0 is installed on the client machine running the Oracle SellingPoint application. This is necessary for initializing replication (Oracle Configurator Mobile Database).

- The client machine is configured to connect to the Oracle Configurator Database on the server machine (Oracle Net8 Easy Config). See Section 3.5.2, "Enable Client for Database Connectivity".
- A datasource name (DSN) for the Oracle Configurator Database on the server machine is defined in ODBC Administrator. See Section 3.5.3, "Create DSNs and DBOwners (Oracle8 and Oracle8i Lite)".
- The end user logged into the client machine is an end user defined in Oracle8*i* Enterprise Edition running the Oracle Configurator Database on the server. See Section 5.3, "End User Administration".
- The end user logged into the client machine is an end user included in the CZ_END_USERS table of the Oracle Configurator Database.
- If Oracle SellingPoint application is being used integrated with Oracle Applications, the user logged into the client machine is a user defined in the Oracle Applications database.
- 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 Database on the server. See Section 3.5.4, "Set Parameters in spx.ini".

For information about setting up a mobile deployment of the Oracle SellingPoint application, see Chapter 7, "Mobile Deployment Administration".

3.4 Oracle Client 8.0 Installation

Oracle Client 8.0 allows the OSPC user to access an Oracle8*i* Enterprise Edition database from a client machine not running Oracle8*i* Enterprise Edition. Networked client machines must have Oracle Client 8.0 installed if they access the Oracle8*i* Enterprise Edition Server database. Mobile client machines need Oracle Client 8.0 installed to initialize replication.

Oracle Client 8.0 is used to access the Oracle Configurator Database from the client machine.

3.5 Set Up Oracle SellingPoint Configurator

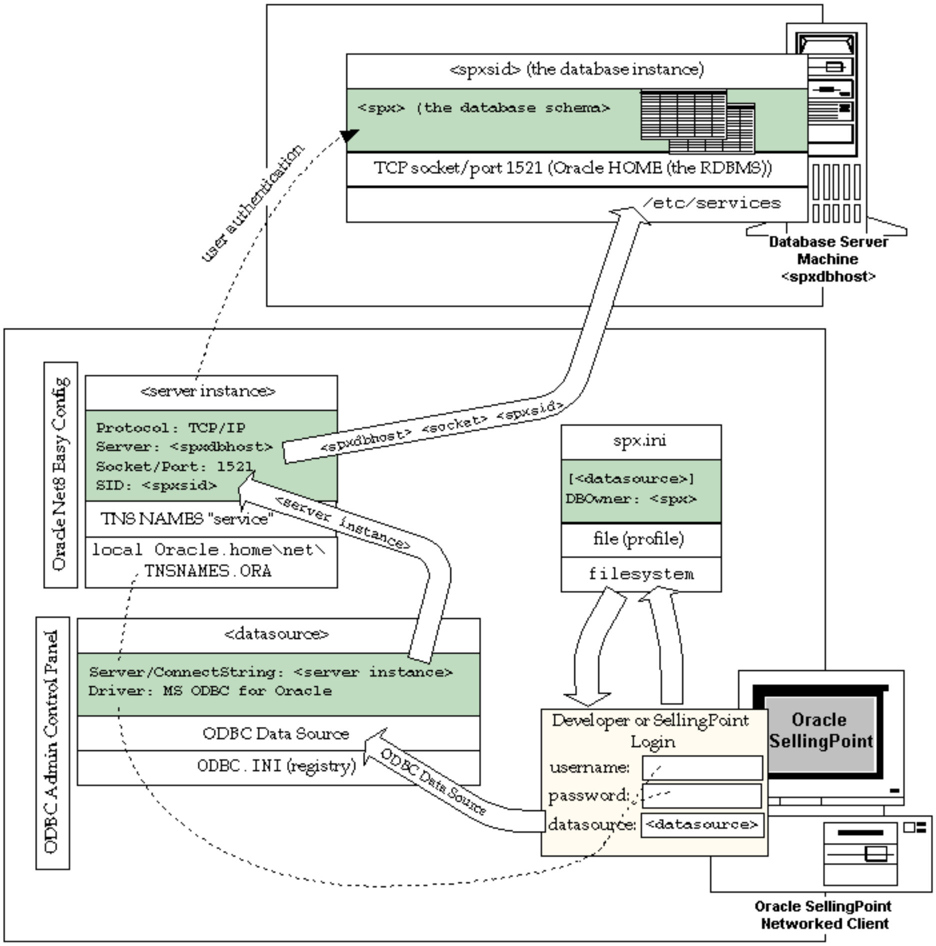
Once your site has Oracle8*i* Enterprise Edition Server and Oracle Client 8.0 installed and you have created the Oracle Configurator Database schema (e.g., CZ_ SERVER.sql), you can install and set up OSPC for your users.

The tasks required to set up OSPC for your users are:

- Install Oracle SellingPoint Configurator
- Enable Client for Database Connectivity
- Create DSNs and DBOwners (Oracle8 and Oracle8i Lite)
- Set Parameters in spx.ini
- Run Developer and the Oracle SellingPoint Application

Figure 3–1, below, illustrates an overview of the architecture of client/server data communication you set up with these tasks.

Figure 3–1 Client/Server Data Communication Architecture



3.5.1 Install Oracle SellingPoint Configurator

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

When creating or establishing an instance of the Oracle Configurator Database, the DBA creates and 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 Database. This is done for you when you create an Oracle Configurator Database (e.g., with GO_IMPORT.sql or CZ_SERVER.sql). Data import populates the Oracle Configurator Database 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 5.3, "End User Administration" for details.

3.5.2 Enable Client for Database Connectivity

Each machine running Oracle SellingPoint Configurator 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 SellingPoint Configurator on a client machine, the client machine needs data connectivity to the Oracle SellingPoint Database 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 name of the database instance or system identifier (SID) where the server Oracle SellingPoint Database is located (<spxsid>)
- the name of the physical server machine where <spxsid> is located (<spxdbhost>)
- the port of the physical server machine through which the client connects to the Oracle SellingPoint Database (<port>)
- the username / password of the DBOwner of the Oracle SellingPoint Database schema to which the client is connecting (<spx> / <spxpass> or <spxdev> / <spxdevpass>).

These are the parameters you must set 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).
2. Install Oracle Net8 Easy Config using the Oracle Installer. Select Add New Service and enter the new service name**.** This is the name of the database instance containing the server Oracle SellingPoint Database you will be connecting to (<spxsid>). 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 (<spxdbhost>) and the Port Number (<port>). The host name is the name of the physical server machine where the server Oracle SellingPoint Database is located. Make a note of the Port Number (default is 1521) for use in OSPC SQL*Plus scripts such as InstAppsIntegrateViaLink.sql. Click Next.
5. Enter the Database SID (<spxsid>). This is the name of the database instance containing the server Oracle SellingPoint Database you will be connecting to, the same name as the New Service you 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 server Oracle SellingPoint Database 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.

3.5.3 Create DSNs and DBOwners (Oracle8 and Oracle8*i* Lite)

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 mobile), the owner for that installation must also be defined in Oracle8*i* Enterprise Edition server. Mobile production installations require a DSN and an owner in Oracle8*i* Lite for the production Oracle Configurator Mobile Database.

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

For an Oracle Client 8.0 database, use the ODBC driver Microsoft ODBC Driver for Oracle.

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 SellingPoint Database, 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 Database (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 (<spxsid>) 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.

3.5.4 Set Parameters in `spx.ini`

The `spx.ini` file sets the DBOwner and other parameters for running Oracle SellingPoint Developer and Oracle SellingPoint application. Oracle SellingPoint Developer and Oracle SellingPoint application require that the DSNs defined in the `spx.ini` file map to an installed Oracle Configurator Database. The DSNs set in the `spx.ini` file must also be registered in the ODBC Administrator for each machine running Oracle SellingPoint Developer or Oracle SellingPoint application.

During installation of Oracle SellingPoint 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 SellingPoint Configurator. The file contains three [DSN] entries (used by Oracle SellingPoint Developer) and three [MDADSNS] entries (used by the Oracle SellingPoint application) for the three sample databases installed with Oracle SellingPoint 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 Database(s). The entries then appear in the Oracle SellingPoint Developer list of available data sources when you log in to Oracle SellingPoint Developer. You must create the Oracle Configurator Database DSN yourself, following the instructions in Section 3.5.3, "Create DSNs and DBOwners (Oracle8 and Oracle8i Lite)"; 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 Database DSNs available to the Oracle SellingPoint application [MDADSNS] and DBOwner.

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

Example 3–1 Example spx.ini File

```
[Merlin]
DBOwner=spx
ShowWarnings=false

[DSN]
SpxLite
TutorialLite
vis11

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

[MDAPLUGINS]

[MDADSNS]
vis11_replica
vis11
SpxLite
TutorialLite

[SpxLite]
DBOwner=spx

[TutorialLite]
```

```
DBOwner=spx

[vis11]
DBOwner = spx

[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=
```

3.5.4.1 [Merlin]

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

DBOwner in [Merlin]

The parameter DBOwner in the section [Merlin] specifies the default username of the owner of the Oracle Configurator Database schema that this spx.ini file accesses when starting up Oracle SellingPoint Developer. Users log into Oracle SellingPoint Developer with the schema name <spx> and the password <spxpass> established when the schema was created (see Section 2.3.1, "Create Oracle Configurator DBOwner" on page 2-5).

This parameter must be updated to specify the actual DBOwner of the Oracle Configurator Database containing your Oracle SellingPoint Developer Project(s). Every database specified by a Data Source Name (see Section 3.5.4.2, "[DSN]", below) is associated with this DBOwner, unless another DBOwner is specified explicitly for both Oracle SellingPoint Developer and the Oracle SellingPoint application with the setting:

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

See Section 3.5.4.6, "[<DSN>]", below.

ShowWarnings in [Merlin]

The parameter ShowWarnings in the section [Merlin] indicates whether or not generation logic warnings should be displayed in a warning dialog. If True, warnings display in a warning dialog until you dimiss 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.

3.5.4.2 [DSN]

The section `[DSN]` lists the Data Source Names for the Oracle Configurator Databases available for use by Oracle SellingPoint Developer. The DSNs listed in the default `spx.ini` file are those installed as part of the OSPC installation. Note, the only databases that can be installed from CD-ROM are Oracle8*i* Lite databases.

Once a server Oracle Configurator Database has been created for use with Oracle SellingPoint Developer, its Data Source Name must be listed here. Furthermore, a section must be created for the server DSN listing the DBOwner by which users will access the server Oracle Configurator Database (see Section 3.5.4.6, "[<DSN>]").

SpxLite is a blank Oracle8*i* Lite version of the Oracle Configurator Database. You must have installed Oracle8*i* Lite before installing Oracle SellingPoint Configurator to use any Oracle8*i* Lite database.

TutorialLite is an Oracle8*i* Lite Oracle Configurator Database populated with data to run the Tutorial example in Oracle SellingPoint Developer and the Oracle SellingPoint application. The Tutorial book is available as a Portable Document File (.pdf) in the `Doc` folder on the Oracle SellingPoint Configurator CD-ROM.

In Figure 3–1, "Example spx.ini File", vis11 is a server Oracle Configurator Database that has been replicated to a client. See Chapter 7, "Mobile Deployment Administration".

3.5.4.3 [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 Database that this `spx.ini` file accesses when starting the Oracle SellingPoint application.

The DBOwner username is automatically inserted in the Oracle Configurator Database CZ_END_USERS table when the Oracle Configurator Database schema is created (see Section 2.3.1, "Create Oracle Configurator DBOwner"). Users or end-users log into the Oracle SellingPoint application 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). During deployment, this is the name of an imported or created user (see Section 2.3.2, "Create OSPC Users and Responsibilities" on page 2-7).

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 SellingPoint Developer and the Oracle SellingPoint application with the setting:

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

See Section 3.5.4.6, "[<DSN>]".

DSN in [MDA]

The parameter `DSN` in the section `[MDA]` stores the last Oracle Configurator Database logged into by the Oracle SellingPoint application user or end-user 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 SellingPoint 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 Database, and all screens are created at startup. 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 application starts up. A value of 'Early' causes the configuration engine to start up when the application does. Any other value causes the configuration engine to start on demand when first creating a configuration.

3.5.4.4 [MDAPLUGINS]

The section [MDAPLUGINS] lists programs that can be launched from within the Oracle SellingPoint application. THIS FEATURE IS NOT CURRENTLY IMPLEMENTED.

3.5.4.5 [MDADSNS]

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

The DSNs listed in the default spx.ini file are those installed as part of the OSPC installation. Note, the only databases that can be installed from CD-ROM are Oracle8*i* Lite databases (see Section 3.5.4.2, "[DSN]").

In Figure 3–1, "Example spx.ini File", vis11 is the DSN of an Oracle Configurator Database that has been replicated to a client, and vis11_replica is the DSN of the replica database. The DSN for a replica database always consists of the server database DSN followed by `_replica`.

3.5.4.6 [<DSN>]

If a DSN is not listed for every Oracle Configurator Database to which you need access from Oracle SellingPoint 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 Database associated with the DSN will be accessed. Since the default install of OSPC includes two Oracle8*i* Lite databases, these are each included as `[<DSN>]` sections (e.g., `[SpxLite]`) with the DBOwner `spx`.

```
[<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.

3.5.4.7 [<DSN>_replica]

In Figure 3–1, "Example spx.ini File", `[vis11]` is the DSN section for an Oracle Configurator Database that has been replicated to a client, and `[vis11_replica]` is the DSN section for the 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 Database from which this replica was created.

Master_schema_name

The parameter `master_schema_name` specifies the schema name of the Oracle Configurator Database 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 determines if the replica client and the server database are out of sync, and updates just that data completely. The value FAST refreshes only changes in the replica client to the server database. The value COMPLETE executes a complete synchronization of all data, not just changes. This setting may 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.

3.5.4.8 [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.

3.5.4.9 [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 SellingPoint Developer.

3.5.4.10 [Test]

The section `[Test]` sets the type of environment to launch when using the Test/Debug button in Oracle SellingPoint Developer. Launch=1 (default) specifies the Oracle SellingPoint application as the test environment running the Oracle Configurator application. 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. These parameters may also be set in the Tools --> Options --> Test dialog in Oracle SellingPoint Developer.

3.5.5 Run Developer and the Oracle SellingPoint Application

Using the base `spx.ini` file for development, start Oracle SellingPoint Developer. Log in as the DBOwner (<spx>/<spxpass>) or <spxdev>/<spxdevpass> or an imported/added user listed in the CZ_END_USERS table with SPX_USER role permissions (or better) granted and who is enabled as a database user.

To run the test Oracle SellingPoint application from Oracle SellingPoint Developer (click Test after Generate Active Model and Generate Active UI), login in as a user listed in the CZ_END_USERS table with SPX_USER role permissions (or better) granted and who is enabled as a database user

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

3.6 Create Oracle8*i* Lite Versions of an Oracle Configurator Database

Several Oracle8*i* Lite databases are included on the Oracle SellingPoint Configurator CD and do not need to be created. These Oracle8*i* Lite databases are used for validation, demonstrations, and running the Oracle SellingPoint Developer tutorial.

Running SPX_Lite.sql allows you to create the primary database objects that constitute an Oracle8*i* Lite Oracle Configurator Database. This is not the same as the

Oracle Configurator Mobile Database (a replica of the Oracle Configurator Database) that you will need for mobile deployment on laptops that synchronize with a server database. SPX_Lite.sql is used for creating Oracle8*i* Lite Oracle Configurator Databases supporting Oracle SellingPoint application prototypes and demonstrations.

When creating an Oracle8*i* Lite database for a mobile deployment of an Oracle SellingPoint application, the DBA installs the replication objects in the Oracle Configurator Database on the server. When the end user selects Initialize Remote DB, the Oracle Configurator Mobile Database synchronizes with the Oracle Configurator Database on the server.

4

Data Import

Oracle SellingPoint applications use a standard schema for configuration data referred to as the Oracle Configurator Database. The Oracle Configurator Database is used to store customer information and all information relative to the configuration model — product data, project structure, configuration rules, and user interface layouts. Customer and product-related data, such as price lists, are generally imported into the Oracle Configurator Database from data sources external to Oracle SellingPoint Configurator.

The two methods for importing external data are:

- ***Direct import*** of data from specific Oracle Applications database tables (e.g., BOM, Item Master, customer, address, user, and contact)
- ***Generic import*** of legacy data

Generic import of legacy data includes importing data from the Oracle Applications database not covered in direct import (e.g., pricing).

Regardless of which method you use, data import refreshes the Oracle Configurator Database with changes and updates for enterprise-wide consistency. Configuration models in the Oracle SellingPoint application maintain all relationships associated with the refreshed data to minimize model maintenance as data is updated.

4.1 Overview of Data Import

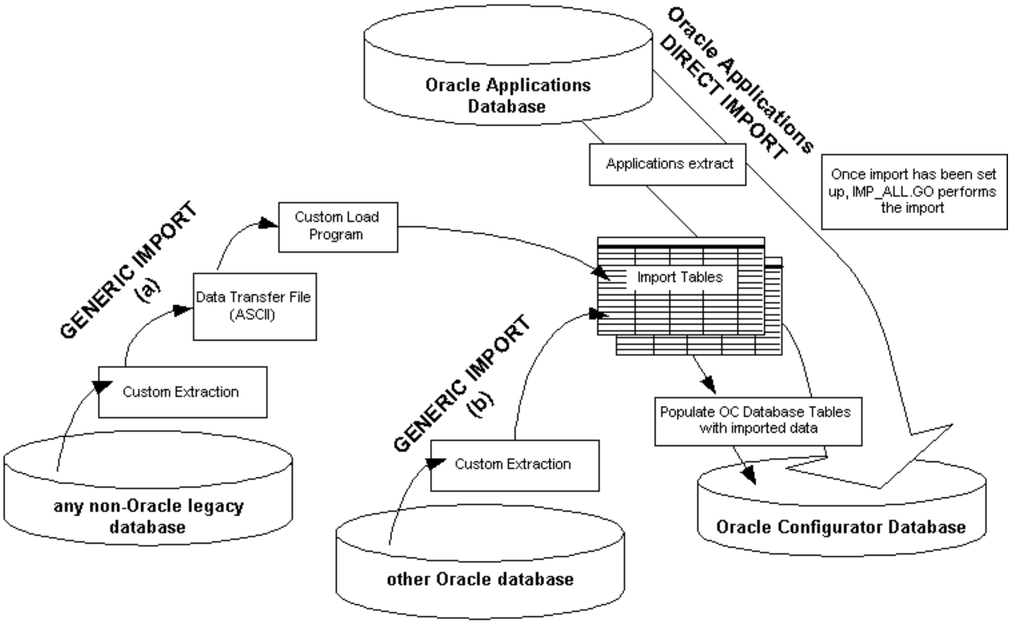
The DBAs need to understand how enterprise or legacy data is used in the Oracle SellingPoint application and how the application is refreshed when enterprise or legacy data changes.

In order to import data into your Oracle SellingPoint application, there must be ***import tables*** in your Oracle Configurator Database. See Section 2.3.3, "Create the Oracle Configurator Database" and Appendix B.2, "List of Import Tables" on page B-1 for information about how to create an Oracle Configurator Database and which import tables should be included in it.

There are several different kinds of data import. You can import data into a single Oracle Configurator Database table (for testing). You can import data from specific Oracle Applications tables (direct import). Or you can import legacy data from other Oracle Applications tables, another Oracle database, or a non-Oracle database (generic import). It is possible to use both the direct and generic methods on one OSPC project to achieve the results you need.

For direct import, scripts are available for executing any and all of these events automatically. For generic import you must create the extraction programs, populate the import tables, and use the scripts available for populating the Oracle Configurator Database tables with the data from the import tables.

Figure 4–1 Overview of Data Import


Oracle Applications Database
Oracle Applications DIRECT IMPORT
Applications extract
Once import has been set up, IMP_ALL.GO performs the import
GENERIC IMPORT (a)
Custom Load Program
Data Transfer File (ASCII)
Custom Extraction
any non-Oracle legacy database
Import Tables
GENERIC IMPORT (b)
Custom Extraction
other Oracle database
Populate OC Database Tables with imported data
Oracle Configurator Database

As you can see in Figure 4–1, both generic and direct import use the import tables in the Oracle Configurator Database to transfer the extracted data to the Oracle

Configurator Database tables used by the application. Exactly which extracted data are imported depends on the settings in the control tables (CZ_XFR_ tables in the Oracle Configurator Database) or your custom load program, if applicable.

The OSPC SQL*Plus scripts do not provide an automated or scheduled mechanism that clears the import tables.

4.1.1 Import Tables

Import tables are the schema of tables used by the Oracle SellingPoint Configurator software to manage and pass data to destination tables in the online schema of Oracle Applications. The destination tables are also referred to as online tables. The import table schema shadows the schema of online tables that are targets of a data import. Import tables pass data from field to field and have the same name as their online schema counterparts, plus an additional _IMP. For example, the import table responsible for passing data to CZ_PROPERTIES is named CZ_IMP_PROPERTIES. In addition, import tables contain the same fields that exist in their online table counterpart, plus additional fields used specifically for the controlling the import process. All import table fields are nullable and no data constraints are applied to any of them, however, some fields must be populated with data to support successful configuration model development or use.

Each import table includes the following types of fields:

- Import Control
- Destination Data
- Surrogate Key

4.1.1.1 Import Control Fields

Import control fields are fields containing data that is used only to manage the Import process for each record. Control fields are not transferred to the destination database and are not used to resolve key values or anything else. The import control fields are:

Table 4–1 Import Table Field Groups

Field Name	Type	Description
RUN_ID	INTEGER	Input field that identifies with which run this record is associated

Table 4–1 Import Table Field Groups

Field Name	Type	Description
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 exists in the online database.

DISPOSITION	CHAR(1)	Output field that indicates the disposition of the record:
		I = Insert
		M = Modify
		N = No change
		R = Rejected
		Null indicates not yet known

4.1.1.2 Destination Data Fields

Destination Data Fields in import tables are fields that 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 user-provided "extrinsic" identifications for data to be imported. These fields include both surrogate primary keys and foreign surrogate keys. A "foreign surrogate 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, import tables contain a single field named ORIG_SYS_REF which is used to hold the imported 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 Data to Be Imported Through Direct Import

Customer and product data may be imported directly into the Oracle Configurator Database from Oracle Applications. When data is imported directly from the Oracle Applications database (e.g., the BOM_EXPLOSIONS table), it first populates the import tables in the Oracle Configurator Database and then the destination tables in the online schema of the Oracle Configurator Database.

It may be possible to adapt the import procedures for these tables for a non-Oracle Applications import.

If you are importing data directly from specific Oracle Applications database tables, you must:

1. Cleanse data for import.
2. Create an Oracle Configurator Database containing integration objects for import/export. See Section 2.3.3.2, "Create the Oracle Configurator Database for Integration" on page 2-11.
3. Prepare for direct import. See Section 4.2, "Import Setup" on page 4-8.
4. Run the appropriate import script, that performs the extraction into the correct format for import, loads the data into the import tables according to the set up, and populates the Oracle Configurator Database tables with imported data from the import tables. See Section 4.3, "Run Direct Import" on page 4-19 for specific information about which scripts to run.

The direct import method enables you to import the following type of data:

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

- Price List Identifiers (IDs)

The Oracle Applications data that can be imported by means of direct import are:

Table 4–2 Direct Import Source and Destination Tables

Oracle Applications Source Table	-->	Oracle Configurator Database Target 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

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

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

Direct import targets 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.

Direct import also targets all or a subset of Price List IDs from the SO_PRICE_LISTS table in the Oracle Applications database. PRICE_LIST_IDs imported from the Oracle Applications database SO_PRICE_LISTS table go into the CZ_PRICE_

GROUPS table in the Oracle Configurator Database. The Price List IDs are used by the Oracle SellingPoint application quote module when submitting quotes to Oracle Order Entry. Discounting can be performed in Oracle Order Entry. You can also choose not to use Oracle Applications Pricing and perform all pricing and discounting in the Oracle SellingPoint application.

Direct import automatically imports all Customer, Contact, Address, and Sales Rep data from the RA_ tables listed in Table 4–2.

4.1.3 Data to Be Imported Through Generic Import

For legacy data residing in non-Oracle databases or Oracle Applications data not accessible through direct import, Oracle SellingPoint Configurator provides a generic import mechanism. The generic import facility utilizes the import tables to load data into tables in the Oracle Configurator Database online 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 or Oracle Applications data not accessible through direct import, you must:

1. Identify and cleanse data for import.

2. Create an Oracle Configurator Database containing integration objects for import/export. See Section 2.3.3.2, "Create the Oracle Configurator Database for Integration" on page 2-11.
3. 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 4.2.4, "Setup for Generic Import" on page 4-18.

     OR
   - b. load the data generated by the custom extraction programs directly into the import tables. You can also use a utility (such as SQL*Loader) to load the data from your ASCII file to the import tables.
4. Run the import script, RunGenImport.sql, which populates the Oracle Configurator Database tables with imported data from the import tables. See Section 4.4, "Run Generic Import" on page 4-21.

4.2 Import Setup

Import setup consists of:

- making sure that data to be imported is clean, and in the case of BOMs, complete and identified at the desired root
- setting parameters in the CZ_DB_SETTINGS table in the Oracle Configurator Database
- identifying and modifying what gets imported in the CZ_XFR_ control tables in the Oracle Configurator Database

4.2.1 DB Settings for Import

You can adjust the way the import is executed with settings in the CZ_DB_SETTINGS table in the Oracle Configurator Database. The following settings are among those that are in effect during the import run:

AUTOCREATE_IMPORTED_USERS (direct)
RUN_BILL_EXPLODER (direct)
CommitSize (direct)
MaximumErrors (direct)
IMPORT_ITEM_TYPE (direct and generic)

See Section 2.4, "Set DB Settings" on page 2-16 for information about these and other DB_SETTINGS, including their default values.

You can specify whether you want to apply item prices in Oracle SellingPoint or in Order Entry by setting the value of OraclePricing in the CZ_DB_SETTINGS table. When OraclePricing=’NO’, you can specify prices in the Oracle SellingPoint application that remain static in the quote exported to Order Entry. To enable pricing in Oracle SellingPoint (OraclePricing=’NO’), you must also insert values for the DB_SETTINGS: DefaultPriceGroupID, DiscountID, OrderTypeID, and OrderImportSourceID. For Oracle SellingPoint-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 Entry of non-Oracle Applications prices created in the Oracle SellingPoint application. To bypass pricing in Oracle SellingPoint and perform final pricing in Order Entry, the OraclePricing flag must be set to ’YES’ and you must additionally set the OrderTypeID and OrderImportSourceID. See Section 2.4.0.6, "DB Settings for ORAAPPS_INTEGRATE" on page 2-23 for more information about Oracle Pricing.

The <SPX_USER> role listed in the DB_USER_ROLES section of the CZ_DB_SETTINGS table is assigned by default to all users defined in the CZ_END_USERS table, including imported Oracle Applications users. The DB_SETTING, AUTOCREATE_IMPORTED_USERS, controls which users are imported into the Oracle Configurator Database and whether or not they are defined as database users. See Section 2.4.0.6, "DB Settings for ORAAPPS_INTEGRATE" on page 2-23 for detailed information about AUTOCREATE_IMPORTED_USERS.

4.2.2 Control Tables (CZ_XFR_)

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

CZ_XFR_TABLES identifies the import table(s) that are enabled for import.

Table 4–3 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 4–3 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 table
DST_SUBSCHEMA	NO	INTEGER	Identifies the subschema in which the destination tables resides
FILTERSYNTAX	NO	VARCHAR (255)	Limits the records imported from the given source 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", transfer of data is disabled at the point in the procedure identified by XFR_GROUP. When XFR_GROUP = "EXTRACT", data is not loaded into the import tables. When XFR_GROUP = "IMPORT" or "GENERIC", data in the import table is ignored and not transferred to corresponding online tables.

The CZ_XFR_FIELDS table identifies rules for transferring fields during the import process. This table primarily specifies overrides to default transfer rules. If a field for which no entry exists in CZ_XFR_FIELDS is transferred to an import table, the field controls defaults of CZ_XFR_FIELDS are applied for that field.

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.

Table 4–4 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’, or ’GENERIC’
ORDER_SEQ	YES	NUMBER	Work with XFR_GROUP to identify the table import to which this record applies
FIELD_ORDER	YES	NUMBER	Distinguishes the order in which the field is imported
SRC_FIELD	NO	VARCHAR 2(40)	Name of the field you are transferring data from
DST_FIELD	NO	VARCHAR 2(40)	Name of the field you are transferring data to
REQUIRED	NO	CHAR(1)	If "1", this field is required by the destination 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 Database. Every imported bill must be represented in CZ_XFR_PROJECT_BILLS.

Direct import uses the CZ_XFR_ control tables in the Oracle Configurator Database. The TOP_ITEM_ID and ORGANIZATION_ID for each bill of materials to be imported from the Oracle Applications database are read from the CZ_XFR_PROJECT_BILLS table. The PS_NODE import updates the CZ_XFR_PROJECT_BILLS table with the timestamp and ID and description of the most recent import run.

Table 4–5 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
LAST_IMPORT_DATE	NO	DATE	Gives the date and time when this bill was last imported
SOURCE_BILL_DELETED	NO	CHAR(1)	Indicates (if ‘1’) that this bill has been deleted from Oracle Applications; ‘0’ if still active (default)
TOP_ITEM_ID	YES	NUMBER	Contains the item ID that identifies this top-level item in Oracle Applications
DELETED_FLAG	YES	CHAR(1)	Disables or excludes this bill from being imported if ‘1’; ‘0’ if still active.
EXPLOSION_TYPE	NO (by default)	VARCHAR2 (10)	Instructs the import software on the type of BOM explosion it should perform for this bill, if automatic explosion is enabled in the CZ_DB_ SETTINGS table. The BOM explosion can be ’OPTIONAL’ (default), ’INCLUDED’, or ’ALL’
BILL_REVISION_DATE	NO	DATE	Describes when the source data for this bill in Oracle Applications was 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 SellingPoint Configurator.

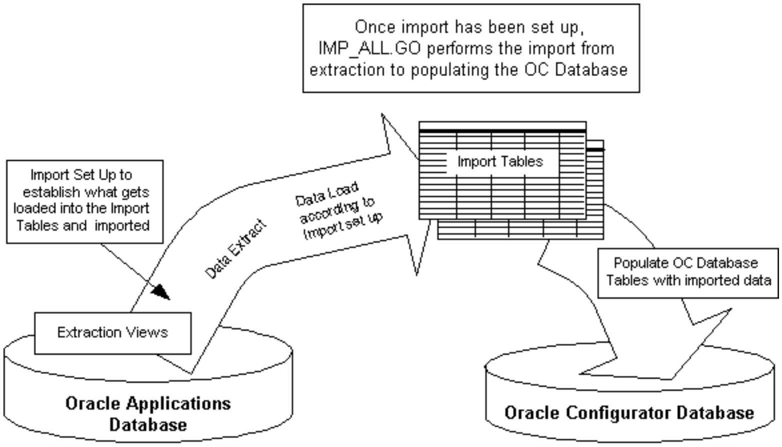
Table 4–6 CZ_XFR_PRICE_LISTS Fields

Field Name	Req'd	Type	Descriptions
PRICE_LIST_ID	YES	NUMBER	Identifies the price list in Oracle Applications
IMPORT_ITEM_PRICES	NO	CHAR(1)	Indicates whether to import item prices for this list but not display them in the Oracle SellingPoint application model (if '1'), import them so they appear in a quote (if '2'), or not import item prices (if '0') (default). If "2", the 'Quoteable' flag in CZ_ITEM_MASTERS must be set to '1' to have the prices display in Oracle SellingPoint.
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 when this price list was last imported
SOURCE_PRICE_DELETED	NO	CHAR(1)	Indicates (if '1') that the price list in this record has been deleted from Oracle Applications; '0' if still active
DELETED_FLAG	YES	CHAR(1)	Disables or excludes the import of this price list if '1'; '0' if the import is still active (default)

4.2.3 Setup for Direct Import

If your OSPC is integrated with Oracle Applications, you can do a direct import.

Figure 4–2 Overview of Direct Import


Once import has been set up, IMP_ALL.GO performs the import from extraction to populating the OC Database
Import Set Up to establish what gets loaded into the Import Tables and imported
Import Tables
Data Extract
Data Load according to Import set up
Populate OC Database Tables with imported data
Extraction Views
Oracle Applications Database
Oracle Configurator Database

Before executing a direct import of data from an Oracle Applications database, you must select what data will be extracted (identify which BOMs should be imported) and what tables will be involved in the import.

The Oracle Configurator Database 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 for use at the Product level in the Oracle SellingPoint application Model. One BOM is imported per Oracle SellingPoint Developer Project. The Developer Project name is defaulted from the BOM model name.

The ORGANIZATION_ID identifies which BOMs are imported. OSPC uses the ORGANIZATION_ID when exporting quotes to Order Entry. An order is only valid with the ORGANIZATION_IDs that correspond to the ORGANIZATION_IDs on all items of the BOM in Oracle Applications.

The BOM data is 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 call the procedure to explode BOMS and populate the BOM_EXPLOSIONS table if the RUN_BILL_EXPLODER setting in the CZ_DB_SETTINGS table is set to YES to indicate that the BOM should be exploded.

You further set up import by indicating in the control tables (CZ_XFR_ tables) what extracted data will be transferred from the Oracle Applications database to the Oracle SellingPoint Configurator import tables as well as what data will be transferred from the import tables to the online tables.

See Section 2.4, "Set DB Settings" on page 2-16 and Section 4.2.2, "Control Tables (CZ_XFR_)" on page 4-9 for more details on the control tables and settings used for setting up import.

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

4.2.3.1 Load All Bills

You can use the script LoadAllBills.sql to build your CZ_XFR_PROJECT_BILLS table with all BOMs and your CZ_XFR_PRICE_LISTS table with price lists from Oracle Applications. You can also control which Bills of Material (BOMs) or price lists will be imported by manually editing the CZ_XFR_PROJECT_BILLS and CZ_XFR_PRICE_LISTS control tables. If you use the script LoadAllBills.sql, and set the DELETED_FLAG in the CZ_XFR_PROJECT_BILLS and the CZ_XFR_PRICE_LISTS control tables to ’1’, you can exclude from the import process the Bills of Material and price lists that you don’t want imported. See Section 4.2.3.2.2, "To Load All Data and Identify Specific Items For Exclusion:" on page 4-16 for details about how to exclude specific Bills of Material or price lists from the import.

4.2.3.2 Selective Direct Import

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:

- specify (with a SQL insert command) specific items from the BOM and price list data to be imported
- load all data into the import tables, identifying specific items for exclusion in the control tables (CZ_XFR_).

Before running an import script for either of these methods, be sure you have completed the following:

1. An Oracle Configurator Database has been established. See Section 2.3, "Establish an Oracle Configurator Database" on page 2-4.

2. The Oracle Configurator Database contains integration objects for import/export. See Section 2.3.3.2, "Create the Oracle Configurator Database for Integration" on page 2-11.
3. Know exactly what data you are importing.
4. You are running SQL*Plus in the <OSPC-scripts> directory connected to the instance <spxsid>. See Section 1.7.2, "Run SQL*Plus in the <OSPC-scripts> Directory" on page 1-14 for instructions on how to do this.

4.2.3.2.1 To Specify Specific Items to be Imported From the BOM and Price List Data:

With this method of controlling import, you manually insert records in the control table that directs an import script to transfer that specific data from the import table to the destination table.

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 Database prices tables but don’t display them in the Oracle SellingPoint application), and '2' (import Oracle Applications item prices so that they can be used in the Oracle SellingPoint application). 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.

4.2.3.2.2 To Load All Data and Identify Specific Items For Exclusion:

With this method of controlling import, you load all data into the import table. You then edit the control fields of the control table that directs the import script to transfer the data from the import table to the destination table.

1. Run LoadAllBills.sql.

   Example:

   ```
   SQL> @LoadAllBills
   ```

   For details about what LoadAllBills.sql does, see Section A.2.6, "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_TABLE (and possibly CZ_XFR_FIELD).

   In the CZ_XFR_TABLE, the value of DST_TABLE establishes which destination tables in the Oracle Configurator Database 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 determines the order in which the participating tables are imported. See Appendix B.3, Section B.3, "Dependencies Among Import Tables"to help you determine a viable order.

   For example (entered all on one line):

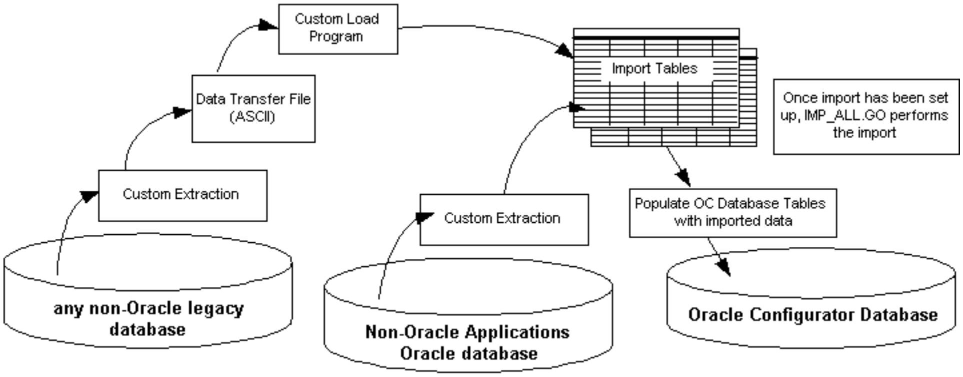
   ```
   SQL> update CZ_XFR_TABLE set DISABLED='1' where DST_TABLE='CZ_PS_NODES' and
   CZ_XFR_GROUP='EXTRACT';
   ```

In this example, you are disabling the extraction of data from the Oracle Applications CZ_IMP_PS_NODE table so that the transfer of data from CZ_IMP_PS_NODE to CZ_PS_NODES in the Oracle Configurator Database 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.

4.2.4 Setup for Generic Import

If your OSPC is not integrated with Oracle Applications and you are importing legacy data or Oracle Applications data not accessible through direct import, 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 Database. You must then develop a custom load program (or use a utility such as SQL*Loader) to populate the import tables in OSPC with that extracted data. The import procedure provided by Oracle SellingPoint Configurator then populates the Oracle Configurator Database from the import tables.

Figure 4–3 Overview of Generic Import

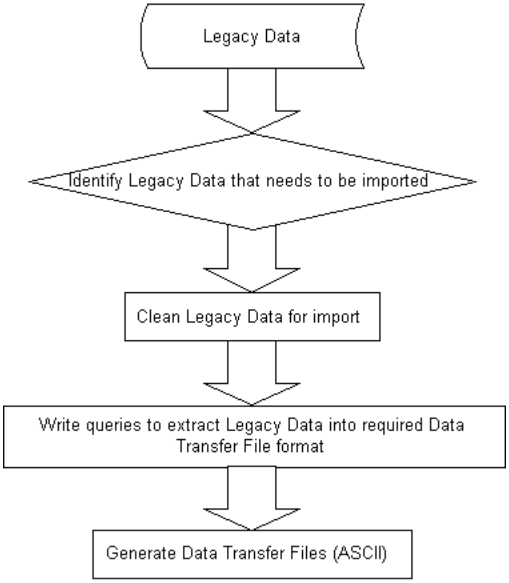

Custom Load Program
Import Tables
Data Transfer File (ASCII)
Once import has been set up, IMP_ALL.GO performs the import
Custom Extraction
Custom Extraction
Populate OC Database Tables with imported data
any non-Oracle legacy database
Non-Oracle Applications Oracle database
Oracle Configurator Database

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 B.5, "Import Tables" on page B-4 for a detailed presentation of all import table fields.

See Appendix B.4, "Required ASCII File Format for Generic Import" on page B-4, for information about the required format of the extracted data.

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 Database import tables, you must also provide the load programs that populate those import tables with the data contained in the Data Transfer Files. You can create a custom load program (or use a utility such as SQL*Loader) to populate the import tables.

4.3 Run Direct Import

You can directly import data for a single table or for many tables at once. In order to import data, import parameters must be set (see Section 4.2.3, "Setup for Direct Import" on page 4-13).

To populate the import tables and the Oracle Configurator Database with 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".

To populate the import tables and the Oracle Configurator Database with the data you have defined for importing multiple BOMs, run the RunImport.sql script. The source of the import is the data extracted from the database by means of Extraction Views (for Oracle Applications database) or a custom extraction program. The destination of the import is the Oracle Configurator Database.

4.3.1 Customizing 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 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 database for import into the Oracle Configurator Database. The result of executing these Extraction Views is data in the format required by the Oracle Configurator Database 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. CREATE_EXP_VIEWS.sql is called by GO_IMPORT.sql, InstAppsIntegrate.sql, and InstAppsIntegrateViaLink.sql when you create the Oracle Configurator Database schema. See Section 2.3.3.2, "Create the Oracle Configurator Database for Integration" for more information about these scripts.

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

Table 4–7 Direct Import Extraction Views

To modify import for...	Customize...
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.

4.4 Run Generic Import

If your OSPC is not integrated with Oracle Applications and you are generically importing data from another Oracle database or non-Oracle database, you are responsible for implementing a custom extraction of your legacy data and generating Data Transfer Files (in ASCII format) required by the Oracle Configurator Database import tables. You can create a custom load program (or use a utility such as SQL*Loader) to populate the import tables.

You must have established namespaces or packages for data import and the integration tables in the Oracle Configurator Database (see Section 2.3.3, "Create the Oracle Configurator Database" on page 2-10).

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

1. You are running SQL*Plus in the <OSPC-scripts> directory connected to the instance <spxsid>. See Section 1.7.2, "Run SQL*Plus in the <OSPC-scripts> Directory" on page 1-14 for instructions on how to do this.
2. A DBOwner for the import tables (<imp>) and online schema (<spx>) have been created. This step is handled by the script SP_BOOTSTRAP.sql, which also assigns the tablespace used by these schemas.
3. Grant connect and resource privileges to the DBOwner of the Oracle Configurator Database. This step is handled by the script SP_BOOTSTRAP.sql, which assigns the <SPX_USER> role to <spx>. See Section 2.3.2, "Create OSPC Users and Responsibilities" on page 2-7.

4. Run the GO_IMPORT_ONLY.sql to create the Oracle Configurator Database schema with integration objects. This creates a blank schema and includes all packages, procedures, etc. See Section 2.3.3.1, "Create the Oracle Configurator Database without Integration" on page 2-11 for additional information.

5. Be sure your legacy data is available in the ASCII file format required by the import tables. See Appendix B.4, "Required ASCII File Format for Generic Import" on page B-4.

6. You have extracted legacy data and loaded it into the Oracle Configurator Database import tables with a custom load program (or a utility such as SQL*Loader). See Section 4.2.4, "Setup for Generic Import" on page 4-18.

7. Set the 'DISABLED' setting in 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

8. Connect to the instance <spxsid> as the online user.

   Example:

   ```
   SQL> conn <spx> <spxpass> @<spxsid>
   ```

9. Import the data to the Oracle Configurator Database 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 Database 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 Database online schema.

4.4.1 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.

4.4.1.1 Re-Run an Entire Import Session

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

4.4.1.2 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 the RUN_ID for the session you are repeating in the RUN_ID column for those records.
2. Run the RunGenImport.sql script and press Enter at the prompt for records where RUN_ID is null or specify the special value entered.

4.5 Verify Import

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

4.6 Refresh and Update

Oracle recommends that changes to legacy data be limited during construction of an Oracle SellingPoint application to avoid potential problems introduced by interim data 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 SellingPoint application. Test cases may have to be updated to match the changes.

Although updating imported data in the Oracle Configurator Database 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 gets imported and/or corrections to the definitions of the configuration model in Oracle SellingPoint application. In many cases, a refresh causes previously imported data to be overwritten.

When importing an Oracle Applications BOM, the ORGANIZATION_ID and TOP_ITEM_ID populate the CZ_DEVL_PROJECTS.ORIG_SYS_REF to uniquely identify the BOM. Since there is only one BOM per Oracle SellingPoint Developer Project, and only one record for that BOM in the CZ_XFR_PROJECT_BILLS table, re-importing the BOM data refreshes the Project. The Project is associated with the BOM by its PS_NODE_ID in the MODEL_PS_NODE_ID field in CZ_XFR_PROJECT_BILLS. If you make a copy of the Project (Save Project As command in Oracle SellingPoint Developer), the copy has the same ORIG_SYS_REF as the original, but gets a new PS_NODE_ID. The record for that BOM in the CZ_XFR_PROJECT_BILLS table is not changed, however. Re-importing the BOM data refreshes the original project, not the copy.

In order to refresh the imported BOM of a copied Project, you must identify the PS_NODE_ID for the copied Project in the CZ_PS_NODES table and update the CZ_XFR_PROJECT_BILLS.MODEL_PS_NODE_ID field to match the copied Project's PS_NODE_ID. For any one BOM (identified by ORIG_SYS_REF), it is only possible to refresh one associated Project (identified by PS_NODE_ID) per import run because there can only be one record for that BOM in the CZ_XFR_PROJECT_BILLS table.

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

5 Production Deployment

For production deployment (including test production 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 SellingPoint Developer, use a project-specific installation program.

Each installed Oracle SellingPoint application runs on a 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 Database providing the data to run the Oracle SellingPoint application is commonly on a different (server) machine.

For information specifically about mobile deployments, see Chapter 7.

5.1 Requirements for Production Deployment

The requirements for a deployed client setup of the Oracle SellingPoint application are:

- The Oracle Configurator Database is running on the server providing the client with data needed by the runtime Oracle SellingPoint application.
- Oracle Client 8.0 is running on the client machine running the Oracle SellingPoint application.
- The client is configured to connect to the Oracle Configurator Database (Oracle Net8 Easy Config). See Section 3.5.2, "Enable Client for Database Connectivity", on page 3-6.
- A datasource name (DSN) for the Oracle Configurator Database is defined in the ODBC Administrator on the client. See Section 3.5.3, "Create DSNs and DBOwners (Oracle8 and Oracle8i Lite)", on page 3-7.

- The `spx.ini` file on the client is edited to include the correct DSN and DBOwner for the Oracle Configurator Database. See Section 3.5.4, "Set Parameters in spx.ini", on page 3-9.
- The end user logged into the Oracle SellingPoint application is an end user included in the CZ_END_USERS table of the Oracle Configurator Database.
- The end user logged into the Oracle SellingPoint application is associated with one Project (Active Model) and at least one account.
- If the Oracle SellingPoint application is being used integrated with Oracle Applications, the end user logged into the client is a user defined in the Oracle Applications database.

5.2 Deploying a Customized Oracle SellingPoint Application

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 SpRuntime.exe to install all of Oracle SellingPoint 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

5.2.1 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 OSPC 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 SellingPoint Developer and WebUI files are not needed for a production deployment.

During development, the following information is inserted into the Oracle Configurator Database that has dependencies with the local file system of the machine running Oracle SellingPoint 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 3-14 and Section , "SLOGO" on page 3-14). 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.

To install your customized Oracle SellingPoint application:

1. Have an Oracle Configurator Database populated with what you need for your Oracle SellingPoint application. See Section 2.3, "Establish an Oracle Configurator Database", on page 2-4 for information about establishing an Oracle Configurator Database.
2. Run `SpRuntime.exe` from the OSPC CD-ROM to install only the Oracle SellingPoint application on a deployment machine.

   The OSPC 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 Database, either local or networked. See Section 3.5.3, "Create DSNs and DBOwners (Oracle8 and Oracle8i Lite)", on page 3-7 for information about establishing data connectivity.
5. Edit the spx.ini file to point to the appropriate DSN. See Section 3.5.4, "Set Parameters in spx.ini", on page 3-9 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 Database as the DBOwner or a user listed in the CZ_END_USERS table, you have access to all the OSPC files needed.

5.2.2 Networked (Client) Applications

To install your customized Oracle SellingPoint application on a networked client machine, your DSN must point to the networked computer in addition to the requirements for any install program mentioned above.

5.2.3 Mobile Applications

In addition to the requirements for any install program mentioned above, you must complete the following when you install your customized Oracle SellingPoint application on a mobile laptop machine:

- Install your local replica database in `%ORACLE_HOME%\OSP\shared\Database` or establish a replica of the Oracle Configurator Database from the Oracle SellingPoint application, i.e., the Oracle Configurator Mobile Database. See Section 7.5, "Create the Replica Database" for more information.
- Set up a DSN to point to the local replica database

5.2.4 Oracle Configurator Window

In order to use the Oracle Configurator window for testing or in a custom application, you must first install the UI Servlet on your Internet server. See the *Oracle Configurator Administration Guide* for additional information about requirements for using the Oracle Configurator Java Applet window. See the *Oracle Configurator Internet Edition Developer's Guide* for additional information about requirements for using the Oracle Configurator DHTML window.

5.3 End User Administration

Setting up a deployed Oracle SellingPoint application for use includes making it possible for specific end users to log into the application and have access to specific configuration models, user interfaces, and account information.

5.3.1 Add End Users

When first creating an Oracle Configurator Database schema for your Oracle SellingPoint application, the DBOwner (<spx>) is the default user for both Oracle

SellingPoint Developer and the end user application. The <spx> user is inserted into the CZ_END_USERS table in the Oracle Configurator Database. When directly importing data from the Oracle Applications database, Oracle Applications users are inserted in the CZ_END_USERS table in the Oracle Configurator Database (see Section 2.3.2, "Create OSPC Users and Responsibilities" on page 2-7).

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 Database schema serving that Oracle SellingPoint application.

   Example, log in as <spx> with the DSN that points to the Oracle Configurator Database 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 4.2.2
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 Database serving the current Oracle SellingPoint application.

5.3.2 Assign End Users to Projects

In Oracle SellingPoint Developer, the work space for building a configuration model is a Project. A Project in 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 Database schema serving that Oracle SellingPoint application.

   Example, log in as <spx> with the DSN that points to the Oracle Configurator Database 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.

5.3.3 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 Database schema serving that Oracle SellingPoint application.

   Example, log in as <spx> with the DSN that points to the Oracle Configurator Database 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 5.3.2, "Assign End Users to Projects").

5.4 Proposal

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. OSPC users can easily create new templates. There are however some administrative tasks associated with providing templates to the Proposal Wizard.

5.4.1 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.

5.4.2 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.

For more information about Proposal, see the *Oracle SellingPoint Application Help* provided with the default Oracle SellingPoint application.

6

Data Export

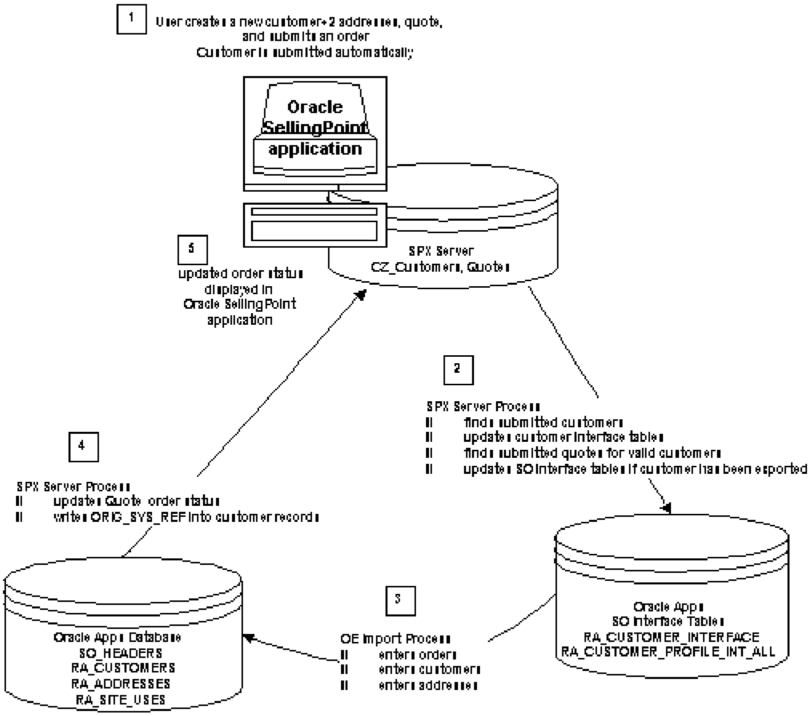
Oracle SellingPoint applications use a standard schema for configuration data referred to as the Oracle Configurator Database. The Oracle Configurator Database is used to store customer information and all information relative to the configuration model — product data, model structure, configuration rules, and user interface layouts. Customer and product configuration quote data can be exported from the Oracle Configurator Database to the Oracle Applications database for use by Oracle Order Entry and Oracle Receivables.

6.1 Overview of Data Export

Data export involves preparing the OSPC and the Oracle Applications environments to support data and status transmission. The Oracle SellingPoint application manipulates and modifies imported BOM data into valid configurations. The Oracle SellingPoint application end user creates a product configuration quote for a specified customer and submits the quote as an order. The quote and customer data are exported to the Oracle Applications database, so that the orders created in the Oracle SellingPoint application are available to Oracle Order Entry. When an order is booked in Oracle Order Entry, the order status is transferred back through the export tables of the Oracle Configurator Database to the Oracle SellingPoint application.

New Customer data can also be exported to the Oracle Application Database.

Figure 6–1 Overview of Data Export



For additional details about how export is handled in a replication environment, see Table 6–3, "Overview of Data Export with Replication".

6.2 Requirements for Export

The Oracle Configurator Database must include export tables (see Section 2.3.3.2, "Create the Oracle Configurator Database for Integration" on page 2-11).

You must have access to the export scripts in the DBAdmin folder on the Oracle SellingPoint Configurator CD-ROM.

The three main processes involved in export are executed successfully, either manually or automatically (see Section 6.5, "Export Process"):

- order and customer data submitted by Oracle SellingPoint application for export from Oracle Configurator Database Quotes and Configuration subschema to Oracle Applications database interface tables

- order and customer data import from Oracle Applications database interface tables to Oracle Order Entry and Oracle Receivables
- order status and customer data update from Oracle Applications database to Oracle Configurator Database for display in Oracle SellingPoint application

6.3 Data to be Exported

Exported configuration and quote data used in Oracle Order Entry refer to the data that are imported directly from an Oracle Configurator Database. New customer data created by the Oracle SellingPoint application are exported to the Oracle Applications database for use by Oracle Receivables.

For information about what data are imported directly or generically, see Section 4.1.2, "Data to Be Imported Through Direct Import" on page 4-5 and Section 4.1.3, "Data to Be Imported Through Generic Import", on page 4-7.

All exported data (quote and customer) are first exported to Oracle Applications interface tables from the Oracle Configurator Database. 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 exported to Oracle Applications (Order Entry and Receivables) are:

- Order Headers
- Order Lines
- BOM (same structure as was imported)
- BOM (same properties as were imported, with possible modifications in OSPD)
- BOM (new properties added in OSPD)
- Customer information (imported customer records must remain unchanged in orders exported back to Oracle Applications)
- New Customer information
- Prices of BOM items (same as were imported if Oracle Pricing is used - not visible in the Oracle SellingPoint application)
- Organization IDs associated with the BOM.

Appendix C, "Data Export Requirements" presents a mapping of the source and destination for data exported from the Oracle Configurator Database tables to the Oracle Applications database so that orders and new customer information submitted by the Oracle SellingPoint application are complete in Oracle Applications.

6.4 Export Setup

Oracle SellingPoint application quotes submitted as orders pass enough parameters to Order Entry for entering orders, but *not* for booking orders. Parameters that should be defaulted in a Standard Value Rule Set in Oracle Applications for easy downstream booking of orders submitted from Oracle SellingPoint are:

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

In the Oracle SellingPoint application, 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 Entry. This end user must have had GRANT_SELECT_FOR.sql run on it to ensure write permission 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.

Export setup consists of:

- completing a successful import on the Oracle Configurator Database. See Section 4.3, "Run Direct Import" on page 4-19 or Section 4.4, "Run Generic Import" on page 4-21.
- setting quote and customer parameters in the CZ_DB_SETTINGS table in the Oracle Configurator Database. See Section 6.4.1, "DB Settings for Export" on page 6-5.
- ensuring that currency is set to only one currency throughout the quote, such as US currency (dollar) in the CZ_PRICE_GROUPS table (this is typically done during the import process)
- ensure that there are data to export. Oracle SellingPoint application end users must have submitted quotes or customer information to be exported to Oracle Applications (order_request_flag=S in CZ_QUOTE_HDRS and/or exported_flag=S in CZ_CUSTOMERS)

- optionally creating and scheduling the export and update processes as database jobs in the Oracle Configurator Database connected to the Oracle Applications database via link (see Section 6.4.2, "Export via Link")
- optionally registering and scheduling the export and update processes as concurrent programs in the instance where both the Oracle Configurator Database and Oracle Applications database are running (see Section 6.4.3, "Export Within one Instance")

6.4.1 DB Settings for Export

You can adjust the way the export is executed with settings in the CZ_DB_SETTINGS table in the Oracle Configurator Database. For example, you can export just order information, order and customer information, or both. 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 2.4, "Set DB Settings" on page 2-16.

6.4.2 Export via Link

If the Oracle Configurator Database and Oracle Applications database are not in the same instance (<spxsid>≠<appssid>), Oracle SellingPoint Configurator provides SQL*Plus scripts for exporting data to the Oracle Applications database and returning status information to the Oracle SellingPoint application via link. The export scripts are in the DBAdmin folder on the Oracle SellingPoint Configurator CD-ROM.

To test or run export manually via a link you can run the scripts in SQL*Plus (see Section 6.6, "Export Manually"). To run the scripts as a scheduled job, use the DATABASE_JOBS package to create and schedule a job for both the export process (CZ_EXPORT.submit_all) and the update process (CZ_EXPORT.order_status_update_process).

It is not possible to schedule the export and update processes via a link as a concurrent programs. The import process within the Oracle Applications database already exists as a concurrent program (see Section 6.4.4, "Setting up the Oracle Applications Database").

6.4.3 Export Within one Instance

If the Oracle Configurator Database and Oracle Applications database are in the same instance (<spxsid>=<appssid>), Oracle SellingPoint Configurator provides SQL*Plus scripts for exporting data to the Oracle Applications database and returning status information to the Oracle SellingPoint application.

To test or run export manually via a link you can run the scripts in SQL*Plus (see Section 6.6, "Export Manually").

To run the scripts as scheduled jobs, register the SQL*Plus scripts for exporting data to the Oracle Applications database and returning status information to the Oracle SellingPoint application as concurrent programs and schedule the programs to run as needed. The import process within the Oracle Applications database already exists as a concurrent program (see Section 6.4.4, "Setting up the Oracle Applications Database").

It is not possible to create and schedule the scripts using the DATABASE_JOBS package if the Oracle Configurator Database and Oracle Applications database are in the same instance.

6.4.4 Setting up the Oracle Applications Database

Whether the Oracle Configurator Database and Oracle Applications database are in the same instance or not, the order and customer export process must include transferring the data from the populated Oracle Applications database interface tables to the Order Entry and Oracle Receivables tables. You can run the order and customer import within the Oracle Applications database as concurrent programs manually (see Section 6.6, "Export Manually" Step 7 and Section 6.7, "Export Customer Only" Step 9). The order import concurrent program includes transferring customer data from the Oracle Applications database interface tables to Oracle Receivables.

To run the order and customer import concurrent program (Requests: Import Orders: Submit Requests) automatically, set it up as a scheduled concurrent program.

6.5 Export Process

Once Oracle SellingPoint application end users have created quotes and submitted them as orders, the data must be exported to Oracle Order Entry. Order export may occur for a single order or for many orders at once. For information on exporting a single order at a time, see Section 6.6, "Export Manually".

If the order includes a new customer, that data will automatically be exported with the order and transferred to Oracle Receivables with the Requests: Import Orders concurrent process. For information on exporting only new customer data, see Section 6.7, "Export Customer Only".

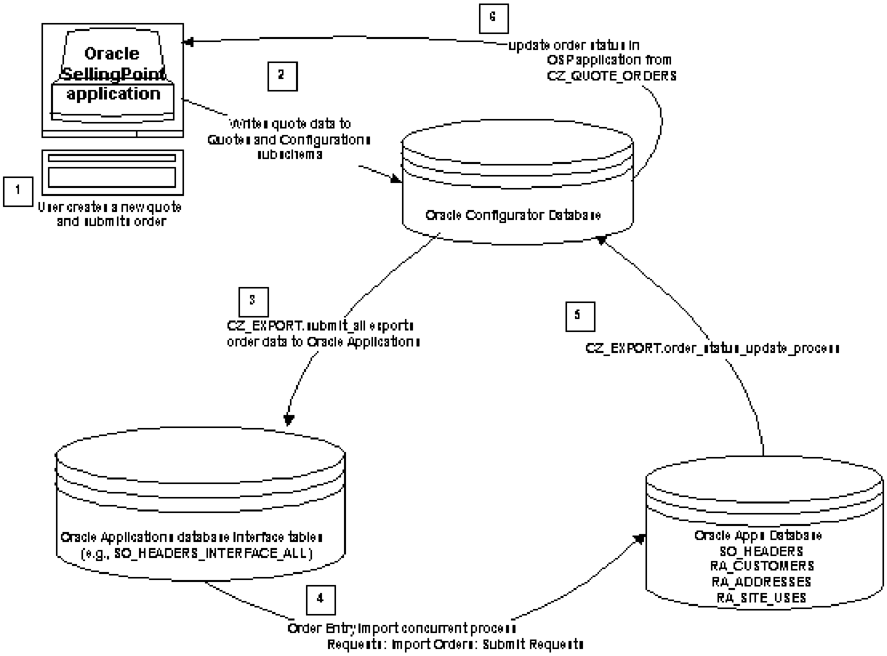
When you submit an order in the Oracle SellingPoint application, the data are stored in the Quotes and Configurations subschema of the Oracle Configurator Database, including a record for the quote in the CZ_QUOTE_HDRS table. The script CZ_EXPORT.submit_all exports order and customer data from the Quotes and Configurations subschema of the Oracle Configurator Database to the Oracle Applications database interface tables (SO_HEADERS_INTERFACE_ALL, etc.). See Appendix C, Section C.2, "Export Data Source and Destination" for the exact field to field mapping. See Section 6.4, "Export Setup" for information about creating database programs to run the export script automatically.

The Order Entry concurrent program Requests: Import Orders: Submit Requests on the Order Source Oracle SellingPoint application imports the order and customer data from the Oracle Applications database interface tables into Oracle Order Entry and Oracle Receivables. This concurrent program can be set up to run automatically on a schedule. For details about this concurrent program, see Section 6.6, "Export Manually" Step 7. For details about a concurrent program that imports only customer data, see Section 6.7, "Export Customer Only" Step 9.

Once an order has been processed, its status must be updated in the Oracle SellingPoint application. The script CZ_EXPORT.order_status_update_process 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 Database. The script updates status in the status field for an order in the Oracle SellingPoint application, and updates the customer information. See Section 6.4, "Export Setup" for information about creating database programs to run the update script automatically.

If an existing quote (i.e., Qv1) in the Oracle SellingPoint application is revised (Qv2) and resubmitted, Oracle Configurator Database export tables pass a new order (Qv2) to the Oracle Applications database interface table SO_HEADERS_INTERFACE_ALL flagged for ’insert’, and mark the old order (Qv1) ’deleted’. Import from the interface tables to Order Entry (i.e., SO_HEADERS) updates the status of the original order (Qv1) to ’cancelled’. When the status update transfers the order data back to the Oracle Configurator Database, the first order’s status (Qv1) in the Oracle SellingPoint application is ’cancelled’ and the second order’s status (Qv2) is active (e.g., ’entered’). The changed status of the old order will not appear in the Oracle SellingPoint application unless every step in the export process has succeeded. See Section 6.9, "Verify Export" for additional information on tracking an order through export.

Figure 6–2 Overview of Export Processes


6
update order status in
OSP application from
CZ_QUOTE_ORDERS
Oracle
SellingPoint
application
2
Writes quote data to
Quotes and Configurations
subschema
1
User creates a new quote
and submits order
Oracle Configurator Database
3
CZ_EXPORT.submit_all exports
order data to Oracle Applications
5
CZ_EXPORT.order_status_update_process
Oracle Applications database interface tables
(e.g., SO_HEADERS_INTERFACE_ALL)
Oracle Apps Database
SO_HEADERS
RA_CUSTOMERS
RA_ADDRESSES
RA_SITE_USES
4
Order Entry Import concurrent process
Requests: Import Orders: Submit Requests

6.6 Export Manually

You might export orders manually for testing. These instructions are valid whether the Oracle Configurator Database and Oracle Applications database are in the same instance or not.

1. You have set up your Oracle Configurator Database to prepare for exporting orders from the Oracle SellingPoint application (see Section 6.4, "Export Setup").
2. Create a quote in the Oracle SellingPoint application and submit it as an order.

Make a notation of the quote number (<quotenumber>) that is assigned to the new quote in the Oracle SellingPoint application if you want to export only that one order.

3. Start SQL*Plus, and connect to the Oracle Configurator Database schema (e.g., <spx>) from which you will export data to Oracle Applications. See Section 1.7.1, "Connect to a Database Instance" for details.
4. To export one quote at a time, run the CZ_EXPORT.submit_for_quote procedure.

   Example:

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

   where <quotenumber> is the number you noted in step , above.

   Leave SQL*Plus in <OSPC-scripts> directory connected to <spxsid> running.

5. To export all quotes in the Quotes and Configurations subschema of the Oracle Configurator Database, run the CZ_EXPORT.submit_all procedure.

   Example:

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

   Leave SQL*Plus in <OSPC-scripts> directory connected to <spxsid> running.

6. Start Oracle Applications.
7. Import into Oracle Order Entry:
   a. Select Order Entry, <your global organization>.
   b. Click OK.
   c. From the right-hand side of the Navigator window, select Requests: Import Orders.
   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 SellingPoint application).

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 exported to Oracle Order Entry.

j. Submit another request? Y or N.

8. To update the status of the order in the Oracle Configurator Database, 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 SellingPoint application, select Order Status from the left pane. The updated order status is displayed for each order.

6.7 Export Customer Only

You might export customer data manually for testing. These instructions are valid whether the Oracle Configurator Database and Oracle Applications database are in the same instance or not.

Customer export may occur only for new customers created in the Oracle SellingPoint application. Customer information imported to the Oracle Configurator Database from Oracle Applications is used for building configurations and quotes, but cannot be modified in the Oracle SellingPoint application and exported back to Oracle Applications.

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

1. You have set up your Oracle Configurator Database to prepare for exporting from the Oracle SellingPoint application (see Section 6.4, "Export Setup").
2. Create a customer in the Oracle SellingPoint application and submit it while logged on as <apps> in the Oracle SellingPoint application.

Make a note of the customer identification number (<customer_id>) that is assigned to the new customer you created in the Oracle SellingPoint application, if you plan to export just that one customer manually to Oracle Applications.

3. Start SQL*Plus, and connect to the Oracle Configurator Database schema (e.g., <spx>) from which you will export data to Oracle Applications. See Section 1.7.1, "Connect to a Database Instance" 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 export to Oracle Receivables.

5. To display possible errors when the export script runs, type the following command.

   ```
   SQL> set serveroutput on;
   ```

6. To export 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 <OSPC-scripts> directory connected to <spxsid> 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 export all customers, output listed shows all customers, but only customers that meet the selection criteria of the export procedure are exported. Likewise, customers that failed to export previously will also be exported if they meet the selection criteria of the export procedure.

Leave SQL*Plus in <OSPC-scripts> directory connected to <spxsid> 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 exported 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 Database, return to the SQL*Plus window and run the CZ_EXPORT.customer_update_process procedure.

    Example:

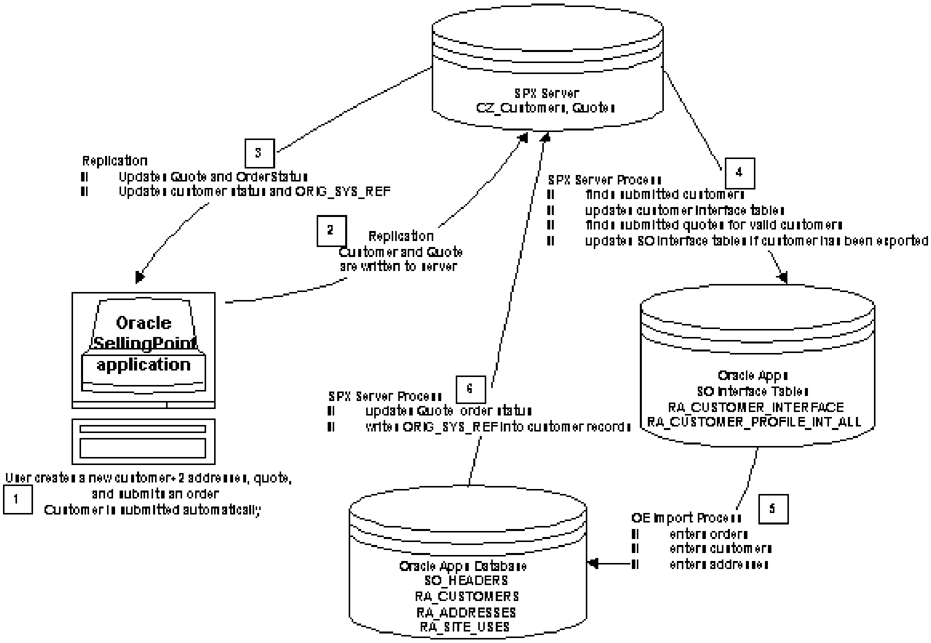
```
SQL> exec cz_export.customer_update_process;
SQL> commit;
```

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

6.8 Replicating Orders From a Mobile Deployment

In a mobile environment, end users have to replicate their order data from the Oracle Configurator Mobile Database on the laptop to the Oracle Configurator Database on the server (see Section 7.6.1, "Synchronize", on page 7-12). The OSPC administrator then executes the export from the Oracle Configurator Database to the Oracle Applications database. To download changes in order status into their mobile deployment of Oracle SellingPoint application, end users have to synchronize with the Oracle Configurator Database on the server after the administrator has completed the export process between the Oracle Configurator Database and the Oracle Applications database. End users synchronize by selecting Tools --> Replicate from the Oracle SellingPoint application toolbar.

Figure 6–3 Overview of Data Export with Replication


SPX Server
CZ_Customers, Quotes
3
Replication
Updates Quote and OrderStatus
Updates customer status and ORIG_SYS_REF
4
SPX Server Process
finds submitted customers
updates customer interface tables
finds submitted quotes for valid customers
updates SO interface tables if customer has been exported
2
Replication
Customer and Quote
are written to server
Oracle
SellingPoint
application
Oracle Apps
SO Interface Tables
RA_CUSTOMER_INTERFACE
RA_CUSTOMER_PROFILE_INT_ALL
6
SPX Server Process
updates Quote order status
writes ORIG_SYS_REF into customer records
User creates a new customer+2 addresses, quote,
and submits an order
Customer is submitted automatically
1
Oracle Apps Database
SO_HEADERS
RA_CUSTOMERS
RA_ADDRESSES
RA_SITE_USES
5
OE Import Process
enters orders
enters customers
enters addresses

6.9 Verify Export

Verify a successful export by viewing the order that was imported from the Oracle Configurator Database in Oracle Order Entry.

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

If replication is to be enabled, Order Entry needs to be tested again from an Oracle SellingPoint application mobile client by means of replication.

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

1. Oracle SellingPoint application order or customer submission process runs. This process inserts data into the Oracle Configurator Database export tables.

If replication is being used, orders from the Oracle Configurator Mobile Database on the laptop must be replicated to the Oracle Configurator Database on the server (see Section 6.8, "Replicating Orders From a Mobile Deployment").

2. The order or customer export process must run successfully. The export process populates the Oracle Applications database interface tables with the order and customer data for extraction to source tables (see Section 6.5, "Export Process").
3. The order or customer update process must run successfully. This propagates changes to the order or customers made in Order Entry and Receivables back to the Oracle Configurator Database so that the status of the order and customer information is updated in the Oracle SellingPoint application.

For information about resubmitting a new version of an already submitted quote, see Section 6.5, "Export Process".

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

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

7 Mobile Deployment Administration

In a mobile deployment, the Oracle SellingPoint application runs on a laptop computer, using a local replica of the Oracle Configurator Database 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 Database, 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 Database and the Oracle Configurator Mobile Database.

Oracle does not support a mobile development environment running Oracle SellingPoint Developer. However, it is possible to develop demonstrations and prototypes using Oracle SellingPoint Configurator in a mobile environment.

The standard supported method of mobile deployment is network-based using SQL*Net over a LAN. The Oracle SellingPoint Configurator also provides minimal support for file-based replication. See Section 7.7, "File-Based Replication" on page 7-13 for information about implementing file-based replication.

See Figure 1–3, "Overview of Deployments Integrated with Oracle Applications" on page 1-8 for an illustration of the mobile deployment architecture.

7.1 Replication Method in Oracle SellingPoint Configurator

The Oracle SellingPoint 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 SellingPoint 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 SellingPoint 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.

7.2 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 Database 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 Database. See Table 2–2, "CZ_DB_SETTINGS". This is done automatically when you run rep_setup.sql.
- Oracle8*i* Lite is running on the laptop.
- Oracle Client 8.0 is running on the client laptop machine.
- The laptop is configured to connect to the Oracle Configurator Database on the server machine (Oracle Net8 Easy Config). See Section 3.5.2, "Enable Client for Database Connectivity" on page 3-6.
- 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 Database 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 Database 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.

7.3 Prepare the Server for Database Replication

Oracle provides several scripts to aid you in preparing the server system to replicate the Oracle Configurator Database. These scripts are available on the OSPC CD-ROM in the DBAdmin folder. Run the scripts from the CD or the directory into which you have copied them (<OSPC-scripts>) using SQL*Plus, not SQL Worksheet. The state you achieve by running these scripts is summarized in Section 7.3.4, "Database Replication: Server Checklist"

7.3.1 Prerequisites for Running the Replication Setup Scripts

Certain 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 replication option on. The advance replication support should be installed before the SellingPoint Configurator objects are installed. All other replication setup can be done independently of creating the Oracle Configurator Database.
- Oracle8*i* Enterprise Edition must be running on the server.
- You must create a production installation of Oracle Configurator Database on the server. See Section 2.3, "Establish an Oracle Configurator Database" on page 2-4. Create a DBOwner in the Oracle Configurator Database instance that has access to all tables and data required for the deployed Oracle SellingPoint application.
- The Oracle Configurator Database must be accessible from the machine where you are running the scripts.

7.3.2 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 mobile client and server.

See the following sections for instructions on running these scripts.

7.3.3 Run the OSPC Replication Setup Scripts

The OSPC 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 Database to support replication as follows:

1. Connect as <dba> to the database instance <spxsid> where you are setting up the production Oracle Configurator Database to support replication (see Section 1.7.1, "Connect to a Database Instance").
2. Run the scripts in <OSPC-scripts> directory using SQL*Plus, not SQL Worksheet (see Section 1.7.2, "Run SQL*Plus in the <OSPC-scripts> Directory").
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 Database schema (<spx>). 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 any schema created by the SP_BOOTSTRAP.sql script. 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. <spx>, the Oracle SellingPoint Configurator schema (user) name

g. <repgroup>, the name of the replication group in the Oracle Configurator Database.

The rep_setup.sql script prepares the Oracle Configurator Database <spx> for replication. The rep_setup.sql script changes several settings in the CZ_DB_SETTINGS to ‘YES” (see Section 7.3.4, "Database Replication: Server Checklist").

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 mobile client and 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 Database 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.

7.3.4 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 Database is accessible from the server.
- The Replication setting in the CZ_DB_SETTINGS table is enabled. See Table 2–2, "CZ_DB_SETTINGS". 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 2.4, "Set DB Settings" for more information about these settings.

7.4 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 Oracle Client 8.0 on the client laptop.
3. Use Oracle Net8 Easy Config to configure the laptop to connect to the Oracle Configurator Database on the server machine. See Section 3.5.2, "Enable Client for Database Connectivity" on page 3-6.
4. Set up the client machine with an ODBC DSN for <spxsid>, as described in Section 3.5.3, "Create DSNs and DBOwners (Oracle8 and Oracle8i Lite)" on page 3-7.
5. Install your Oracle SellingPoint application.

   See the *Oracle SellingPoint Configurator ReadMe* and *Oracle SellingPoint Configurator Release Notes* (on the Oracle SellingPoint Configurator 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 SellingPoint Developer) for the Project to which the Oracle SellingPoint application end users need access.

   If there is a parameter `WriteLogicRecords` in the `spx.ini` file on the client machine, be sure it is set to `true` when executing the Generate Active Model command.

Example:

```
[Merlin]
DBOwner=<spx>
WriteLogicRecords=true
```

This parameter is not available in the initialization file installed with OSPC Release 4.2.2.

7. You must also verify that the UI has been generated (Generate Active UI command in Oracle SellingPoint Developer) for the Project to which the Oracle SellingPoint application end users need access.

7.5 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., <spx>). Use Assign Customers on the Tools>Administration menu in the Oracle SellingPoint application to assign customers and projects to an appropriate end user. See Section 5.3, "End User Administration" for more information.
2. On your client machine, connect to the server machine <spxdbhost> using either a network or dial-up connection. Start your Oracle SellingPoint application on the client machine. Log in as yourself on <spx>.
3. Go to the Tools > Administration menu and select Initialize Remote DB. This creates an Oracle8*i* Lite replica of <spx> on your client machine, creates an ODBC DSN for <spx>_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 Database. You must be running Oracle Client 8.0 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 <spx>_replica as follows.

- The Data Source Name for <spx> on <spxsid) is added to the `[DSN]` section.

Example:

```
[DSN]
SpxLite
HomeTheaterDemo
TutorialLite
<spxsid>
```

- Parameters for the replica and the schema, in this example vis11_replica and vis11, are added to the `[MDADSNS]` section.

  Example:

  ```
  [MDADSNS]
  <spx>_replica
  <spxsid>
  SpxLite
  HomeTheaterDemo
  TutorialLite
  ```

- The schema section is added.

  ```
  [spxsid]
  DBowner = spx
  ```

- The replica section is added.

  ```
  [<spx>_replica]
  DBOwner=<spx>
  master_name=<spx>
  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
  ```

At this point, creation of the replica database is complete, and the laptop is ready for use by a mobile end user. See Section 3.5.4.7, "[<DSN>_replica]" on page 3-16 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 (<spx>), as described in Section 7.6, "Data Synchronization" on page 7-12.

7.5.1 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 Database, 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 Database on the server.

To 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 7.5, "Create the Replica Database", on page 7-9.
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\`. (You should also find the database for SPXLite.odb in this directory.)

   b. Make a backup copy of your Oracle SellingPoint 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 SellingPoint 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 3.5.2, "Enable Client for Database Connectivity", on page 3-6 and Section 3.5.3, "Create DSNs and DBOwners (Oracle8 and Oracle8i Lite)", on page 3-7 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 the end users.

7.6 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 Database 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.

7.6.1 Synchronize

Follow these steps to synchronize data.

1. On your client machine where you created the Oracle Configurator Mobile Database (<spx>_replica), connect to the server machine <spxdbhost> 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 <spx>_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 <spx>_replica to the server, and downloads new data on the server to <spx>_replica.

This procedure does not synchronize data in a production Oracle Configurator Database with an Oracle Applications database. You must use the import and export mechanisms described in Chapter 4, "Data Import" and Chapter 6, "Data Export" to keep your server Oracle Configurator Database 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 Import and Export 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. You can change this local filename 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.

7.6.2 Test Synchronization

7.6.2.1 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.

7.6.2.2 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 order submission in Order Entry. See Section 6.9, "Verify Export" for more information.

7.7 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 Database. 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 to 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 Database 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 Database 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.

7.7.1 Setting Up File-based Replication

To set up file-based replication:

1. Be sure the requirements for mobile deployment, described in Section 7.2, "Requirements for Mobile Deployment", have been met.
2. Prepare the server machine for data replication as described in Section 7.3, "Prepare the Server for Database Replication" except change the CZ_DB_SETTINGS for file-based replication as described in Table 2–2, "CZ_DB_SETTINGS" 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 Database 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 7.4, "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:

   ```
   [spx_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 7.5, "Create the Replica Database".

10. Test synchronization. See Section 7.6.2, "Test Synchronization" 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.

A

OSPC SQL*Plus Scripts and Procedures

This Appendix is organized as follows:

A.1 The DBAdmin Folder on the OSPC CD-ROM

Your Oracle SellingPoint Configurator CD-ROM contains a folder, DBAdmin, containing the OSPC SQL*Plus scripts to perform basic administrative procedures, and the packages that support those scripts. The folder is organized as follows:

Table A–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 Database. Oracle Applications-unaware.
DBAdmin/Server/	Scripts for creating/upgrading the server version of the Oracle Configurator Database. Oracle Applications-unaware.

A.2 Scripts

Oracle SellingPoint Configurator provides SQL*Plus scripts to perform basic administrative procedures. All files are included in the DBAdmin folder on the OSPC CD-ROM.

Note: The OSPC SQL*PLUS scripts provided in the DBAdmin folder on the OSPC CD-ROM 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 <OSPC-scripts> directory, or start SQL*Plus, go to File --> Open, navigate to the directory or folder where the OSPC scripts are located (<OSPC-scripts>), and click Cancel.

Caution: Run OSPC SQL*Plus scripts from SQL*Plus. **Do not run** OSPC SQL*Plus scripts from SQL Worksheet.

All arguments for the scripts are listed and described below in Appendix A.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 A–2 OSPC 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 Database 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 Database. 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 A–2 OSPC SQL*Plus Scripts*

Script Name	What it does
CTRA_ADMIN_B.sql	This file is delivered as part of Oracle SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 Database 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 Database.

*Table A–2 OSPC SQL*Plus Scripts*

Script Name	What it does
CZ_CF_API_B.sql	This file is delivered as part of Oracle SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script. 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 Database.
CZ_CF_API_S.sql	This file is delivered as part of Oracle SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script

Table A–2 OSPC 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script

*Table A–2 OSPC SQL*Plus Scripts*

Script Name	What it does
CZ_LIST_PRICE_S.sql	This file is delivered as part of Oracle SellingPoint 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 Database 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script

*Table A–2 OSPC SQL*Plus Scripts*

Script Name	What it does
CZ_PR_MGR.sql	This file is delivered as part of Oracle SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script
CZ_REBUILD_INDEXES.sql	
CZ_SEQUENCES.sql	(Lite version) This file is delivered as part of Oracle SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script

Table A–2 OSPC SQL*Plus Scripts

Script Name	What it does
CZ_SEQUENCES.sql	(Server version) This file is delivered as part of Oracle SellingPoint 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 Database schema on an Oracle8 server. This script does not require parameters.
cz_standalone.sql	(Lite version) This file is delivered as part of Oracle SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script

*Table A–2 OSPC 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 A.2.8, "Using EndUsers.sql"
export_conc_prog.sql	Registers the Oracle SellingPoint Configurator Order Export software for periodic execution by Concurrent Manager

*Table A–2 OSPC SQL*Plus Scripts*

Script Name	What it does
FND_STATS.sql	This file is delivered as part of Oracle SellingPoint 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 Database, creates the OSPC tables and import schema and all Oracle Applications integration, and performs initial import See also Appendix A.2.3, "Using GO_IMPORT.sql"
GO_IMPORT_ONLY.sql	PL/SQL script installs Oracle Configurator Database, import tables, and import packages for non-Oracle Applications installations. See also Appendix A.2.5, "Using GO_IMPORT_ONLY.sql"
GRANT_SELECT_FOR.sql	PL/SQL script for Applications integration. Grants access on Applications tables to SellingPoint owner. See also Appendix A.2.7, "Using GRANT_SELECT_FOR.sql"
GRANT_TO_ROLE.sql	PL/SQL script grants access privileges to the 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 SellingPoint 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 SellingPoint 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 SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script

*Table A–2 OSPC SQL*Plus Scripts*

Script Name	What it does
IMP_AC_MAIN_S.sql	This file is delivered as part of Oracle SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script
IMP_AC_XFR_B.sql	This file is delivered as part of Oracle SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script

*Table A–2 OSPC SQL*Plus Scripts*

Script Name	What it does
IMP_EXTRACT_S.sql	This file is delivered as part of Oracle SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script
IMP_IM_KRS_B.sql	This file is delivered as part of Oracle SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script

*Table A–2 OSPC SQL*Plus Scripts*

Script Name	What it does
IMP_PR_KRS_S.sql	This file is delivered as part of Oracle SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script
IMP_PR_MAIN_B.sql	This file is delivered as part of Oracle SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script

*Table A–2 OSPC SQL*Plus Scripts*

Script Name	What it does
IMP_PS_NODE_S.sql	This file is delivered as part of Oracle SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script
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 for import (granting all necessary SELECT privileges first), 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 (granting all necessary SELECT privileges first), 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 A.2.4, "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 A.2.6, "Using LoadAllBills.sql"
ospc_messages.sql	(Lite version) This file is delivered as part of Oracle SellingPoint 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 SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script

*Table A–2 OSPC SQL*Plus Scripts*

Script Name	What it does
order_status_conc_prog.sql	Registers the Oracle SellingPoint application order-status-retrieval software for periodic execution by Concurrent Manager
register_resolvers.sql	Registers custom conflict resolution functions
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 each replication end user
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 Database 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
SP_BOOTSTRAP.sql	DBA script to create owner for Oracle Configurator Database and Import databases, and to create the end-user role for granting access to the Oracle Configurator Database. See also Appendix A.2.2, "Using SP_BOOTSTRAP.sql"

*Table A–2 OSPC SQL*Plus Scripts*

Script Name	What it does
TimeStampTriggers.class	This file is delivered as part of Oracle SellingPoint 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 SellingPoint 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 (14b)
UPGRADE_DEFAULTS.sql	(Lite version) This file is delivered as part of Oracle SellingPoint 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 SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script
UPGRADE_DROP_OBJECTS.sql	This file is delivered as part of Oracle SellingPoint 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 SellingPoint 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 SellingPoint 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 (14b)

*Table A–2 OSPC SQL*Plus Scripts*

Script Name	What it does
UPGRADE_SEQUENCES.sql	(Lite version) This file is delivered as part of Oracle SellingPoint 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 SellingPoint 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 Database (Oracle8*i* Enterprise Edition) from 4.1.1 (12f) to 4.2 (14b)
UPGRADE_TABLES.sql	(Lite version) This file is delivered as part of Oracle SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script
UPGRADE_TABLES.sql	(Server version) This file is delivered as part of Oracle SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script

*Table A–2 OSPC SQL*Plus Scripts*

Script Name	What it does
UpgradeExpressions.java	This file is delivered as part of Oracle SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script
UpgradeTables.java	This file is delivered as part of Oracle SellingPoint 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 SellingPoint 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 SellingPoint 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 SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script

*Table A–2 OSPC SQL*Plus Scripts*

Script Name	What it does
user4.sql	This file is delivered as part of Oracle SellingPoint 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 SellingPoint 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 SellingPoint Configurator internal database installation and configuration. It is not intended, documented, or supported for customer use as a standalone script

A.2.1 Script Arguments

This book refers to the Oracle Configurator Database schema and its DBOwner as <spx>. This manual refers to the instance hosting the Oracle Configurator Database that contains <spx> as <spxsid>. This manual refers to the machine where <spxsid> is established as <spxdbhost>.

Wherever scripts take arguments, the following conventions are used:

*Table A–3 Argument Tokens for OSPC SQL*Plus Scripts*

Token	Description
<apppass>	Password for <apps>
<apps>	Owner of the Oracle Applications database, usually ’apps’
<appsdbhost>	Hostname for the system on which the Oracle Applications Oracle8 instance is running.
<appssid>	Instance name for the Oracle8 instance on which the Oracle Applications database is installed
<dba>	Username of DBA with DBA privileges.
<dbapass>	Password for <dba>.

Table A–3 Argument Tokens for OSPC SQL*Plus Scripts

Token	Description
<DBAUser>	Username of the SellingPoint administrator who has DBA privileges.
<DBAUserpass>	Password for <DBAUser>
<defaultspace>	Default permanent tablespace to be used by <spx>, <spxdev>, 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 SellingPoint Import/Integration 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>
<imppass>	Password for <imp>
<imptempspace	Temporary tablespace to be used by <imp> for temporary tables
<indxspace>	Tablespace to be used by <spx>, <spxdev>, and end users for indexes
<oe>	Owner of Oracle Applications order-entry tables
<oepass>	Password for <oe>
<OSPC-scripts>	Directory where SQL*Plus scripts used to set up the Oracle Configurator Database are located
<port>	Listener port number (1521 in most locations)
<repadmin>	Name of the Replication Administration user creating the replication objects in the Oracle Configurator Database. Has access to the packages dbms_repcat, etc.
<repadminpass>	Password for <repadmin>
<repgroup>	Name of the Replication Group in the Oracle Configurator Database
<spx>	Owner of the Oracle Configurator Database schema.

*Table A–3 Argument Tokens for OSPC SQL*Plus Scripts*

Token	Description
<spxpass>	Password for <spx>
<SPX_USER>	Name of database role for Oracle SellingPoint application end users for grants. Default value in CZ_DB_SETTINGS table in Oracle Configurator Database is 'SPX_USER'.
<spxdbhost>	Hostname for the system on which the Oracle Configurator Database Oracle8*i* Enterprise Edition instance is running.
<spxdev>	Oracle SellingPoint Developer user
<spxdevpass>	Password for <spxdev>
<spxsid>	Name for the Oracle8*i* Enterprise Edition instance on which the Oracle Configurator Database is installed. If on the same installation of Oracle8*i* Enterprise Edition as where the Oracle Applications database is installed, <spxsid>=<appssid>.
<tempspace>	Temporary tablespace to be used by <spx>, <spxdev>, 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.
<TNSalias>	The TNS alias is the service name you enter in Net8 Easy Config (vis11) that represents the full service name to which you connect for database access.

A.2.2 Using SP_BOOTSTRAP.sql

A.2.2.1 Parameters

The parameters for SP_BOOTSTRAP.sql are specified as arguments. See Table A–3, "Argument Tokens for OSPC SQL*Plus Scripts" for descriptions of the arguments. The parameters are listed in the order in which they must be specified when executing SP_BOOTSTRAP.sql. See also Section 2.3.1, "Create Oracle Configurator DBOwner" on page 2-5.

- &1 - New user to own the Oracle Configurator Database <spx> (must be a DBOwner with administrative privileges)
- &2 - Password for new user <spxpass>
- &3 - New Developer user <spxdev> (user without administrative privileges)

- &4 - Password for new Developer user <spxdevpass>
- &5 - Default tablespace <defaultspace> to which to assign <spx> and <spxdev>
- &6 - Temporary tablespace <tempspace> to which to assign <spx> and <spxdev>
- &7 - New user to own the SellingPoint Import database <imp> (for import staging purposes)
- &8 - Password for import user <imppass>
- &9 - Default tablespace <impdefaultspace> to which to assign <imp> (import staging tablespace)
- &10 - Temporary tablespace <imptempspace> to which to assign <imp>
- &11 - User indexes tablespace name <indxspace>
- &12 - Name of DB role for OSPC users for Grants <SPX_USER>
- &13 - Name of database service for SQL*Plus login <spxsid> (this location should be the same as where the Oracle Configurator Database will reside so it logs you in as the user there)

A.2.2.2 Restrictions

The restrictions on SP_BOOTSRAP.sql are that the script:

a. does not provide for per-table tablespace assignments. You can customize the table create commands to assign them to tablespaces without affecting application operation.

A.2.2.3 Actions

The actions taken by SP_BOOTSTRAP.sql:

a. create user &1 identified by &2 default tablespace &5 temporary tablespace &6;

b. grant create user to &1;

c. grant connect to &1 with admin option;

d. grant resource to &1 with admin option;

e. create user &7 identified by &8 default tablespace &9 temporary tablespace &10;

f. grant connect to &7;

g. grant resource to &7;

h. create role &12 not identified;

i. grant &12 to &1 with admin option;

j. create user &3 identified by &4;

k. grant connect to &3;

l. grant &12 to &3;

m. commit;

n. create CZ_DB_SETTINGS table and put the spx_user role there

o. connect &1 / &2@&13

p. create table CZ_DB_SETTINGS

COLUMN	DATA_TYPE	DATA_ LENGTH	NULLABLE
setting_id	VARCHAR	40	NOT NULL
section_name	VARCHAR2	30	NOT NULL,
data_type	NUMBER	8, 0	NULL
value	VARCHAR2	255	NULL
desc_text	VARCHAR2	255	NULL

q. alter table CZ_DB_SETTINGS

ADD CONSTRAINT CZ_DB_SETTINGS_PK (PRIMARY KEY (section_name, setting_id))

r. insert into CZ_DB_SETTINGS

setting_id	section_name	data_type	value	desc_text
0	DB_USER_ROLES	4	&12	Role name for a Oracle SellingPoint application sales end-user
SpxDefaultTablespace	SCHEMA	4	&5	Default tablespace to be used by <spx>, <spxdev>, and end users for permanent tables

setting_id	section_name	data_type	value	desc_text
SpxTemporaryTablespace	SCHEMA	4	&6	Temporary tablespace to be used by <spx>, <spxdev>, and end users for temporary tables
SpxIndxTablespace	SCHEMA	4	11	Tablespace to be used by <spx>, <spxdev>, and end users for indexes

s. commit;

A.2.3 Using GO_IMPORT.sql

A.2.3.1 Parameters

The parameters for GO_IMPORT.sql are specified as arguments. See Appendix A.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. See also Section 2.3.1, "Create Oracle Configurator DBOwner" on page 2-5.

- &1 - Name of Oracle Configurator Database user (<spx>, <spxdev>, etc.)
- &2 - Password for &1 (<spxpass>, <spxdevpass>, 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 <spxsid>

A.2.3.2 Restrictions

The restrictions on GO_IMPORT.sql are that the script:

a. assumes that 2 empty users (<spx> and <imp>) are available with sufficient privileges (connect, resource)

b. assumes all schemas should share the same database instance

c. should be run from <spx>

A.2.3.3 Actions

When you execute GO_IMPORT.sql, it does the following:

- **a.** creates SellingPoint online schema <spx>
- **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

A.2.4 Using InstAppsIntegrateViaLink.sql

A.2.4.1 Parameters

The parameters for InstAppsIntegrateViaLink.sql are specified as arguments. See Appendix A.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. See also Section 2.3.3.2, "Create the Oracle Configurator Database for Integration" on page 2-11.

- &1 - Owner of the Oracle Configurator Database <spx>
- &2 - Password for <spx> <spxpass>
- &3 - Owner of the SellingPoint Import/Integration database <imp>
- &4 - Password for <imp> <imppass>
- &5 - TNS entry for the Oracle Configurator Database <spxsid>
- &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>

A.2.4.2 Restrictions

The restrictions on InstAppsIntegrateViaLink.sql are that the script:

a. assumes that the Oracle Configurator Database schema has already been created in <spx> 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 <spx>

A.2.4.3 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>

A.2.5 Using GO_IMPORT_ONLY.sql

A.2.5.1 Parameters

The parameters for GO_IMPORT_ONLY.sql are specified as arguments. See Appendix A.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 4.4, "Run Generic Import" on page 4-21.

- &1 - Owner of the Oracle Configurator Database <spx>
- &2 - Password for <spx> <spxpass>
- &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 Database is installed <spxsid>

A.2.5.2 Restrictions

The restrictions on GO_IMPORT_ONLY.sql are that the script:

a. assumes that 2 empty users (<spx> and <imp>) are available with sufficient privileges (connect, resource)

b. both schemas should share the same database instance

c. should be run from <spx>

A.2.5.3 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

A.2.6 Using LoadAllBills.sql

A.2.6.1 Parameters

LoadAllBill.sql has no parameters.

A.2.6.2 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 <spx>

A.2.6.3 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

A.2.7 Using GRANT_SELECT_FOR.sql

A.2.7.1 Parameters

The parameters for GRANT_SELECT_FOR.sql are specified as arguments. See Appendix A.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 GRANT_SELECT_FOR.sql. See also Section 2.3.2, "Create OSPC Users and Responsibilities" on page 2-7.

- &1 -- Owner of the Oracle Configurator Database <spx>

A.2.7.2 Actions

When you execute GRANT_SELECT_FOR.sql, it does the following:

a. Grants ’select’ access to <spx> for importing data from Oracle Applications schema.

A.2.7.3 Restrictions

The restrictions on GRANT_SELECT_FOR.sql are that the script:

a. Must be run by <apps> owner.

A.2.8 Using EndUsers.sql

A.2.8.1 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.

A.2.8.1.1 (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 Database.

A.2.8.1.2 (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 A.2.8.1.3, "(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’.

A.2.8.1.3 (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 SellingPoint application needing access to the Oracle Configurator Database, 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 Database if the imported users weren’t automatically enabled (i.e., DB_SETTING AUTOCREATE_IMPORTED_USERS=‘NO’).

A.2.8.1.4 (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 Database.

A.2.8.1.5 (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.

A.2.8.1.6 (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.

A.2.8.1.7 (CTRL-c) Return to Previous Screen

Returns you to the previous prompt as you step through EndUsers.sql.

A.2.9 Summary of Actions by OSPC SQL*Plus Scripts

*Table A–4 Summary of Actions by OSPC 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 Database (<spx>)	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 A–4 Summary of Actions by OSPC 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		

A.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 SellingPoint Configurator CD-ROM, 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 A–5 Packages Used in OSPC 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 Database tables (in DBAdmin/Server/)
CZ_DEFAULTS.sql	Inserts initial data into the Oracle Configurator Database tables (in DBAdmin/Lite/)
CZ_EXPORT.sql	Package for Oracle Applications Order Export from the Oracle Configurator Database
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 Database tables (in DBAdmin/Server/)
CZ_INDEXES.sql	Creates indexes in SellingPoint schema (in DBAdmin/Lite/)
CZ_LC_MGR.sql	Manager package for LCE (compiled logic) subschema (in DBAdmin/Server/)

Table A–5 Packages Used in OSPC 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 Database 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 Database packages (in DBAdmin/Server/)
CZ_PK_CONSTRAINTS.sql	Defines primary keys on Oracle Configurator Database tables (in DBAdmin/Lite/)
CZ_PK_CONSTRAINTS.sql	Script creates primary key constraints on Oracle Configurator Database 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

Table A–5 Packages Used in OSPC Administrative Tasks

Package Name	Description
CZ_PS_COPY.sql	Package with support functions for project copy, copy with rules, rule-folder copy, etc. (in DBAdmin/Server/)
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 Database (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 Database tables (in DBAdmin/Lite/)
CZ_TABLES.sql	Script installs tables into Oracle Configurator Database database (in DBAdmin/Server/)
CZ_TRIGGERS.sql	Script attaches trigger procedures to tables in Oracle Configurator Database (in DBAdmin/Server/)
CZ_UI_MGR.sql	Manager package for User-interface subschema (in DBAdmin/Server/)
CZ_VIEWS.sql	Creates Oracle Configurator Database views (in DBAdmin/Lite/)
CZ_VIEWS.sql	Script installs views into Oracle Configurator Database (in DBAdmin/Server/)
CZ_XF_MGR.sql	Manager package for transfer (Import, export, integration) subschema (in DBAdmin/Server/)

Table A–5 Packages Used in OSPC Administrative Tasks

Package Name	Description
FND_STATS.sql	Standalone routine used for performance tuning. This routine is a replacement for the Oracle Applications package FND_STATS.
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

Table A–5 Packages Used in OSPC Administrative Tasks

Package Name	Description
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
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

Table A–5 Packages Used in OSPC Administrative Tasks

Package Name	Description
IMP_PS_NODE_B.sql	Body of import package for transferring Apps BOM data into the Oracle Configurator Database. 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 Database
IMP_PS_NODE_ONLY_B.sql	Body of import package for transferring imported product structure related data into the Oracle Configurator Database 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 Database without Oracle Applications
ospc_messages.sql	Inserts OSPC messages into FND_NEW_ MESSAGES (in DBAdmin/Server/)
ospc_messages_lite.sql	Inserts OSPC messages into FND_NEW_ MESSAGES (in DBAdmin/Lite)
TimeStampTrigger.class	Compiled class file for Java timestamp/user triggers (in DBAdmin/Lite/)
TimeStampTrigger.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/)

Table A–5 Packages Used in OSPC Administrative Tasks

Package Name	Description
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/)
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/)

B

Data Import Requirements

B.1 Overview

This appendix contains the following information:

Appendix B.2, "List of Import Tables" in the Oracle Configurator Database

Appendix B.3, "Dependencies Among Import Tables"

Appendix B.4, "Required ASCII File Format for Generic Import"

Appendix B.5, "Import Tables", which lists all import table fields and the data they expect.

B.2 List of Import Tables

The import tables are listed in the order in which the 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 (not populated by Release 4.2.2 import procedures)
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 4.2.2 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 4.2.2 import procedures)

B.3 Dependencies Among Import Tables

Dependencies among import tables must be heeded when importing single tables. In Table B–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 B–1 Dependencies Among Oracle Configurator Database 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 B–1 Dependencies Among Oracle Configurator Database 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

B.4 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|||||||||||||||||||
```

B.5 Import Tables

Table B–2, "Import Table Field Disposition Codes", describes the disposition codes that may result when required columns are queried against the source table. Table B–3, "Import Table Record Status Codes", describes the record status codes for required columns that have not resulted in a successful query. Table B–4 through Table B–21, describe all of the columns in the Oracle Configurator DB 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 B–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 B–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 B–4 Description of Fields in CZ_IMP_CUSTOMER Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
CUSTOMER_ID	N	M	NUMBER	Designates the SPX identifier for the account (mandatory null)
PRICE_LIST_ID	N	Y	NUMBER	Contains the SPX price list ID for this account
WAREHOUSE_ID	N	Y	NUMBER	Contains the SPX 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. Disposition: ■ if found - M ■ if not found - I ■ if not unique - R-DUPL ■ 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

Table B–4 Description of Fields in CZ_IMP_CUSTOMER 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 (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

Table B–4 Description of Fields in CZ_IMP_CUSTOMER Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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_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 B–5 Description of Fields in CZ_IMP_CUSTOMER_END_USER Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
CUSTOMER_ID	N	Y	NUMBE R	Contains the SPX account ID for an end user
END_USER_ID	N	Y	NUMBE R	Contains the SPX 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
EFF_TO	N	Y	DATE	Indicates the ending date through which this record is effective
CREATED_BY	N	Y	NUMBE R	Identifies the user that created this record
LAST_UPDATED_ BY	N	Y	NUMBE R	Records the login name for the user that last modified this record
SECURITY_MASK	N	Y	VARCH AR2 (40)	Reserved for future use in record-level access control
EFF_MASK	N	Y	VARCH AR2 (40)	Reserved for future use
REC_NBR	N	Y	NUMBE R	Provides a one-up sequence number identifying the record in this run
RUN_ID	N	Y	NUMBE R	Contains a number indicating the Import run in which this record was loaded / processed
REC_STATUS	N	Y	VARCH AR2 (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 B–5 Description of Fields in CZ_IMP_CUSTOMER_END_USER Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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. Disposition: ■ if found - assign customer ID ■ if not found - R-F13 ■ if not unique - R-DUPL ■ if null - R-N13
FSK_ENDUSER_2_1	Y	N	VARCHAR2(255)	Contains a foreign surrogate-key value matching CZ_END_USERS.LOGIN_NAME. Disposition: ■ if found - assign end user ID ■ if not found - R-F15 ■ if not unique - R-DUPL ■ 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. Disposition: ■ if found - M ■ if not found - I

Table B–6 Description of Fields in CZ_IMP_ADDRESS Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
ADDRESS_ID	N	Y	NUMBER	Designates the SPX identifier for the address
CUSTOMER_ID	N	Y	NUMBER	Contains the SPX 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
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

Table B–6 Description of Fields in CZ_IMP_ADDRESS Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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. Disposition: ■ if found - M ■ if not found - I ■ if not unique - R-DUPL ■ 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
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. Disposition: ■ if found - assign customer ID ■ if not found - R-F28 ■ if null - R-N28

Table B–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 SPX identifier for the address use
ADDRESS_ID	N	Y	NUMBER	Contains the SPX 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. Disposition: ■ if found - M ■ if not found - I ■ if not unique - R-DUPL ■ if null - R-N7
SITE_USE_CODE	N	Y	VARCHAR2 (20)	Contains code for site use
WAREHOUSE_ID	N	Y	NUMBER	Contains the SPX warehouse ID for address use
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. Disposition: ■ if found - assign address ID ■ if not found - R-F5 ■ if null - R-N5

Table B–8 Description of Fields in CZ_IMP_CONTACT Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
CONTACT_ID	N	Y	NUMBER	Designates the SPX identifier for this contact
CUSTOMER_ID	N	Y	NUMBER	Contains the SPX account ID for this contact
ADDRESS_ID	N	Y	NUMBER	Contains the SPX 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. Disposition: ■ if found - M ■ if not found - I ■ if not unique - R-DUPL ■ if null - R-N17
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

Table B–8 Description of Fields in CZ_IMP_CONTACT Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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 freeform 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
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

Table B–8 Description of Fields in CZ_IMP_CONTACT 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
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

Table B–8 Description of Fields in CZ_IMP_CONTACT Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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. Disposition: ■ if found - assign customer ID ■ if not found - R-F41 ■ if not unique - R-DUPL ■ 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. Disposition: ■ if found - assign address ID ■ if not found - R-F44 ■ if not unique - R-DUPL ■ if null - R-N44

Table B–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 SPX identifier for the development project
INTL_TEXT_ID	N	Y	NUMBER	Contains the SPX 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. Disposition: ■ if found - M ■ if not found - I ■ if not unique - R-DUPL ■ 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

Table B–9 Description of Fields in CZ_IMP_DEVL_PROJECT Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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
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 B–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 SPX 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. Disposition: ■ if found - M ■ if not found - I ■ if not unique - R-DUPL ■ 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

Table B–10 Description of Fields in CZ_IMP_END_USER Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
ADDR_LINE2	N	Y	VARCHAR2 (150)	Contains end user's second line address
CITY	N	Y	VARCHAR2 (56)	Contains end user's city name
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	VARHCAR2 (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

Table B–10 Description of Fields in CZ_IMP_END_USER 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	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

Table B–10 Description of Fields in CZ_IMP_END_USER Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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_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	VARHCAR2 (255)	Not currently used
NAME	N	Y	VARCHAR2 (255)	Contains name of the end user

Table B–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 SPX ID of the associated user
USER_GROUP_ID	N	Y	NUMBER	Contains the SPX 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
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

Table B–11 Description of Fields in CZ_IMP_END_USER_GROUP Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
LAST_UPDATE_ DATE	N	Y	DATE	Contains the date on which the record was last modified
CREATED_BY	N	Y	NUMBE R	Identifies the user that created this record
LAST_UPDATED_ BY	N	Y	NUMBE R	Records the login name for the user that last modified this record
SECURITY_MASK	N	Y	VARCH AR2 (40)	Reserved for future use in record-level access control
FSK_ENDUSER_1_ 1	Y	N	VARCH AR2 (255)	Contains a foreign surrogate-key value matching CZ_ END_USERS.ORIG_SYS_REF. Disposition: ■ if found - assign end user ID ■ if not found - R-F20 ■ if not unique - R-DUPL ■ if null - R-N20
FSK_ENDUSER_1_ EXT	N	Y	VARCH AR2 (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

Table B–11 Description of Fields in CZ_IMP_END_USER_GROUP Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
FSK_USERGROUP_2_1	Y	N	VARCHAR2 (255)	Contains a foreign surrogate-key value matching CZ_USER_GROUPS.GROUP_NAME. Disposition: ■ if found - assign user group ID ■ if not found - R-F22 ■ if not unique - R-DUPL ■ 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. Disposition: ■ if found - M ■ if not found - I

Table B–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 SPX 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. Disposition: ■ if found - M ■ if not found - I ■ if not unique - R-DUPL
CREATION_DATE	N	Y	DATE	Indicates the date on which this record was created

Table B–12 Description of Fields in CZ_IMP_INTL_TEXT Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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	NUMBE R	Identifies the user that created this record
LAST_UPDATED_ BY	N	Y	NUMBE R	Records the login name for the user that last modified this record
SECURITY_MASK	N	Y	VARCH AR2 (40)	Reserved for future use in record-level access control
EFF_MASK	N	Y	VARCH AR2 (40)	Reserved for future use
CHECKOUT_USER	N	Y	VARCH AR2 (40)	Reserved for future use in checkout user
RUN_ID	N	Y	NUMBE R	Contains a number indicating the Import run in which this record was loaded / processed
REC_STATUS	N	Y	VARCH AR2 (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 B–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 SPX identifier for this record
ITEM_TYPE_ID	N	Y	NUMBER	Contains the SPX 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. Disposition: ■ if found - M ■ if not found - I ■ if not unique - R-DUPL ■ 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
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

Table B–13 Description of Fields in CZ_IMP_ITEM_MASTER 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

Table B–13 Description of Fields in CZ_IMP_ITEM_MASTER Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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. Disposition: ■ if found - assign item type ID ■ if not found - assign default item type ID, if defined, R-F27 ■ 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 B–14 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 SPX ID for the referenced property
ITEM_ID	N	Y	NUMBER	Contains the SPX 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
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

Table B–14 Description of Fields in CZ_IMP_ITEM_PROPERTY_VALUE 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_PROPERTY_1_1	Y	N	VARCHAR2 (255)	Contains a foreign surrogate-key value matching CZ_PROPERTIES.ORIG_SYS_REF. Disposition: ■ if found - assign property ID ■ if not found - R-F23 ■ if not unique - R-DUPL ■ if null - R-N23

Table B–14 Description of Fields in CZ_IMP_ITEM_PROPERTY_VALUE Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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. Disposition: ■ if found - assign item ID ■ if not found - R-F25 ■ if not unique - R-DUPL ■ 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. Disposition: ■ if found - M ■ if not found - I

Table B–15 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 SPX 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. Disposition: ■ if found - M ■ if not found - I ■ if not unique - R-DUPL ■ 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

Table B–15 Description of Fields in CZ_IMP_ITEM_TYPE Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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

Table B–15 Description of Fields in CZ_IMP_ITEM_TYPE Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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 B–16 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 SPX identifier for an item type
PROPERTY_ID	N	Y	NUMBER	Contains the SPX 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

Table B–16 Description of Fields in CZ_IMP_ITEM_TYPE_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

Table B–16 Description of Fields in CZ_IMP_ITEM_TYPE_PROPERTY Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
LAST_UPDATED_ BY	N	Y	NUMBE R	Records the login name for the user that last modified this record
SECURITY_MASK	N	Y	VARCH AR2 (40)	Reserved for future use in record-level access control
FSK_ITEMTYPE_1_ 1	Y	N	VARCH AR2 (255)	Contains a foreign surrogate-key value matching CZ_ ITEM_TYPES.ORIG_SYS_REF. Disposition: ■ if found - assign item type ID ■ if not found - R-F22 ■ if not unique - R-DUPL ■ if null - R-N22
FSK_ITEMTYPE_1_ EXT	N	Y	VARCH AR2 (255)	Contains a foreign surrogate-key value matching CZ_ IMP_ITEM_TYPE on USER_STR03
FSK_PROPERTY_2_ 1	Y	N	VARCH AR2 (255)	Contains a foreign surrogate-key value matching CZ_ PROPERTIES.ORIG_SYS_REF. Disposition: ■ if found - assign property ID ■ if not found - R-F24 ■ if not unique - R-DUPL ■ if null - R-N24
FSK_PROPERTY_2_ EXT	N	Y	VARCH AR2 (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. Disposition: ■ if found - M ■ if not found - I

Table B–17 Description of Fields in CZ_IMP_PRICE Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
ITEM_ID	N	Y	NUMBER	Contains the SPX Item-Master ID for this record
PRICE_GROUP_ID	N	Y	NUMBER	Contains the SPX 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
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

Table B–17 Description of Fields in CZ_IMP_PRICE Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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	Y	N	VARCHAR2 (255)	Contains a foreign surrogate-key value matching CZ_ITEM_MASTERS.ORIG_SYS_REF. Disposition: ■ if found - assign item ID ■ if not found - R-F28 ■ if not unique - R-DUPL ■ 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. Disposition: ■ if found - assign price group ID ■ if not found - R-F30 ■ if not unique - R-DUPL ■ 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. Disposition: ■ if found - M ■ if not found - I

Table B–18 Description of Fields in CZ_IMP_PRICE_GROUP Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
PRICE_GROUP_ID	N	Y	NUMBE R	Designates an SPX identifier for this price group
DESC_TEXT	N	Y	VARCH AR2 (255)	Describes this price group
NAME	N	Y	VARCH AR2 (255)	Contains the name of the Price Group
ORIG_SYS_REF	Y	N	VARCH AR2 (255)	Unique identification of a record in this table. Contains a foreign surrogate-key value matching CZ_PRICE_ GROUPSS.ORIG_SYS_REF. Disposition: ■ if found -M ■ if not found - I ■ if not unique - R-DUPL ■ if null - R-N5
REC_NBR	N	Y	NUMBE R	Provides a one-up sequence number identifying the record in this run
RUN_ID	N	Y	NUMBE R	Contains a number indicating the Import run in which this record was loaded / processed
REC_STATUS	N	Y	VARCH AR2 (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	VARCH AR2 (20)	Contains the type of currency used
EFF_FROM	N	Y	DATE	Indicates the beginning date for which this record is effective

Table B–18 Description of Fields in CZ_IMP_PRICE_GROUP 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

Table B–18 Description of Fields in CZ_IMP_PRICE_GROUP Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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 B–19 Description of Fields in CZ_IMP_PROPERTY Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
PROPERTY_ID	N	Y	NUMBER	Designates an SPX 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

Table B–19 Description of Fields in CZ_IMP_PROPERTY Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
RUN_ID	N	Y	NUMBE R	Contains a number indicating the Import run in which this record was loaded / processed
REC_STATUS	N	Y	VARCH AR2 (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	VARCH AR2 (255)	Unique identification of a record in this table. Contains a foreign surrogate-key value matchingCZ_ PROPERTIES.ORIG_SYS_REF. Disposition: ■ if found - M ■ if not found - I ■ if not unique - R-DUPL ■ if null - R-N17
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	VARCH AR2 (40)	Reserved for future use
USER_STR01	N	Y	VARCH AR2 (255)	Textual user expansion field
USER_STR02	N	Y	VARCH AR2 (255)	Textual user expansion field
USER_STR03	N	Y	VARCH AR2 (255)	Textual user expansion field; may be used as an alternate 'surrogate key' for the record

Table B–19 Description of Fields in CZ_IMP_PROPERTY Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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 B–20 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 SPX Identifier for the model node
DEVL_PROJECT_ ID	N	Y	NUMBER	Contains the SPX development project ID for this record
FROM_ POPULATOR_ID	N	Y	NUMBER	Set when the node is created by a populator; designates the SPX 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 SPX international text ID for this record
SUB_CONS_ID	N	Y	NUMBER	Currently not used
ITEM_ID	N	Y	NUMBER	Contains the SPX item ID for this record
NAME	N	Y	VARCHAR2 (255)	Contains the SPX 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 SPX 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

Table B–20 Description of Fields in CZ_IMP_PS_NODE Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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
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 oder 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. Disposition: ■ if found - M ■ if not found - I ■ if not unique - R-DUPL ■ if null - R-N9

Table B–20 Description of Fields in CZ_IMP_PS_NODE Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
RUN_ID	N	Y	NUMBE R	Contains a number indicating the Import run in which this record was loaded / processed
REC_STATUS	N	Y	VARCH AR2 (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	VARCH AR2 (40)	Reserved for future use
USER_STR01	N	Y	VARCH AR2 (255)	Textual user expansion field
USER_STR02	N	Y	VARCH AR2 (255)	Textual user expansion field
USER_STR03	N	Y	VARCH AR2 (255)	Textual user expansion field; may be used as an alternate 'surrogate key' for the record
USER_STR04	N	Y	VARCH AR2 (255)	Textual user expansion field
USER_NUM01	N	Y	NUMBE R (16,9)	Numeric user expansion field
USER_NUM02	N	Y	NUMBE R (16,9)	Numeric user expansion field

Table B–20 Description of Fields in CZ_IMP_PS_NODE Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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_INTLTEXT_1_1	Y	N (see description)	VARCHAR2 (255)	Contains a foreign surrogate-key value matching CZ_INTL_TEXTS.TEXT_STR. Disposition: ■ if found - assign international text ID ■ if not found - R-F44 (except model and project structure nodes) ■ 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. Disposition: ■ if found - assign item ID ■ if not found - R-F46 (except model and project structure nodes)

Table B–20 Description of Fields in CZ_IMP_PS_NODE Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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. Disposition: ■ if found - assign parent Project Structure node ID ■ if not found - R-F48 (except model nodes) ■ 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
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. Disposition: ■ if found - assign development project ID ■ if not found - R-F50 ■ 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

Table B–20 Description of Fields in CZ_IMP_PS_NODE Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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 B–21 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 SPX 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. Disposition: ■ if found - M ■ if not found - I ■ if not unique - R-DUPL ■ if null - R-N2

Table B–21 Description of Fields in CZ_IMP_USER_GROUP Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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	VARCHA2 (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
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

Table B–21 Description of Fields in CZ_IMP_USER_GROUP Import Table

COLUMN_NAME	Required	Nullable	DATA_TYPE	Description
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

C

Data Export Requirements

C.1 Overview

This appendix contains the following information:

Section C.2, "Export Data Source and Destination", which presents what data in the Oracle Configurator Database are exported to which tables and fields in the Oracle Applications database.

C.2 Export Data Source and Destination

Table C–1 presents the data in the Oracle Configurator Database that are exported to populate the destination 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 SellingPoint application are complete.

Table C–1 Data Export to Oracle Applications Source and Interface Tables

Oracle Configurator Database Source	->	Oracle Applications Interface Table(s)
CZ_QUOTE_HDRS.QUOTE_HDR_ID CZ_QUOTE_HDRS.QUOTE_REV_NBR Integer for standard User ID	-->	SO_HEADERS_INTERFACE_ALL.ORIGINAL_SYSTEM_REFERENCE SO_LINES_INTERFACE_ALL.ORIGINAL_SYSTEM_REFERENCE SO_PRICE_ADJUSTMENT_INTERFACE.ORIGINAL_SYSTEM_REFERENCE
CZ_QUOTE_HDRS.ORDER_SUBMITTED_DATE	-->	SO_LINES_INTERFACE_ALL.CREATION_DATE SO_PRICE_ADJUSTMENT_INTERFACE.CREATION_DATE
CZ_QUOTE_HDRS.USER_ID_CREATED	-->	SO_HEADERS_INTERFACE_ALL.CREATED_BY SO_LINES_INTERFACE_ALL.CREATED_BY SO_PRICE_ADJUSTMENT_INTERFACE.CREATED_BY

Table C–1 Data Export to Oracle Applications Source and Interface Tables

Oracle Configurator Database Source	->	Oracle Applications Interface Table(s)
CZ_QUOTE_HDRS.ORDER_SUBMITTED_DATE	-->	SO_HEADERS_INTERFACE_ALL.LAST_UPDATE_DATE SO_LINES_INTERFACE_ALL.LAST_UPDATE_DATE SO_PRICE_ADJUSTMENT_INTERFACE.LAST_UPDATE_DATE
CZ_QUOTE_HDRS.USER_ID_CREATED	-->	SO_HEADERS_INTERFACE_ALL.LAST_UPDATED_BY SO_LINES_INTERFACE_ALL.LAST_UPDATED_BY SO_PRICE_ADJUSTMENT_INTERFACE.LAST_UPDATED_BY
CZ_QUOTE_HDRS.ORDER_SUBMITTED_DATE	-->	SO_HEADERS_INTERFACE_ALL.DATE_ORDERED
CZ_QUOTE_HDRS 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 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 SO_LINES_INTERFACE_ALL.ORDER_SOURCE_ID 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

Table C–1 Data Export to Oracle Applications Source and Interface Tables

Oracle Configurator Database Source	->	Oracle Applications Interface Table(s)
CZ_ITEM_MASTERS.PRIMARY_UOM_CODE	-->	SO_LINES_INTERFACE_ALL.UNIT_CODE
Function of tCZ_ITEM_MASTERS.ORIG_SYS_REF 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 SO_LINES_INTERFACE_ALL.ORG_ID
CZ_CUSTOMERS.CUSTOMER_ID	-->	RA_CUSTOMER_PROFILES_INT_ALL.ORIG_SYSTEM_CUSTOMER_REF 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

Table C–1 Data Export to Oracle Applications Source and Interface Tables

Oracle Configurator Database Source	->	Oracle Applications Interface Table(s)
Function of CZ_CUSTOMERS.ORIG_SYS_REF derived from QUOTE_HDR.SHIPTO_CUSTOMER_ID	-->	SO_HEADERS_INTERFACE_ALL.SHIP_TO_CUSTOMER_ID
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 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 RA_CUSTOMER_PROFILES_INT_ALL.LAST_UPDATE_DATE
Integer for standard User ID	-->	RA_CUSTOMERS_INTERFACE_ALL.CREATED_BY RA_CUSTOMER_PROFILES_INT_ALL.CREATED_BY
SYSDATE	-->	RA_CUSTOMERS_INTERFACE_ALL.CREATION_DATE RA_CUSTOMER_PROFILES_INT_ALL.CREATION_DATE
Integer for standard User ID	-->	RA_CUSTOMERS_INTERFACE_ALL.LAST_UPDATE_LOGIN 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

Table C–1 Data Export to Oracle Applications Source and Interface Tables

Oracle Configurator Database Source	->	Oracle Applications Interface Table(s)
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
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

Glossary of Terms

This glossary contains definitions of terms you might encounter while working with the Oracle SellingPoint Configurator software and documentation.

Acceptance Test

Test for validating the system (the correctness of results). Acceptance tests are based on acceptance criteria specified in the project's Test Plan.

Active Model

The part of Oracle SellingPoint Configurator runtime architecture that processes model structure and rules to create configurations. Interfaces dynamically with the end user Active User Interface and data.

Active User Interface

The part of Oracle SellingPoint Configurator runtime architecture that provides the 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.

Alpha

An internal release of the application before implementation is complete, delivered as a build, and subject to integration, verification, and system testing.

Application

The Oracle SellingPoint application. The end-user runtime environment that provides configuration functionality and output. Also called sales configuration application or enterprise selling system. See also Oracle Configurator.

Application Architecture

The software structure of an application at runtime. Architecture affects how an application is used, maintained, extended, and changed.

Application Architecture and Design Document

Document presenting the overall architecture for the application and how the application will be implemented.

Application Design

The task in the implementation stage of a project for determining, documenting, reviewing, and delivering the scheme that will turn user requirements into an operational application. Occurs in parallel with the end of the Test Case Definition task and the beginning of the Construction task. Application Design results in an Application Architecture and Design Document.

Application Development

See Construction.

Application Implementer

The person who uses Oracle SellingPoint Developer to construct an Oracle SellingPoint application or the model structure, rules, and UI customizations for a Oracle Configurator.

Application Testing

See Full Application Testing.

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 SellingPoint Developer Model that correspond to an Oracle BOM.

BOM Model

The imported Model node in the Oracle SellingPoint Developer that corresponds to Standard Model in an Oracle BOM.

BOM OptionClass

The imported Model node in the Oracle SellingPoint Developer that corresponds to Option Class in an Oracle BOM.

BOM StandardItem

The imported Model node in the Oracle SellingPoint 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.

Change Control Board

A group of people responsible for evaluating Change Request Forms, approving or rejecting them, and notifying affected parties of how each one was resolved.

Change Control Procedures

A plan that describes how change control will be conducted during a project.

Change Request Form

A form used to propose changes as part of a standard change control process. A Change Request Form typically includes a description of the proposed change and an evaluation of impacts on cost and schedule.

CIO

See Oracle Configuration Interface Object.

CIO protocols support creating and navigating the Model, querying and modifying selection states, and saving and restoring configurations.

Client

A runtime program using a server to access functionality shared with other clients.

Comparison Rule

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

A relationship among features in the Model that specifies the allowable combinations of options.

Compatibility Rule

A kind of compatibility relationship where the allowable combinations of options are specified implicitly by relationships between property values of the options.

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 typically containing features.

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 Management

A process for managing the versions of the application and its documentation during construction.

Configuration Model

The model structure and rules-based content of an Oracle SellingPoint application or Oracle Configurator. The configuration model is constructed and maintained using Oracle SellingPoint Developer, and is interpreted at runtime by the Active Model.

Configuration Rules

The logic rules and numeric rules available for defining configurations.

Configurator

The part of the Oracle SellingPoint application or Oracle Configurator that provides custom configuration capabilities.

Constraint Rule

A logical relationship amongst features and options. See also Rules.

Construction

The task in the implementation stage of a project for building the Oracle SellingPoint application using Oracle SellingPoint Developer. Construction is based on the user’s requirements and an approved application design. Occurs in parallel with completion of the Design task. Construction includes reviews and testing.

Contributes

A numeric rule for accumulating a total value.

Consumes

A numeric rule for specifying the quantity of a resource used.

Core Functionality

Also called Phase 1. The first release of the application delivered in 14 weeks. After validation with a subset of users, the core functionality application can be fully deployed to all intended users, maintained, or extended to offer additional product families or more functionality. See Full Deployment, Additional Product Families, and Extended Functionality.

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 SellingPoint application or other Order Management and CRM applications.

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-centric Extensions

See Customer-centric Views.

Customer Requirements

The needs of the customer that serve as the basis for determining the configuration of products, systems, and/or services.

Data Import

Populating the Oracle Configurator Database 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 Database.

Data Maintenance Environment

The environment in which the Oracle SellingPoint application or Oracle Configurator data is maintained.

Data Replication

The activity of downloading and uploading configuration, quote, and order data between the Oracle Configurator Database 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.

Data Synchronization

A process for matching the data in the Oracle Configurator Database 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

A 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 work-arounds 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.

Definition

Defining and scoping the first phase of a project, and selecting a vendor such as Oracle to implement a sales configuration application.

Deliverable

A work product that is specified for review and delivery.

Delivery

The task in the implementation stage of a project for organizing a deployment of the application. Includes beta testing.

Demonstration

A presentation of the tested application, showing a particular usage scenario.

Deployment

The stage in a project between implementation and maintenance when the fully operational application is distributed to users. See also Pilot and Full Deployment.

Design

See Application Design.

Design Chart

An Oracle SellingPoint 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.

Development

See Construction.

DIO

See Data Integration Object.

Domain Expert

A member of the customer's staff who has specific product or process knowledge needed in the Oracle SellingPoint application or Oracle Configurator.

End User

The ultimate user of the Oracle SellingPoint application or Oracle 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

A logic rule to determine 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, OMS, and COM-compliant third-party software.

Feature

An element of the Model. A configurable parameter of a component. Features can either have a value (numeric or boolean) or enumerated options.

Full Application Testing

The task in the implementation stage of a project for testing the constructed application prior to delivery. Full Application Testing results in a completely validated application approved for delivery to users.

Full Deployment

A release of the application to all intended users after implementation and validation of the core functionality.

Full Roll Out

An external release delivered to all intended end users of the application. See also Full Deployment.

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.

Increments

A logical relation that increments a count or value associated with an item by an integer quantity.

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.

Implies

A logic rule 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 Database Item Master structure, but without integrity constraints. Import tables allow batch population of the Oracle Configurator Database 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 Database 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 Database 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 Database containing data used to classify the product data in the item master table.

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.

Maintenance

The effort of keeping a system running once it has been deployed, through bug fixes, procedure changes, infrastructure adjustments, data replication schedules, etc.

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 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. Oracle's Application and System Maintenance Plan presents a generic model of activities and deliverables necessary for successful maintenance.

OC

See Oracle Configurator.

Methodology

A standard, step-by-step process designed to achieve consistent, reliable results. Oracle Configurator Deployment Methodology is a repeatable implementation process based on software development standards and Oracle SellingPoint application implementation best practices.

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.

Oracle Configurator

The 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, components, features, and options). May include reusable components.

MRP

Manufacturing Resource Planning. A software system and process for monitoring and maintaining the customer's manufacturing systems.

Negates

A logic rule to determine 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.

Next Phase

The phase following Phase 1 Core Functionality, consisting of Full Deployment of Phase 1, expanding the application for coverage of additional Product Families or extending the functionality to include more complexity and customer-centric views.

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. These include Contributes, Supplies, and Consumes. See also Numeric Rules, Contributes, Supplies, and Consumes.

OSPC

See Oracle SellingPoint Configurator.

Opportunity

The workspace in the Oracle SellingPoint application and Oracle Field Sales 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 SellingPoint Configurator

The product family consisting of development tools and runtime applications such as Oracle SellingPoint Developer and Oracle SellingPoint, variously packaged for use in networked, mobile, or web deployments.

Oracle Configurator Database

The implementation version of the standard Oracle SellingPoint application or Oracle 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 application.

Oracle Configurator Deployment Methodology

The methodology of stages, tasks, steps, and activities to deliver a core functionality, Phase 1 Oracle SellingPoint application to users in 14 weeks. Oracle Configurator Deployment Methodology includes templates and checklists for organizing and managing an Oracle SellingPoint application project.

Oracle Configurator Deployment Methodology Program

The rapid application development program that combines the tools of Oracle SellingPoint Developer with the application construction methodology and project management methods of Oracle Configurator Deployment Methodology to ensure delivery of a core functionality Phase 1 Oracle SellingPoint application application in 14-weeks. Oracle Education provides books and courses in support of using Oracle SellingPoint Developer and Oracle Configurator Deployment Methodology most effectively. See also Oracle SellingPoint Developer and Oracle Configurator Deployment Methodology.

Oracle Configuration Interface 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 underlying representation of model structure and rules in the Active Model.

Oracle Configurator Mobile Database

The runtime version of the standard Oracle Configurator Database that manages data for the configuration model. The runtime schema includes customer, product, and pricing data as well as data created during operation of an Oracle SellingPoint application.

Oracle SellingPoint Developer

The suite of tools in the Oracle SellingPoint Configurator product for constructing and maintaining sales configuration applications.

Oracle SellingPoint Application

The end-user application created with the Oracle SellingPoint Developer product. See also Application and Oracle Configurator.

Oracle SellingPoint Application Architecture

The application runtime architecture consists of the Active User Interface, the Active Model, and the Oracle Configurator Database or Oracle Configurator Mobile Database. The application development architecture consists of Oracle SellingPoint Developer and the Oracle Configurator Database.

Output

The output generated by the sales configuration application, 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.

Phase 1 (one)

All the sales configuration functionality that can be implemented within the Active User Interface and Oracle SellingPoint Developer within the 14-week Oracle Configurator Deployment Methodology program, and nothing that requires programming outside of these environments. See also Core Functionality.

Pilot

An external release to a subset of 20-25 end users for validation of the system.

Populator

An entity in the Oracle SellingPoint Developer that defines how to create a Model from information in the item master.

Preliminary Project Plan

An initial high-level project plan and schedule describing the events in the project in terms of time relations rather than specific delivery dates.

Pre-selection

The default state in a sales configuration application that defines an initial selection of components, features, and options for sales 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.

Product Maintenance

A release of the application after delivery of core functionality that adds wider product coverage or keeps product data correct.

Product Structure

See Model Structure

Project

A project is the process of implementing and delivering an Oracle SellingPoint application or Oracle Configurator.

A Project in Oracle SellingPoint Developer is the workspace in which sales configuration applications 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.

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 4.2 or before.

Regression Test

An automated test that ensures the newest build still meets previously tested requirements and functionality.

Requirements

The task in the implementation stage of a project when the project team explores and understands what the application will do. Occurs in parallel with starting the implementation task Test Case Definition. Requirements gathering results in a Functional Specification.

Requires

A logic rule to determine 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 4.2 or before.

Reusability

The extent to and ease with which parts of a system can be put to use in other systems.

Roll Out

See Full Roll Out.

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.

SellingPoint

The configuration engine used in the Oracle SellingPoint Configurator.

Server

Centrally located software processes or hardware, shared by clients.

Solution

The deployed system as a response to a problem or problems.

Statement of Work

Document describing the work required to deliver an application based on pre-sales scoping activities.

Studio

See Oracle SellingPoint Developer.

Supplies

A numeric rule for specifying how much of a resource is available.

System

The hardware and software components and infrastructure integrated to satisfy sales configuration requirements.

System Project

The project of implementing the solution, including the Oracle SellingPoint application or Oracle Configurator. See also Project.

System User

Any user in contact with the system.

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.

Test Case Definition

The task in the implementation stage of a project for defining the Test Cases and describing the tests that will be performed to validate the application. Occurs in parallel with the end of the Requirements task and much of the Design task. Test Case Definition results in a detailed Test Plan for the project.

Testing

See Full Application Testing

Test Log

A record of what tests were run at what time on which version of the application to what effect.

Test Plan

The plan for defining and executing tests.

Timeline

The schedule for completing the tasks and activities required to implement a phase of the sales configuration application project.

Total

A variable in the Model used to accumulate a numeric total, such as total price or total weight.

Training

Training that prepares Oracle SellingPoint Configurator users for creating, changing, extending, and supporting the application.

Training that prepares the application end user for operating the system.

Training Guide

A guide created by members of the project team to instruct end users how to use the system. See also User's Guide.

Undetermined

The logic state that is neither true nor false, but unknown at the time a logic rule is executed.

Unit Test

Execution of individual routines and modules by the application implementer or by an independent test consultant for the purposes of finding defects.

User

The person using the Oracle SellingPoint Configurator tools and methods to build an Oracle SellingPoint application or Oracle 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 SellingPoint application or Oracle Configurator is expected to do from the end user's perspective.

User's Guide

Documentation on using the application to solve the intended problem. See also Training Guide.

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 represent either totals or resources.

Verification

Tests that check whether the result agrees with the specification.

Glossary of Acronyms

This glossary contains a reference list of acronyms you may encounter while using the Oracle SellingPoint Configurator and its documentation. Each acronym is listed with the full version of the group of words it represents.

API

Application Programming Interface

ATP

Available to Promise

BOM

Bill of Material

CB

Companion Builder

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

FC

Functional Companion

GSE

Generative Specification Environment

GSL

Generative Specification Language

HT

High Tech

HTML

Hypertext Markup Language

IP

Industrial Products

Information Provider

IS

Information Services

ISS

Interactive Selling System

ISV

Independent Software Vendor

LAN

Local Area Network

LCE

Logical Configuration Engine

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

OApp

Oracle Applications

OAS

Oracle Application Server

OCD

Oracle Configurator Developer

OCDB

Oracle Configurator Database

OCDFC

Oracle Configurator Developer: Functional Companions

OCEE

Oracle Configurator Engineering Edition

OCIE

Oracle Configurator Internet Edition

OCMDB

Oracle Configurator Mobile Database

OCX

Object Control File, OLE custom controls

ODBC

Open Database Connectivity

OLE

Object linking and embedding

OMA

Oracle Mobile Agents

OMS

Opportunity Management System

OOD

Object-Oriented Design

ORB

Object Request Broker

OSP

Oracle SellingPoint

OSPA

Oracle SellingPoint application

OSPC

Oracle SellingPoint Configurator

PDM

Product Data Management

PIA

Project Impact Assessment

PM

Project Manager

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

SOD

Statement of Direction

SOT

Strategic Options Theory

SOW

Statement of Work

CDBI

Configurator Database Interface

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

Y2K

Year 2000 Compliant

Index