Oracle® Configurator

Installation Guide

Release 11*i*

September 2000

Part No. A85421-01

ORACLE®

Oracle Configurator Installation Guide, Release 11*i*

Part No. A85421-01

Primary Authors: Denise Boyer

Contributing Authors: Tina Brand, Mark Sawtelle

Contributors: Tim Abbott, Adam Schwartz

Contents

Send Us Your Comments ix

Preface xi

Audience xi
Structure xii
Related Documents xii
Conventions xiii

1 Installing Oracle Configurator

1.1 System Requirements 1–2
1.2 Supported Platforms 1–3
1.3 Required Patches 1–3
1.4 Completing the Oracle Configurator Installation 1–3
1.4.1 Modify Database Configuration File Parameters 1–5
1.4.2 Set Profile Options 1–5
1.4.3 Set up Oracle Applications Responsibilities and Users 1–6
1.4.4 Verify and Test Apache Configuration 1–6
1.5 Migrating From a Standalone Oracle SellingPoint Schema 1–7
1.5.1 Migration Prerequisites 1–7
1.5.2 Migrate to Release 11*i* 1–7

2 UI Servlet Considerations

2.1 Configuring Apache and JServ 2–1
2.1.1 Verifying httpd.conf 2–3
2.1.2 Verifying jserv.conf 2–4
2.1.3 Verifying jserv.properties 2–5
2.1.4 Verifying zone.properties 2–6
2.1.5 Verifying Java System Properties 2–7
2.2 Load Balancing 2–8
2.3 Java System Property Parameters for the UI Servlet 2–12

3 Troubleshooting Servlet Installation

3.1 Miscellaneous 3–1
3.2 Checking the Operation of the Apache Internet Server 3–1
3.3 Checking the Response of the UI Servlet 3–3
3.4 Restarting the UI Server 3–4
3.5 Checking Your Model in the Oracle Configurator Window 3–5

4 Installing Oracle Configurator Developer

4.1 The Oracle Configurator Developer Software 4–1
4.2 The Oracle Configurator Developer Installation 4–2
4.3 Oracle Configurator Developer Client/Server Setup 4–3
4.3.1 Oracle Configurator Developer Server Machine 4–4
4.3.1.1 Establish Users and Responsibilities 4–5
4.3.1.2 Accessing Oracle Configurator Windows 4–5
4.3.2 Oracle Configurator Developer Client Machine 4–6
4.3.2.1 Oracle Configurator Window Client Requirements 4–6
4.3.2.2 Oracle Client 8.0.6 Installation 4–7
4.3.2.3 Set Up Oracle Configurator Developer 4–7
4.3.2.4 Enable the Client for Database Connectivity 4–8
4.3.2.5 Create DSNs and DBOwners 4–10
4.3.2.6 Set Parameters in the spx.ini File for Development and Testing 4–11
4.4 Test Your Oracle Configurator Developer Installation 4–14

Glossary of Terms

Glossary of Acronyms

Index

List of Examples

2–1 Default Java System Properties Created by Oracle Rapid Install 2–8
2–2 Shell Script for Starting JServ in Manual Mode (in Oracle Applications) 2–11
3–1 Hello.java Test Class 3–2
3–2 HTML Test Page for Invoking the DHTML Oracle Configurator window 3–5

List of Figures

3–1 A Model, in the Oracle Configurator window 3–7
4–1 Server Configuration for OC Server 4–4
4–2 Client/Server Data Communication Architecture 4–8

List of Tables

2–1 Textual Placeholders for Configuration Files 2–2
2–2 Java System Properties for the UI Servlet 2–13
4–1 Spx.ini Parameters for Development and Testing 4–12

Send Us Your Comments

Oracle Configurator Installation Guide, Release 11*i*

Part No. A85421-01

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

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

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

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

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

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

Preface

This *Oracle Configurator Installation Guide* provides explanations and instructions for tasks required to install the Oracle Configurator schema, Oracle Configurator Developer, and Oracle Configurator windows.

Audience

If you are responsible for installing Oracle Configurator, be sure you have read and understand the information in *Oracle Applications Concepts* and *Installing Oracle Applications*. *Oracle Applications Concepts* explains the technology, architecture, and terminology used with all Oracle Applications. *Installing Oracle Applications* provides instructions for installing Oracle Applications products and the Oracle Configurator schema using Oracle Rapid Install.

This manual is intended for anyone installing or supporting the installation of Oracle Configurator (OC).

Ordinarily, the tasks presented in this book are performed by one of the following people:

- System Administrator

  Responsible for administering the Oracle Applications system, including:

  - Ensuring that hardware is correctly configured
  - Installing, configuring, and maintaining production and development software
  - Ensuring that the system is backed up daily
  - Designing and maintaining system security such as system accounts

The system administrator provides support for problems with the system. They may perform setup and initial maintenance of the production system or advise their client's operational staff on these tasks. The system administrator works with the project team to optimize system performance, install packaged applications environments, and convert data.

- Database Administrator

  Installs and configures the Oracle8<italic>i database and maintains database access controls. This person also provides consultation on performance and is responsible for monitoring growth and fragmentation of the production database and ensuring database backup and recovery.

Structure

This manual contains the following chapters and appendixes:

- "Installing Oracle Configurator" provides an overview of Oracle Configurator and describes the software components and system requirements. It also describes installation prerequisites and provides information for installing Oracle Configurator with Oracle Applications Release 11*i* and upgrading from a prior standalone version of the Oracle Configurator schema to the current version.
- "UI Servlet Considerations" describes the tasks to install, configure, and adjust your Apache configuration to balance the load of visits to the UI Servlet by your end users.
- "Installing Oracle Configurator Developer" describes prerequisites and provides information for installing Oracle Configurator Developer from the Oracle Configurator Developer compact disc. It also describes how to set up Oracle Configurator Developer client and server machines in order to use Oracle Configurator Developer to create a custom Oracle Configurator window.
- "Glossary of Terms"
- "Glossary of Acronyms"
- "Index"

Related Documents

For more information, see the documentation for your release of Oracle Applications, Release 8i Oracle RDBMS documentation, Oracle Configurator

documentation, and the product-specific *Release Notes* for releases supported to work with Oracle Configurator.

Conventions

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

The following conventions are also used in this manual:

Convention	Meaning
. . .	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.
italic text	Italic text in code examples indicates user-supplied parameters or arguments.
[ ]	Brackets enclose optional clauses from which you can choose one or none.

1

Installing Oracle Configurator

Oracle Configurator consists of the Oracle Configurator schema, Oracle Configurator Developer (a graphical, drag-and-drop development and maintenance environment), and a runtime configurator called the Oracle Configurator window (the end-user environment). The Oracle Configurator schema and the Oracle Configurator window are installed with Oracle Applications Release 11*i* using Oracle Rapid Install. The Oracle Configurator Developer is installed from the Oracle Configurator Developer compact disc.

The Oracle Rapid Install wizard guides you through the Oracle Applications installation or upgrade process. You select whether you want to install Oracle Applications for the first time, perform an upgrade of an existing Oracle Applications instance, or configure an existing applications instance. You can install or upgrade up to 3 instances at one time. Oracle Rapid Install does not automatically select dependent products. You must select each product and any dependent products you want to install. The information you supply in the Oracle Rapid Install wizard is captured in a configuration file, which you store for use during the various stages of your installation or upgrade. For more information about how Oracle Rapid Install works, see *Installing Oracle Applications*.

This manual presents the installation tasks necessary for completing the installation of Oracle Configurator and the Oracle Configurator window add-on in Order Management, Telesales, iStore, Sales Online, or Order Capture. It also describes the installation tasks for installing and running Oracle Configurator Developer to create an Oracle Configurator window in the following environments:

- A custom web application using Oracle Configurator.
- A test environment launched from Oracle Configurator Developer.

For the most up-to-date information on installing and using Oracle Configurator (OC) successfully, see the *Oracle Configurator Release Notes* (A73283-04).

1.1 System Requirements

Oracle Configurator Window (Java Applet or DHTML)

Web		
Browser:	Netscape 4.06 or higher Internet Explorer 4.0 with Service Pack 2 or higher	
		Application Server
		Processor: Pentium II 300Mhz +
		Memory: 512Mb minimum
		Disk Space: 2GB or higher
		OS: Windows NT 4.0
		Internet Server
		Oracle Apache 1.3.9
		Data Server
		Oracle8*i* Enterprise Edition

Note: Any browser running an Oracle Configurator DHTML window must be set to display and use Javascript and Cascading Stylesheets, and must enable cookies. These requirements are met by Netscape 4.06 or later and Internet Explorer 4.0 with Service Pack 2 or later.

Oracle Configurator Developer

Development Workstation		Enterprise Data Server
Processor:	Pentium 300Mhz or equivalent	Oracle Enterprise Edition 8i
Memory:	256Mb minimum 512Mb or higher recommended	
Disk Space:	40Mb Free	

Development Workstation		Enterprise Data Server
OS:	Windows 95/98 and Windows NT 4.0	
DBMS:	Oracle Client 8.0.6	

1.2 Supported Platforms

The Oracle Configurator Developer is supported on Windows NT (Intel) only. The Oracle Configurator schema and the Oracle Configurator window are supported on all of the same platforms as Oracle Applications including the following:

- IBM RS 6000 (AIX)
- Compaq Tru64 UNIX
- HP 9000 Series HP-UX
- Windows NT (Intel)
- Sun Solaris xd86 (Intel)
- Sun Solaris (SPARC)
- Linux

1.3 Required Patches

After you install Oracle Applications, you must run `adpatch` to apply the latest patches to your Oracle Applications Release 11*i* environment. Contact Oracle Support or Metalink at http://metalink.oracle.com for the latest required `adpatch`. Log in to Metalink and click Technical Libraries > Alerts for the latest information regarding patches. See *Maintaining Oracle Applications* for information about applying these patches.

1.4 Completing the Oracle Configurator Installation

Installing Oracle Configurator with Oracle Applications by running Oracle Rapid Install does ***not*** complete the Oracle Configurator installation. There are additional steps you must complete before you can run Oracle Configurator Developer or an Oracle Configurator window from one of the Oracle Applications. Once you have installed or upgraded Oracle Configurator with Oracle Applications, you must:

1. Log into Oracle Applications as the system administrator.

2. Modify the database configuration file parameters for the database instance. See Section 1.4.1, "Modify Database Configuration File Parameters" on page 1-5.
3. Set profile options that enable you to run the Oracle Configurator window within Oracle Applications. See Section 1.4.2, "Set Profile Options" on page 1-5.
4. Configure the UI Servlet on your Internet server. See Section 2.1, "Configuring Apache and JServ" on page 2-1.
5. Load balance the Apache servlet to support multiple simultaneous users of the Oracle Configurator window. See Section 2.2, "Load Balancing" on page 2-8.
6. Set up your Oracle Applications responsibilities and users. See Section 1.4.3, "Set up Oracle Applications Responsibilities and Users" on page 1-6.
7. If you are not installing Oracle Configurator Developer to develop, maintain, or customize the Oracle Configurator window, complete your installation by verifying and testing your Apache configuration. See Section 1.4.4, "Verify and Test Apache Configuration" on page 1-6.

   If you are installing Oracle Configurator Developer to develop, maintain, or customize the Oracle Configurator window, complete the following additional steps.
8. Install Oracle Configurator Developer to develop, maintain, or customize the Oracle Configurator window. See Section 4.2, "The Oracle Configurator Developer Installation" on page 4-2.
9. Set up your server machine and establish Oracle Configurator Developer users and responsibilities. See Section 4.3.1, "Oracle Configurator Developer Server Machine" on page 4-4.
10. Set up your client machine. See Section 4.3.2, "Oracle Configurator Developer Client Machine" on page 4-6.
11. Establish data connectivity between the client machine and the Oracle Configurator schema on the server machine. See Section 4.3.2.4, "Enable the Client for Database Connectivity" on page 4-8.
12. Set parameters in the `spx.ini` file on the client machine to run Oracle Configurator Developer and the test environment you want. See Section 4.3.2.6, "Set Parameters in the spx.ini File for Development and Testing" on page 4-11 and the *Oracle Configurator and SellingPoint Administration Guide* for more information about these parameters.

1.4.1 Modify Database Configuration File Parameters

When you install or upgrade Oracle Applications, the database configuration file for the database instance (*database_instance_name*.dbc located in $FND_TOP/secure) must include the Oracle Configurator user and password as follows if you wish to run Oracle Configurator:

```
BATCH_VALIDATE_USER=APPS
BATCH_VALIDATE_PWD=APPS
```

Oracle Configurator Developer uses thick drivers. To use Oracle Configurator Developer you must comment out all the thin driver entries in the database configuration file.

1.4.2 Set Profile Options

In order to run the Oracle Configurator window within Oracle Applications such as Order Management, you must set a profile option that allows the hosting application to find the UI Manager URL. The UI Manager URL is the location where the configurator servlet resides. This URL is set up by the installer of the servlet.

You can set or view this profile option in Oracle Order Management or Oracle Bills of Material.

Note: This profile setting is not required for installations of an Oracle Configurator window running in a custom web application. The developer of the hosting application for the Oracle Configurator window will need to specify the URL of the configurator servlet, and then post the initialization message to that URL.

Profile Option	User	System Administrator				Requirements	
	User	User	Resp	App	Site	Required?	Default Value
BOM:CZ_UIMGR_URL	X	X	X	X	X	Required	No Default
Key:	X	You can update the profile option.					
	-	You can view the profile option value but you cannot change it.					
	0	You cannot view or change the profile option value.					

BOM:Configurator URL of UI Manager Indicates the location where the configurator servlet resides. The URL is the same as the Java system property `cz.uiservlet.url` defined for your UI servlet. All URLs in your profile options should be specified with the URL format: *machine_name.domain*:*port_number*; where *machine_name* is the name of the server machine, *domain* is your domain name, and *port_number* is the port where your service is running. The Apache server port is typically 880*n*. For example, `http://appsmachine.appsdomain:8800/servletvpath/oracle.apps.cz.servlet.UiServlet`.

1.4.3 Set up Oracle Applications Responsibilities and Users

Any Oracle Configurator user must also be defined as a database user in the Oracle8*i* Enterprise Edition server database running the Oracle Configurator schema. Additionally, the Oracle Configurator user must be assigned the Super User responsibility in the host Oracle Application.

1.4.4 Verify and Test Apache Configuration

Verifying Apache involves starting and testing it to be sure it is working properly.

To verify and test Apache Configuration:

1. Start the Apache Web Server using the appropriate command for your platform:

   `$COMMON_TOP/admin/scripts/adapcctl.sh start (UNIX)`

   or

   `adapcct1.cmd (Windows NT console)(`*`adapcctl.cmd`* `is located in the %APACHE_TOP% directory)`

2. Test Apache by executing the following command:

   `http://`*`local_host.domain`*`:`*`port_number`*

   For example, given http://app.future.com:8800, app is the local host name, future.com is the domain, and 8800 is the Apache listener port number.

   The Oracle Applications Release 11*i* Rapid Install portal displays.

   If you do not see this, check the error log located in the %APACHE_TOP/log directory.

3. Stop Apache using the following command:

```
$COMMON_TOP/admin/scripts/adapcctl.sh stop (UNIX)
```

or

```
adapcct1.cmd stop (Windows NT console)
```

1.5 Migrating From a Standalone Oracle SellingPoint Schema

This section includes information for migrating from a standalone Oracle SellingPoint schema (Oracle SellingPoint) to an Oracle Configurator schema of the Oracle Applications, Release 11*i* database.

Warning: This data migration is a "one-time" process. It is not to be re-run or used to refresh data in the Oracle Applications database.

1.5.1 Migration Prerequisites

Before you migrate from your current standalone schema, **be sure** you have satisfied the following prerequisites:

- Perform a clean Oracle Rapid Install of the Oracle Applications 11*i* database and ensure that the Oracle Configurator schema (CZ_SCHEMA) is **not** populated.
- Ensure that your source (standalone) schema has minimally been upgraded to schema 14c. If not, first run the appropriate upgrade script to upgrade your source schema (i.e., 13i or 14b) to schema 14c. See the *Oracle Configurator ReadMe* on the Oracle Configurator Developer compact disc for standalone schema upgrade instructions.
- If you want to exclude the migration of logically deleted data, PURGE your standalone schema instance. See the *Oracle Configurator and SellingPoint Administration Guide* for information about PURGE.
- Know the location of your source and target databases.
- Know the SID, hostname, and listener port number of the source database.

1.5.2 Migrate to Release 11*i*

Use the following procedure to migrate from your standalone Oracle Configurator schema (14c) to the Oracle Configurator schema (CZ_SCHEMA, 14d) in the Oracle Applications 11*i* database.

Warning: There is no data processing allowed during data migration, therefore, before migrating to Release 11*i* be sure you have met all of the prerequisites and that developers and end users are not connected to either your standalone or 11*i* CZ_SCHEMA, the Oracle Applications 11*i* database, Oracle Configurator Developer, or a test or mobile deployment of the Oracle SellingPoint application.

1. If you are currently running Oracle Applications release 10.7 or 11.0, you must first upgrade to the Oracle Applications 11*i* database. During this upgrade the CZ_SCHEMA remains empty.
2. Connect to your Oracle Applications 11*i* schema as *cz_user* where *cz_user* is granted RESOURCE privileges.

   For example:

   ```
   SQL> conn cz_user/cz_userpasssword@appssid
   ```

   where *cz_user* is the owner (DBOwner) of the Oracle Configurator schema, and *appssid* is the name for the Oracle8*i* Enterprise Edition instance on which the Oracle Applications 11*i* database is installed.
3. Run SQL*Plus in the *OC-scripts* directory; which is the directory containing the scripts provided in the `DBAdmin/Migration` folder on the Oracle Configurator Developer compact disc. Do not run any OC SQL*Plus scripts from SQL Worksheet.

   For example:

   a. Start SQL*Plus

   b. Go to File --> Open

   c. Navigate to *OC-scripts*

   d. Click Cancel
4. Run the `CZ_MIGRATE_SETUP.sql` script to setup your migration environment.

   For example:

   ```
   SQL> @cz_migrate_setup
   ```
5. Enter the username for the source (standalone) database schema.

6. Enter the password for the source (standalone) database schema.
7. Enter the name of the database link to be created.
8. If the source (standalone) schema is located on the same database instance as the target (11i) schema, enter the TNS service name to access the source (standalone) schema and press Enter for the two prompts that follow.
9. If the source (standalone) schema is not located on the same database instance as the target (11i) schema, enter the SID, hostname, and the listener port number for the remote database where the source schema is located.
10. The script creates the migration packages and a database link and compares the tables in the two schemas. The table comparison verifies whether or not:
    - like columns exist
    - datatype matches
    - nullabe fields match

    During the table comparison, you will receive mismatch messages due to the following known schema differences:
    - CZ_DB_SETTINGS table contains two additional fields, DBLinkName and MigrationStatus, in the 11*i* schema which are used for migration purposes.
    - CZ_XFR_ TABLES contains many additional fields in the 11*i* schema which are used for migration purposes.
    - CZ_DB_LOGS is not migrated.
    - The CHAR datatype used in the standalone schema is replaced with the VARCHAR2 datatype in the 11*i* schema.
    - The CZ_EXP_TMP_LINES table is in the 11*i* schema only.
    - All CZ_IMP_xx (import) tables are in the 11*i* schema only.
    - The PROPERTY_VALUE_NUM column of the CZ_PSNODE_ PROPCOMPAT_GENS table is in the 4.2.2 schema only.
    - All nullable fields in the standalone CZ_PRICING_STRUCTURE, CZ_ PSNODE_PROPCOMPAT_GENS, and CZ_RULE_FOLDERS tables are **not** nullable in the 11*i* schema.
    - The CZ_END_USERS table is migrated, however the users are not defined as DB users. To define end users as DB users, run the `EndUsers.sql` script. See the *Oracle Configurator and SellingPoint Administration Guide* for more information about this script.

> **Note:** Additionally, if a column datatype is the same in both schemas but the length is different, a truncation error may occur during migration.

A cz_migrate_setup.log file of this process is created.

11. Run the `CZ_RUN_MIGRATE.sql` script to migrate from your standalone Oracle Configurator schema to the Oracle Configurator schema (CZ_SCHEMA, 14d) in the Oracle Applications 11*i* database.

For example:

```
SQL> @cz_run_migrate
```

> **Note:** Fast Mode migration (Insert into xx as Select * from yy) can be used for most tables, however, a rollback segment error could occur for large volume tables if the rollback segment is not set high.

12. The script copies all of the tables from the source schema to the target schema and issues a completion message.

2

UI Servlet Considerations

In order to run an Oracle Configurator window (Java applet or DHTML), you must have the UI Servlet installed on your internet server. Installing the UI Servlet includes:

1. Using Oracle Rapid Install to install Oracle Applications Release 11*i* and Oracle Internet Application Server (iAS), which also installs the Apache Web Server and supporting software.
2. Configuring Apache and JServ to work with the Oracle Configurator UI Servlet, by verifying (and modifying, if necessary) the Apache configuration files.
3. Load-balancing your Apache Web Server.

You may want to consult the *Apache 1.3 User's Guide* and Apache web site (`java.apache.org`) when installing, configuring, or load balancing the Oracle Configurator servlet:

2.1 Configuring Apache and JServ

After you have installed Apache and its supporting software with Oracle Rapid Install, you must verify certain configuration files to ensure that they contain the correct parameters to work with the Oracle Configurator UI Servlet. This section contains a summary of the parameters you must have in your Apache configuration files.

- For details on Apache configuration files, consult the Apache documentation (at `http://java.apache.org`).
- You must log in as the owner of the Apache files in order to modify these files.
- The files to be verified are described in:
  - Verifying httpd.conf on page 2-3

- Verifying jserv.conf on page 2-4
- Verifying jserv.properties on page 2-5
- Verifying zone.properties on page 2-6
- Verifying Java System Properties on page 2-7

- See Java System Property Parameters for the UI Servlet on page 2-12 for descriptions of the individual parameters for the Oracle Configurator UI Server that are used in these files.

In this chapter, various textual placeholders are used. Table 2–1 lists the placeholders that may require some explanation (the names of the other placeholders should be self-explanatory).

Table 2–1 Textual Placeholders for Configuration Files

Placeholder	Example Values	Comment
ias_install	/d01/oracle/viscomn/util/apache/1.3.9/Apache	The directory in which you install iAS, using Oracle Rapid Install.
apache_install	*ias_install*/Apache (i.e., /d01/oracle/viscomn/util/apache/1.3.9/Apache/Apache)	The directory in which you install Apache, as part of iAS, using Oracle Rapid Install.
jserv_install	*ias_install*/Jserv (i.e., /d01/oracle/viscomn/util/apache/1.3.9/Apache/Jserv)	The directory in which you install JServ, as part of iAS, using Oracle Rapid Install.
hostname	www.mysite.com	The name of the host machine.
portnum	8802	The port number used by the Apache listener, which is specified by Port in httpd.conf.
html_vpath	OA_HTML	When running under Oracle Applications, the location pointed to by $OA_HTML.
	apache_install/htdocs/html	Otherwise, a directory located under /htdocs, which is specified by DocumentRoot in httpd.conf.

Table 2–1 Textual Placeholders for Configuration Files

Placeholder	Example Values	Comment
media_vpath	OA_MEDIA	When running under Oracle Applications, the location pointed to by $OA_MEDIA.
	apache_install/htdocs/media	Otherwise, a directory located under /htdocs, which is specified by DocumentRoot in httpd.conf.
servlet_vpath	configurator	A mounting location specified by ApJServMount in jserv.conf.

In the case of a placeholder that refers to an environment variable, the configuration file should contain the actual value of the environment variable, not the variable itself. For example, for a placeholder such as:

local_value_of_$OA_HTML

your configuration file should contain:

/d01/oracle/viscomn/html

rather than:

$OA_HTML

2.1.1 Verifying httpd.conf

By default. Oracle Rapid Install places httpd.conf in *apache_install*/conf.

Verify that the following parameters are set to point to the appropriate locations:

ServerRoot "*apache_install*"

DocumentRoot "*apache_install*/htdocs"

Alias /icons/ "*apache_install*/icons/"

Alias /OA_HTML/ "*local_value_of_$OA_HTML*/"

Alias /OA_MEDIA/ "*local_value_of_$OA_MEDIA*/"

Most of these parameters have corresponding <Directory> entries that should also be verified.

Note the use of trailing slash characters added to certain parameters.

You can ignore any settings of the aliases `/html/` and `/media/`. They are not used by the Oracle Configurator UI Servlet.

Verify that Apache's listening port is one that is not being used on the server machine:

```
Port portnum
```

At the very end of `httpd.conf`, verify that there is a line that points to the location of the JServ configuration file `jserv.conf`, which is located in *`jserv_install`*`/etc`. For example:

```
Include jserv_install/etc/jserv.conf
```

Verify that the Timeout parameter is set to a minimum of 1800 seconds (30 minutes).

```
Timeout 1800
```

2.1.2 Verifying jserv.conf

By default. Oracle Rapid Install places `jserv.conf` in *`jserv_install`*`/etc`.

Verify that the following parameter is set to point to *`apache_install`*:

```
LoadModule jserv_module apache_install/libexec/mod_jserv.so
```

Verify that the following parameter is set to point to *`jserv_install`*:

```
ApJServLogFile jserv_install/logs/mod_jserv.log
```

Verify that `ApJServProperties` is set to point to the location of the JServ properties file `jserv.properties`, which is located in *`jserv_install`*`/etc`:

```
ApJServProperties jserv_install/etc/jserv.properties
```

Verify that JServ's listening port is one that is not being used on the server machine (and different from the Port setting in `httpd.conf`):

```
ApJServDefaultPort portnum_jserv
```

Ensure that there is a valid mount point for the UI Servlet zone, and the virtual path for the zone:

```
ApJServMount /name_of_zone  /virtual_path_of_jserv
```

- This setting is required, in order to determine the *`servlet_vpath`*:

```
ApJServMount /configurator /root
```

- These settings can also be used:

```
ApJServMount /servlet /root
ApJServMount /servlets /root
```

2.1.3 Verifying jserv.properties

By default. Oracle Rapid Install places `jserv.properties` in *`jserv_install`*`/etc`.

Verify that certain Java system property values are being set for the UI Servlet, by being passed as arguments to the Java interpreter. See Verifying Java System Properties on page 2-7 for details.

Verify that a single entry like one of the following examples is added to set the maximum heap size. (We recommend at least 128MB; your configuration may differ.)

```
wrapper.bin.parameters=-mx209715200

wrapper.bin.parameters=-mx128m

wrapper.bin.parameters=-mx256m
```

Verify the classpath for the UI Servlet. Oracle Rapid Install may set additional classpath entries. Only the entries affecting the UI Servlet are described here.

If you are configuring Apache for using the Oracle Configurator window embedded in Oracle Applications Release 11*i*, verify the following `wrapper.classpath` values or modify the values to point to the appropriate locations:

```
wrapper.classpath=local_value_of_$OA_JAVA/apps.zip

wrapper.classpath=local_value_of_$OA_JAVA/jdbc111.zip

wrapper.classpath=local_value_of_$OA_JAVA/xmlparserv2.zip
```

If you are configuring Apache for using the Oracle Configurator window in a custom web application, outside Oracle Applications Release 11*i*, add the following `wrapper.classpath` values and modify the values to point to the appropriate locations:

```
wrapper.classpath=your_choice_of_path/cz3rdpty.jar
```

```
wrapper.classpath=your_choice_of_path/config.jar

wrapper.classpath=your_choice_of_path/jdbc111.zip
```

The following entries are required for the operation of the Apache web server and the JServ engine:

```
wrapper.classpath=jserv_install/libexec/ApacheJServ.jar

wrapper.classpath=ias_install/Jsdk/lib/jsdk.jar
```

If you have created Functional Companion classes, or "return_url" servlets, then you can install them in *jserv_install*/servlets, and add them to the UI Servlet's classpath:

```
wrapper.classpath=jserv_install/servlets
```

See the *Oracle Configurator Custom Web Deployment Guide* for information on return_url servlets. See the *Oracle Configuration Interface Object (CIO) Developer's Guide* for information on Functional Companions.

Verify that the `wrapper.env` section specifies Oracle Configurator's load library path:

```
wrapper.env=LD_LIBRARY_PATH=local_value_of_$CZ_TOP/bin
```

LD_LIBRARY_PATH must include the directory in which shared object files are available. For Oracle Configurator, these files are `libcz.so` and `libczjni.so`. After using Oracle Rapid Install, the shared object files may need to be relinked with `adrelink`.

Verify that the port number matches the ApJServDefaultPort that you set in `jserv.conf`:

```
port=portnum_jserv
```

Verify that the following parameters point to *jserv_install*

```
root.properties=jserv_install/etc/zone.properties

log.file=jserv_install/logs/jserv.log
```

2.1.4 Verifying zone.properties

By default. Oracle Rapid Install places `zone.properties` in *jserv_install*/etc.

Verify that the following parameter points to the directories that contain the classes to be reloaded when they are modified.

```
repositories=where_modified_classes_are_reloaded_from
```

Example:

```
repositories=jserv_install/servlets,local_value_of_$OA_JAVA/apps.zip
```

2.1.5 Verifying Java System Properties

Verify that certain Java system property values are being set for the UI Servlet, by being passed as arguments to the Java interpreter. These properties can be specified in either of two locations:

- in `jserv.properties`, using this syntax:

  ```
  wrapper.bin.parameters=-Dproperty_name=property_value
  ```

  Example:

  ```
  wrapper.bin.parameters=-Dcz.uiservlet.url=http://www.mysite.com:8802/configu
  rator/oracle.apps.cz.servlet.UiServlet
  ```

- in `zone.properties`, using this syntax:

  ```
  servlets.default.initArgs=-Dproperty_name=property_value
  ```

  Example:

  ```
  servlets.default.initArgs=-Dcz.uiservlet.url=http://www.mysite.com:8802/conf
  igurator/oracle.apps.cz.servlet.UiServlet
  ```

This syntax is like using the `-D` option of the Java interpreter:

```
java -Dproperty_name=property_value
```

If you specify properties in `zone.properties`, then using the `servlets.default.initArgs` parameters replaces the use of a separate .initArgs file.

By default, Oracle Rapid Install creates property settings in `jserv.properties`. Example 2–1 lists the default property settings created by Oracle Rapid Install.

You can also set other optional Java system properties to control the behavior of the UI Servlet, such as:

```
wrapper.bin.parameters=-Dcz.frameset.allocations.top=frameset_widths
```

```
wrapper.bin.parameters=-Dcz.scrolling.treeview=auto
```

Example 2–1 Default Java System Properties Created by Oracle Rapid Install

```
cz.uiservlet.templateurl=http://hostname:portnum/html_vpath/US/czFraNS.htm
cz.uiservlet.templateurl.ie=http://hostname:portnum/html_vpath/US/czFraIE.htm

cz.uiservlet.stylesheet.applet=http://hostname:portnum/html_vpath/czcmdcvt.xsl
cz.uiservlet.stylesheet.applet.client=http://hostname:portnum/html_vpath/czclient.xsl
cz.uiservlet.stylesheet.applet.server=http://hostname:portnum/html_vpath/czserver.xsl
cz.uiservlet.stylesheet.dhtml=http://hostname:portnum/html_vpath/czxml2js.xsl

cz.uiservlet.url=http://hostname:portnum/servlet_vpath/oracle.apps.cz.servlet.UiServlet

cz.uiservlet.proxyurl=http://hostname:portnum/servlet_vpath/oracle.apps.cz.servlet.Proxy
cz.uiservlet.proxyscript=http://hostname:portnum/html_vpath/czProxy.js
cz.uiservlet.sourcefile=http://hostname:portnum/html_vpath/czSource.htm
cz.html.source.treeview=http://hostname:portnum/html_vpath/cztree.htm
cz.html.source.display=http://hostname:portnum/html_vpath/czdisp.htm

cz.uiservlet.jdbcdriver=oracle.jdbc.driver.OracleDriver

cz.uiservlet.logfilename=jserv_install/logs/cz
cz.uimanager.logpath=jserv_install/logs
cz.uiservlet.applet.tmp=jserv_install/logs

FND_TOP=local value of $FND_TOP
```

2.2 Load Balancing

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

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

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

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

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

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

```
ulimit -n 1024
```

You should use JDK 1.1.8. This is the supported level of the JDK for the UI Servlet.

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

```
-mx209715200
or
-mx200m
```

Load Balancing Procedure

Here is a possible procedure to set up load balancing:

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

   ```
   ApJServManual on
   ```

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

   ```
   ApJServMount /configurator balance://set1/root
   ```

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

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

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

   Where:

`set1`	is the name of the set of JServ instances that you specified for `ApJServMount`, in step 2.
`host1jserv1` `host1jserv2`	are your arbitrary names for use in tracing in log files.
`ajpv12`	indicates the protocol used (Apache Jserv Protocol version 1.2)

hostname	is/are the host machine(s) on which you are running your JServ instances.
portnum_jserv1 *portnum_jserv2*	are the ports for JServ instances that you specified in `jserv.conf`, with `ApJServDefaultPort` *portnum_jserv*.

4. Set up your JServ instances to run standalone. To do this you should use a shell script. An example script is shown in Example 2–2 on page 2-11.

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

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

   ```
   bindaddress=*
   ```

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

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

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

   ```
   port=portnum_jserv
   ```

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

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

8. Start up Apache.

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

9. Start up your JServ instances by running the script in Example 2–2.

Configuration for Multiple Properties Files

If you are following step 7 on page 2-10, edit the configuration files as follows, for each JServ instance that you are creating.

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

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

  `root.properties=`*`jserv_install`*`/etc/zone2.properties`

- Point to a different log file, in a different log file directory:

  `log.file=`*`jserv_install`*`/logs2/jserv.log`

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

  `port=`*`portnum_jserv2`*

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

- Point to different log files:

  ```
  servlets.default.initArgs=cz.uiservlet.logfilename=jserv_install/logs2/cz
  servlets.default.initArgs=cz.uiservlet.applet.tmp=jserv_install/logs2/
  ```

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

Example 2–2 Shell Script for Starting JServ in Manual Mode (in Oracle Applications)

```
#!/bin/sh

ulimit -n 1024

properties=/d01/oracle/viscomn/util/apache/1.3.9/Apache/Jserv/etc/jserv.properties
# for 2nd JServ instance, change above line to:
# properties=/d01/oracle/viscomn/util/apache/1.3.9/Apache/Jserv/etc/jserv2.properties

log=/d01/oracle/viscomn/util/apache/1.3.9/Apache/Jserv/logs/jserv1.log
# for 2nd JServ instance, change above line to:
# log=/d01/oracle/viscomn/util/apache/1.3.9/Apache/Jserv/logs/jserv2.log
```

```
cmdlineargs="-Dcz.uimanager.logpath=/d01/oracle/viscomn/util/apache/1.3.9/Apache/Jserv/logs/ \
    -Dcz.html.source.treeview=http://www.mysite.com:8802/OA_HTML/cztree.htm \
    -Dcz.html.source.display=http://www.mysite.com:8802/OA_HTML/czdisp.htm \
    -mx256m"
# for 2nd JServ instance, change the "logpath" line above to:
# -Dcz.uimanager.logpath=/d01/oracle/viscomn/util/apache/1.3.9/Apache/Jserv/logs2/ \

echo $cmdlineargs >> $log

CLASSPATH=$CLASSPATH:/d01/oracle/viscomn/util/apache/1.3.9/Apache/Jsdk/lib/jsdk.jar
CLASSPATH=$CLASSPATH:/d01/oracle/viscomn/util/apache/1.3.9/Apache/Jserv/libexec/ApacheJServ.jar
CLASSPATH=$CLASSPATH:/d01/oracle/viscomn/java/apps.zip
CLASSPATH=$CLASSPATH:/d01/oracle/viscomn/java/jdbc111.zip
CLASSPATH=$CLASSPATH:/d01/oracle/viscomn/java/xmlparserv2.zip
CLASSPATH=$CLASSPATH:/d01/oracle/viscomn/util/apache/1.3.9/Apache/Jserv/servlets
echo $CLASSPATH >> $log

java $cmdlineargs -classpath $CLASSPATH org.apache.jserv.JServ $properties $1 2>> $log &
sleep 3
```

Note: The script in Example 2–2 is for the UI Servlet running inside Oracle Applications Release 11*i*. If you are running the UI Servlet outside Oracle Applications Release 11*i*, then you need to change the classpath, by adding cz3rdpty.jar and config.jar, and removing apps.zip and xmlparserv2.zip. The path to jdbc111.zip must also change. See Verifying jserv.properties on page 2-5 for details.

2.3 Java System Property Parameters for the UI Servlet

You can control certain behaviors of the UI Servlet by passing values for Java system properties. For Apache, these properties can be specified in `jserv.properties`, like this:

```
wrapper.bin.parameters=-Dproperty_name=property_definition
```

or in `zone.properties`, like this:

```
servlets.default.initArgs=-Dproperty_name=property_definition
```

The system properties for which you can set parameters are listed in Table 2–2. The columns "Applet" and "DHTML" indicate the type of Oracle Configurator window with which the property can be used. For additional details on the use of some of

these properties with the DHTML Oracle Configurator window, see the *Oracle Configurator Custom Web Deployment Guide*.

Table 2–2 Java System Properties for the UI Servlet

Property Name	Description	Applet	DHTML
Properties used by both Applet and DHTML windows:			
cz.uimanager.logpath	on page 2-18	Y	Y
cz.uiservlet.jdbcdriver	on page 2-17	Y	Y
cz.uiservlet.logfilename	on page 2-17	Y	Y
cz.uiservlet.name	on page 2-17	Y	Y
cz.uiservlet.url	on page 2-16	Y	Y
FND_TOP	on page 2-19	Y	Y
Properties used only by Applet window:			
cz.uiserver.media.folder	on page 2-15	Y	N
cz.uiservlet.applet.tmp	on page 2-18	Y	N
cz.uiservlet.stylesheet.applet	on page 2-15	Y	N
cz.uiservlet.stylesheet.applet.client	on page 2-15	Y	N
cz.uiservlet.stylesheet.applet.server	on page 2-15	Y	N
Properties used only by DHTML window:			
cz.activemodel	on page 2-14	Y	Y
cz.frameset.allocations.top	on page 2-19	N	Y
cz.html.source.display	on page 2-18	N	Y
cz.html.source.treeview	on page 2-18	N	Y
cz.scrolling.treeview	on page 2-19	N	Y
cz.uiservlet.proxyscript	on page 2-16	N	Y
cz.uiservlet.proxyurl	on page 2-16	N	Y
cz.uiservlet.sourcefile	on page 2-16	N	Y
cz.uiservlet.stylesheet.dhtml	on page 2-15	N	Y

Table 2–2 (Cont.) Java System Properties for the UI Servlet

Property Name	Description	Applet	DHTML
cz.uiservlet.templateurl	on page 2-14	N	Y
cz.uiservlet.templateurl.ie	on page 2-14	N	Y

cz.activemodel

Path to read and write logic files on the server. The Active Model Path comes from either cz.activemodel, cz.uimanager.logpath, or user.dir, in that order.

This property is only used by Oracle SellingPoint Configurator. If the UI Servlet is for the Oracle Configurator window running embedded within Oracle Applications, this parameter is not needed.

Syntax:

`cz.activemodel=`*`activemodel_dir/`*

Example for Windows NT:

```
cz.activemodel=D:\orant\OSP\Shared\ActiveModel\
```

cz.uiservlet.templateurl

URL of HTML template for DHTML client running in a Netscape browser.

Syntax:

`cz.uiservlet.templateurl=http://`*`hostname:portnum/html_vpath`*`/US/czFraNS.htm`

Example:

```
cz.uiservlet.templateurl=http://www.mysite.com:8802/OA_HTML/US/czFraNS.htm
```

cz.uiservlet.templateurl.ie

URL of HTML template for DHTML client running in a Microsoft Internet Explorer browser.

Syntax:

`cz.uiservlet.templateurl.ie=http://`*`hostname:portnum/html_vpath`*`/US/czFraIE.htm`

Example:

```
cz.uiservlet.templateurl.ie=http://www.mysite.com:8802/OA_HTML/US/czFraIE.htm
```

cz.uiservlet.stylesheet.applet

Syntax:

```
cz.uiservlet.stylesheet.applet=http://hostname:portnum/html_vpath/czcmdcvt.xsl
```

Example:

```
cz.uiservlet.stylesheet.applet=http://www.mysite.com:8802/OA_HTML/czcmdcvt.xsl
```

cz.uiservlet.stylesheet.applet.client

Syntax:

```
cz.uiservlet.stylesheet.applet.client=http://hostname:portnum/html_vpath/czclien
t.xsl
```

Example:

```
cz.uiservlet.stylesheet.applet.client=http://www.mysite.com:8802/OA_HTML/czclien
t.xsl
```

cz.uiservlet.stylesheet.applet.server

Syntax:

```
cz.uiservlet.stylesheet.applet.server=http://hostname:portnum/html_vpath/czserve
r.xsl
```

Example:

```
cz.uiservlet.stylesheet.applet.server=http://www.mysite.com:8802/OA_HTML/czserve
r.xsl
```

cz.uiservlet.stylesheet.dhtml

Syntax:

```
cz.uiservlet.stylesheet.dhtml=http://hostname:portnum/html_vpath/czxml2js.xsl
```

Example:

```
cz.uiservlet.stylesheet.dhtml=http://www.mysite.com:8802/OA_HTML/czxml2js.xsl
```

cz.uiserver.media.folder

Location of the directory containing image media for the UI Servlet. The default value is /OA_MEDIA/, which is used if this property is not set.

Syntax:

cz.uiserver.media.folder=http://*hostname*:*portnum/dir_path/*

Example:

cz.uiserver.media.folder=http://www.mysite.com:8802/OA_MEDIA/alt/

cz.uiservlet.url

URL of the Oracle Configurator UI Servlet. Installed in config.jar.

Syntax:

cz.uiservlet.url=http://*hostname:portnum/servlet_vpath*/oracle.apps.cz.servlet.UiServlet

Example:

cz.uiservlet.url=http://www.mysite.com:8802/configurator/oracle.apps.cz.servlet.UiServlet

cz.uiservlet.proxyurl

URL of the OC Proxy class. Installed in config.jar.

Syntax:

cz.uiservlet.proxyurl=http://*hostname:portnum/servlet_vpath*/oracle.apps.cz.servlet.Proxy

Example:

cz.uiservlet.proxyurl=http://www.mysite.com:8802/configurator/oracle.apps.cz.servlet.Proxy

cz.uiservlet.proxyscript

URL of JavaScript source file for the proxy frame.

Syntax:

cz.uiservlet.proxyscript=http://*hostname:portnum/html_vpath*/czProxy.js

Example:

cz.uiservlet.proxyscript=http://www.mysite.com:8802/OA_HTML/czProxy.js

cz.uiservlet.sourcefile

Specifies the HTML template file that produces the Source Frame in the Oracle Configurator window.

Syntax:

```
cz.uiservlet.sourcefile=http://hostname:portnum/html_vpath/czSource.htm
```

Example:

```
cz.uiservlet.sourcefile=http://www.mysite.com:8802/OA_HTML/czSource.htm
```

cz.uiservlet.jdbcdriver

JDBC driver for connecting the UI Servlet to a database.

Syntax:

```
cz.uiservlet.jdbcdriver=driver_class
```

Example:

```
cz.uiservlet.jdbcdriver=oracle.jdbc.driver.OracleDriver
```

cz.uiservlet.logfilename

The path into which the UI Servlet will write logging files. Do not specify a specific file name. If this parameter is omitted, no logging files will be written.

Syntax:

```
cz.uiservlet.logfilename=logging_dir
```

Examples for Solaris:

```
cz.uiservlet.logfilename=jserv_install/logs/cz
```

```
cz.uiservlet.logfilename=/d01/oracle/viscomn/util/apache/1.3.9/Apache/Jserv/logs/cz
```

Example for Windows NT:

```
cz.uiservlet.logfilename=D:\orant\OSP\OSP\log
```

cz.uiservlet.name

Provides a name for use in tracing your servlet in log files. The name is inserted into the name of the log files written to the location specified by cz.uiservlet.logfilename.

Syntax:

```
cz.uiservlet.name=string
```

Example:

```
cz.uiservlet.name=smith
```

cz.html.source.treeview

Specifies the HTML template file that produces the tree view in the Oracle Configurator window.

Syntax:

```
cz.html.source.treeview=http://hostname:portnum/html_vpath/cztree.htm
```

Example:

```
cz.html.source.treeview=http://www.mysite.com:8802/OA_HTML/cztree.htm
```

cz.html.source.display

Specifies the HTML template file that produces the primary display view in the Oracle Configurator window.

Syntax:

```
cz.html.source.display=http://hostname:portnum/html_vpath/czdisp.htm
```

Example:

```
cz.html.source.display=http://www.mysite.com:8802/OA_HTML/czdisp.htm
```

cz.uimanager.logpath

The path into which the UI Server will write a log file when it handles an exception and exits.

Syntax:

```
cz.uimanager.logpath=logging_dir/
```

Note the need for a trailing slash character added to the value.

Example for Solaris:

```
cz.uimanager.logpath=jserv_install/logs/
```

Example for Windows NT:

```
cz.uiservlet.logfilename=D:\orant\OSP\OSP\log\
```

cz.uiservlet.applet.tmp

Syntax:

```
cz.uiservlet.applet.tmp=logging_dir/
```

Note the need for a trailing slash character added to the value.

Example for Solaris:

```
cz.uiservlet.applet.tmp=jserv_install/logs/
```

Example for Windows NT:

```
cz.uiservlet.applet.tmp=D:\orant\OSP\OSP\log\
```

cz.frameset.allocations.top

Controls the relative sizes of the tree view and display frames in the Content Frame of the Oracle Configurator window. The default value is `30%,*`.

Syntax:

```
cz.frameset.allocations.top=frameset_widths
```

Example:

```
cz.frameset.allocations.top=25%,*
```

cz.scrolling.treeview

Controls whether there is scrolling in the tree view frame in the configuration window. The default value is `auto`.

Syntax:

```
cz.scrolling.treeview=[auto|yes|no]
```

Example:

```
cz.scrolling.treeview=yes
```

FND_TOP

Required for locating the DBC file used for database connectivity. To determine the value to use, enter the command

```
% echo $FND_TOP
```

in a command shell having the desired Oracle Applications environment.

Syntax:

```
FND_TOP=local value of $FND_TOP
```

Example:

```
FND_TOP=/d01/oracle/visappl/fnd/11.5.0
```

3

Troubleshooting Servlet Installation

This section provides suggestions for resolving problems that may arise when installing the Oracle Configurator servlet. This installation is described in Chapter 2, "UI Servlet Considerations".

3.1 Miscellaneous

- Make sure that you have set your virtual paths correctly.
- Make sure that your executable path includes the Shared Object files, (.so or .dll) . A symptom of this problem might be an error message starting with a line like this one:

```
java.lang.UnsatisfiedLinkError: no ocijdbc8 in shared library path
```

3.2 Checking the Operation of the Apache Internet Server

What you are checking

Does your Apache internet server respond at all?

The test

1. Compile the code in Example 3–1 into the file Hello.class in your servlets directory.
2. In a web browser, invoke this URL:

`http://hostname:portnum/servlet_vpath/Hello`

where *`yourserver`* is the server that you have installed on, and *`portnum`* is the port number configured for the HTTP listener. For example:

```
http://www.mysite.com:10130/servlets/Hello
```

3. The browser should display the HTML message written by your test class.

Example 3–1 Hello.java Test Class

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

/**
 * This is a simple example of an HTTP Servlet.  It responds to the GET
 * and HEAD methods of the HTTP protocol.
 */
public class Hello extends HttpServlet
{
    /**
     * Handle the GET and HEAD methods by building a simple web page.
     * HEAD is just like GET, except that the server returns only the
     * headers (including content length) not the body we write.
     */
    public void doGet (HttpServletRequest request,
                       HttpServletResponse response)
        throws ServletException, IOException
        {
            PrintWriter out;
            String title = "Example Apache JServ Servlet";

            // set content type and other response header fields first
            response.setContentType("text/html");

            // then write the data of the response
            out = response.getWriter();

            out.println("<HTML><HEAD><TITLE>");
            out.println(title);
            out.println("</TITLE></HEAD><BODY bgcolor=\"#FFFFFF\">");
            out.println("<H1>" + title + "</H1>");
            out.println("<H2> Congratulations, ApacheJServ is working!<br>");
            out.println("</BODY></HTML>");
            out.close();
        }
}
```

If the test fails

Check that the internet server installation, the port number, and that the server you chose is available on the right network. Sometimes this can be a server name problem. To get around that problem, refer to the server by its IP address.

3.3 Checking the Response of the UI Servlet

What you are checking

Does the UiServlet respond to a test message?

The test

Invoke this URL in a web browser:

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

where *hostname* is the name of your internet server, *portnum* is the port number for your web listener, *servlet_vpath* is the virtual path that you set up when you installed the servlet, and *test_string* is an unquoted character string. Do not include any whitespace characters in *test_string*. If the servlet is installed correctly and running, it should produce an HTML page that prints the results listed below for different values of *test_string*.

Value of *test_string*	Result printed in HTML page
`version`	The current build version of Oracle Configurator and the expected version for the Oracle Configurator schema.
`host`	The host name of the web server, and the listener port used. These are the values of *hostname* and *portnum* specified in your test URL.
any_other_string	The string that you entered.

Example:

```
http://www.mysite.com:8802/configurator/oracle.apps.cz.servlet.UiServlet?test=version
```

produces a result like the following:

```
Using configuration software build: 11.5.2.14.33
```

```
Expecting schema: 14d
```

Example:

```
http://www.mysite.com:8802/configurator/oracle.apps.cz.servlet.UiServlet?test=he
llo_world
```

produces the following result:

```
hello_world
```

If the test fails

In your internet server, turn on the maximum amount of logging. Look in the log file to see which classes it loads, and from which JAR files. Towards the end of this file, there may be a message that some class failed to load. It is probably the case that there is a JAR file in the list that is not in the path specified or that there was an error in specifying its name.

3.4 Restarting the UI Server

What you are checking

If you have made changes to the Oracle Configurator Model, Configuration Rules, or User Interface in Configurator Developer, is the UiServlet working from the latest data from the database?

The test

Invoke this URL in a web browser:

```
http://hostname:portnum/servlet_vpath/oracle.apps.cz.servlet.UiServlet?killAndRe
startServer=true
```

where *hostname* is the name of your internet server, *portnum* is the port number for your web listener, *virtual_path* is the virtual path that you set up when you installed the servlet, and *test_string* is an arbitrary unquoted character string.

This forces the UI Servlet to reread the latest data from the Oracle Configurator schema in the Oracle Applications database, obtaining any changes you have made to the Model, Rules, or User Interface in Configurator Developer.

The response in your browser should be:

```
The UI server has been restarted.
```

Warning: **Using this servlet parameter kills any session that is using your UI Server instance. It is intended for use in development, or demos, not for end user sessions.**

3.5 Checking Your Model in the Oracle Configurator Window

What you are checking

Does your configuration Model behave as you expect in the Oracle Configurator window?

Note: You can also test your Model through Configurator Developer, using the Test/Debug module. See the *Oracle Configurator Developer User's Guide* for details.

The test

You can test the behavior of the DHTML Oracle Configurator window by creating an HTML test page that substitutes for your host application.

1. Create an HTML test page that posts the OC initialization message to the UI Servlet.

   See the chapter on session initialization in the *Oracle Configurator Custom Web Deployment Guide* for an explanation of the OC initialization message.

   See Example 3–2, and the *Oracle Configurator Custom Web Deployment Guide* for examples of simple test pages.

Example 3–2 HTML Test Page for Invoking the DHTML Oracle Configurator window

```
<html>
<head>
<title>Minimal Configurator Test</title>
</head>
<body>
<form action="http://www.mysite.com:8802//oracle.apps.cz.servlet.UiServlet"
method="post">
<input type="hidden" name="XMLmsg" value=
'<initialize>
  <param name="two_task">vis</param>
  <param name="gwyuid">applsyspub/pub</param>
```

```
  <param name="fndnam">apps</param>
  <param name="user">apps</param>
  <param name="pwd">apps</param>
  <param name="ui_type">DHTML</param>
  <param name="ui_def_id">3120</param>
</initialize>'>

<p>Click the button to configure the model...
<input type="submit" value="Configure">
</form>
</body>
</html>
```

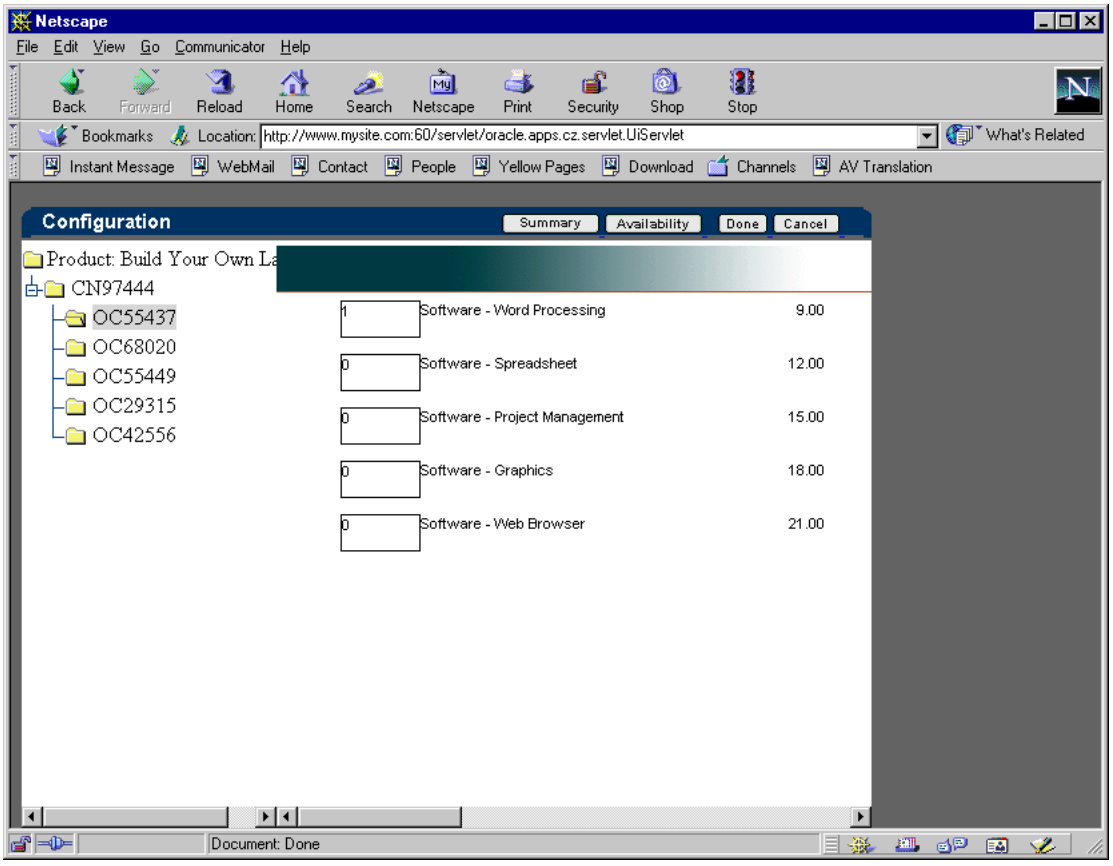
2. Insure that you have the necessary database connectivity, and that your UI Servlet is installed and configured correctly. See Chapter 2, "UI Servlet Considerations".

3. Test the Oracle Configurator window by opening the HTML test page.

   Your default web browser opens, displaying the current Model using the selected User Interface, in a frame built with the Oracle Configurator HTML templates. If you used Example 3–2, click the button to produce the User Interface.

   See Figure 3–1 on page 3-7 for a general example of the appearance of the DHTML Oracle Configurator window.

Figure 3–1 A Model, in the Oracle Configurator window



4

Installing Oracle Configurator Developer

Oracle Configurator Developer is the development and maintenance environment used to create configuration models and custom Oracle runtime configurator windows. Before installing Oracle Configurator Developer, **be sure** you have satisfied the following prerequisites:

- If you have a prior version of Oracle Configurator, uninstall all components before installing the new version. If you are reinstalling Oracle Configurator and have customized DHTML template files in %ORACLE_HOME%\OSP\WebUI\ and copied them to your web server, be sure to back up your files and replace them after the reinstall, as the Oracle Configurator install program will overwrite these files.
- If you are running Windows 95/98, install DCOM98 from the Oracle Configurator Developer compact disc.
- Ensure that you have release 10.7, 11.01, 11.0.2, or 11.0.3 of Oracle Applications installed and configured.
- For all development installations, install Oracle8*i* Enterprise Edition (or higher) on the server machine and Oracle Client 8.0.6 on the client machine.
- For web deployment or development testing using the DHTML window or JAVA applet, install a supported internet server and the UI servlet. See Chapter 2, "UI Servlet Considerations".

4.1 The Oracle Configurator Developer Software

This Oracle Configurator Developer Release 11*i* includes the following software:

Product	Revision	Level	License
Oracle Configurator Developer Installation	11*i*	Production	Yes
Oracle Configurator Developer	11*i*	Production	Yes
Oracle Configurator Documentation (`doc` folder)	11*i*	Production	No
SQL *Plus Scripts (`DBAdmin` folder)	11*i*	NA	No
Adobe Acrobat Reader	4.0	NA	No
DCOM98	1.0	NA	No
MDAC	2.1	NA	No
MS Java VM (only required for the Oracle SellingPoint Application)	Build 3181	NA	No

The folders called `DBAdmin` and `doc` that are included in the Oracle Configurator Installation are described below:

- The `DBAdmin` folder contains SQL create scripts for creating, configuring, and upgrading an Oracle Configurator schema for a custom web application, and scripts for setting up and running data import and export between Oracle Applications and Oracle Configurator. See the *Oracle Configurator and SellingPoint Administration Guide* for information about these scripts.
- The `doc` folder contains the following book files:

  Oracle Configurator Developer User's Guide.pdf
  Oracle Configurator Custom Web Deployment Guide.pdf
  Oracle Configurator Installation Guide.pdf
  Oracle Configurator and SellingPoint Administration Guide.pdf
  Oracle Configuration Interface Object (CIO) Developer's Guide.pdf

  You must have Adobe Acrobat Reader to view these books. Adobe Acrobat Reader is available on the Oracle Configurator Developer compact disc, by opening the file ar40eng.exe.

4.2 The Oracle Configurator Developer Installation

Follow these steps to install Oracle Configurator Developer from the Oracle Configurator Developer compact disc:

1. Exit all other applications.
2. Insert the Oracle Configurator Developer compact disc.
3. From the Start menu, run:

   `D:\OCSetup.EXE`
   where `D` is your CD-ROM drive.
4. Specify the destination location for the installed components or accept the default location (`%ORACLE_HOME%\OSP\`).
5. A dialog box appears with the following option:
   - Oracle Configurator Developer
6. Select Oracle Configurator Developer.

   Note: Selecting **Oracle Configurator Developer** automatically installs the Oracle SellingPoint application as a test environment.

7. After the component installation is complete, a dialog box appears asking if you would like to install the Microsoft Data Access Components (MDAC). If you do not already have MDAC 2.1 installed, choose "Yes".
8. Once the installation is complete on Windows 95/98, a confirmation dialog appears. If you have installed MDAC, your Windows 95/98 machine will reboot. You should reboot after installing any Oracle Configurator component on a Windows 95/98 machine. No reboot is necessary on Windows NT.
9. An `spx.ini` file with default settings was installed. This file may need to be edited for your site. Any previously installed `spx.ini` file was not overwritten by this install, but renamed `spx.ini.bak`. See Section 4.3, "Oracle Configurator Developer Client/Server Setup" and the *Oracle Configurator and SellingPoint Administration Guide* for details on setting parameters in the `spx.ini` file.

To run Oracle Configurator Developer and an Oracle Configurator window, see Section 4.4, "Test Your Oracle Configurator Developer Installation" on page 4-14.

4.3 Oracle Configurator Developer Client/Server Setup

Running Oracle Configurator and Oracle Configurator Developer involves servers that are both hardware devices running processes shared by client machines, as well as the shared server processes themselves.

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

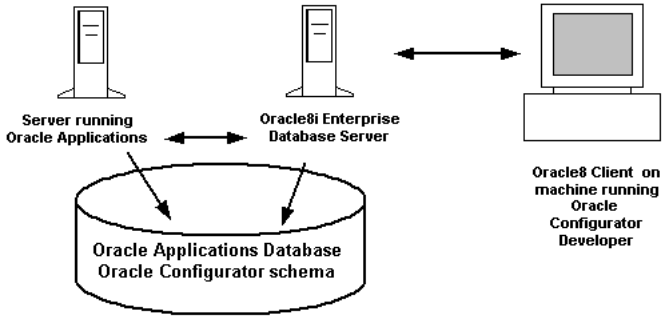
4.3.1 Oracle Configurator Developer Server Machine

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

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

In addition, Oracle Configurator requires an application server to broker the processes and http connection. Figure 4–1, "Server Configuration for OC Server" illustrates how Oracle Applications Forms and the Oracle Applications database containing the Oracle Configurator schema can be located on separate machines.

Figure 4–1 Server Configuration for OC Server



In general, the Oracle Applications database with the Oracle Configurator schema is installed on a machine that acts as the server for networked clients running Oracle Configurator Developer and test Oracle Configurator windows. This would be the case for any installation of Oracle Configurator. Oracle Applications Forms and the

Oracle Applications database containing the Oracle Configurator schema can be located on separate machines. The server machine where the Oracle Configurator schema is installed must also have Oracle8*i* Enterprise Edition installed and the client machine(s) must have Oracle Client 8.0.6 installed.

The server database setup includes the following tasks:

- Install Oracle8*i* Enterprise Edition first
- Install the Oracle Applications database (*appssid* and *apps*)
- Create Users and Responsibilities (see Section 4.3.1.1, "Establish Users and Responsibilities" on page 4-5)
- Set Oracle Configurator Developer DB settings

For information about installing the Oracle Applications database, see *Installing Oracle Applications*. For detailed information about the Oracle Configurator schema or specific information about Oracle Configurator schema DB settings, see *Oracle Configurator and SellingPoint Administration Guide*.

4.3.1.1 Establish Users and Responsibilities

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

The Oracle Configurator schema DBOwner can log in to the Oracle Configurator Developer. A non-DBOwner user of Oracle Configurator Developer need only have an Oracle8*i* Enterprise Edition account with SPX_USER role privileges to log into Configurator Developer. The DBOwner is to the Oracle Configurator schema as FNDNAM is to the rest of Oracle Applications. You may also log in to Oracle Configurator Developer as an Oracle Applications FND user. To do this, however, the datasource description in the spx.ini file on the client machine must provide additional gateway parameters (gwyuid and gwypass) that specify the Oracle public gateway login information. See the *Oracle Configurator and SellingPoint Administration Guide* for more information on these parameters and the SPX_USER role.

4.3.1.2 Accessing Oracle Configurator Windows

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

4.3.2 Oracle Configurator Developer Client Machine

Installing Oracle Configurator for an implementation, test, or maintenance environment consists of installing Oracle Configurator Developer (which includes the Oracle Configurator window). Oracle Configurator Developer runs on a client machine with Oracle Client 8.0.6 for:

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

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

- The Oracle Configurator schema is running on the server.
- Oracle Client 8.0.6 is installed on the client machine that is running Oracle Configurator Developer. See Section 4.3.2.2, "Oracle Client 8.0.6 Installation" on page 4-7.
- The client machine is configured to connect to the Oracle Configurator schema on the server machine (Oracle Net8 Easy Config). See Section 4.3.2.4, "Enable the Client for Database Connectivity" on page 4-8.
- A datasource name (DSN) for the Oracle Configurator schema on the server machine is defined in ODBC Administrator on the client machine. See Section 4.3.2.5, "Create DSNs and DBOwners" on page 4-10.
- The user logged into the client machine is a user defined in Oracle8*i* Enterprise Edition running the Oracle Configurator schema on the server.
- The user logged into the client machine is a user defined in the Oracle Applications database.
- The `spx.ini` file is edited to include the correct DSN and DBOwner for the Oracle Configurator schema on the server. See Section 4.3.2.6, "Set Parameters in the spx.ini File for Development and Testing" on page 4-11.

4.3.2.1 Oracle Configurator Window Client Requirements

When running a test instance of the Oracle Configurator window (JAVA applet or DHTML) from within Oracle Configurator Developer, the requirements for running Configurator Developer must be satisfied. In addition, the URL for the DHTML and Java applet Servlet UI and JDBC thin client parameter (for DHTML only) must be

specified in the `spx.ini` file (see Table 4–1, "Spx.ini Parameters for Development and Testing" on page 4-12).

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

4.3.2.2 Oracle Client 8.0.6 Installation

Oracle Client 8.0.6 allows the Oracle Configurator Developer 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 8.0.6 installed if they access the Oracle8*i* Enterprise Edition Server database.

Oracle Client 8.0.6 is used to access the Oracle Configurator schema from a client machine.

4.3.2.3 Set Up Oracle Configurator Developer

Once your site has Oracle8*i* Enterprise Edition Server and Oracle Client 8.0.6 installed and you have installed Oracle Applications and the Oracle Configurator schema, you can install and set up Oracle Configurator Developer for your users.

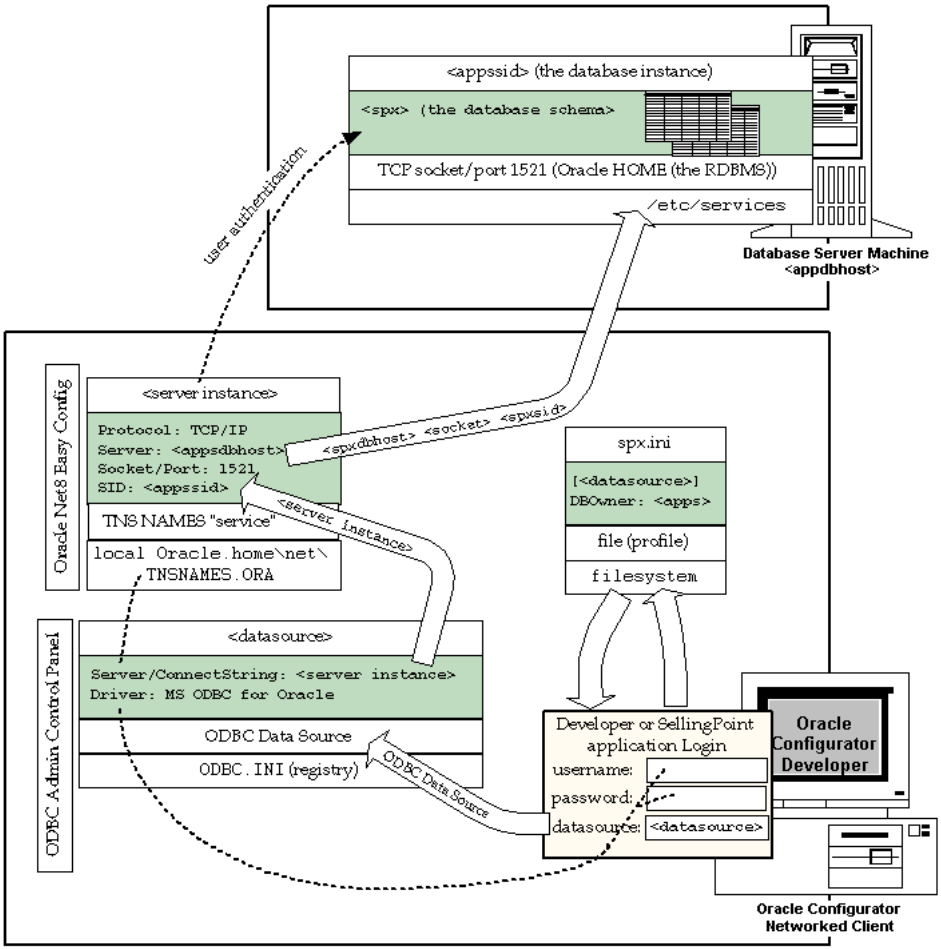
The tasks required to set up Oracle Configurator Developer for your users are:

- Enable the Client for Database Connectivity
- Create DSNs and DBOwners
- Set Parameters in the spx.ini File for Development and Testing
- Test Your Oracle Configurator Developer Installation

The Oracle Applications System Administrator can customize the DBOwner, access roles, and privileges.

Figure 4–2, "Client/Server Data Communication Architecture" illustrates an overview of the architecture of client/server data communication you set up with these tasks.

Figure 4–2 Client/Server Data Communication Architecture



4.3.2.4 Enable the Client for Database Connectivity

After installing Oracle Configurator Developer, you must establish data connectivity with the Oracle Configurator schema on the server machine and set parameters accordingly in the `spx.ini` file on the client machine. See the *Oracle Configurator Installation Guide* for information about installing and setting up Oracle Configurator Developer and client requirements.

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

When running Oracle Configurator Developer or a test configurator on a client machine, the client machine needs data connectivity to the Oracle Configurator schema on the server machine. The client machine must be running Oracle Client 8.0.6. To establish data connectivity on the client machine, you need to know the following parameters:

- the name of the server database instance or system identifier (SID) where the Oracle Configurator schema is located (*appssid*)
- the name of the physical server machine where *appssid* is located (*appsdbhost*)
- the port of the physical server machine through which the client connects to the Oracle Configurator schema (*port*)
- the username/password of the DBOwner of the Oracle Configurator schema schema to which the client is connecting.

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

1. In Windows 95/98 or Windows NT 4.0, select Start > Programs > Oracle for Windows > Oracle Net8 Easy Config. (If you do not have this option, you have an outdated version of Oracle, or no client software). Install Oracle Net8 Easy Config using the Oracle Installer.
2. Select Add New Service and enter the new service name**.** This is the name of the server database instance containing the Oracle Configurator schema you will be connecting to. 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 and the Port Number. The host name is the name of the physical server machine where the Oracle Configurator schema is located. Make a note of the Port Number (default is 1521) in case it is needed for future reference. Click Next.
5. Enter the Database SID. This is the name of the server database instance containing the Oracle Configurator schema you will be connecting to, the same name as the New Service you just added. Click Next.
6. Test the connection by clicking on the Test Service button.

7. Enter the Username and Password for the owner of the Oracle Configurator schema you've been setting up data connectivity to. Click Test.
8. After clicking on the Test button, the results display. Click Done when the result shows the test has completed successfully.
9. Click Finish to save your service configuration and exit Oracle Net8 Easy Config.

4.3.2.5 Create DSNs and DBOwners

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

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

For an Oracle Client 8.0.6 database, use the ODBC driver Microsoft ODBC Driver for Oracle.

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

1. In Windows 95/98 or Windows NT 4.0, select Start > Settings > Control Panel and open ODBC Data Sources (32bit). This opens the ODBC Data Source Administrator.
2. Select the System DSN tab.
3. Click Add... This opens the Create New Data Source dialog.
4. Select "Microsoft ODBC for Oracle" (version 2.573.xxxx.xx) for setting a server DSN. Click Finish.
5. The Microsoft ODBC for Oracle Setup dialog appears. Enter the name of the database you want to access (including the extension, i.e., .db or .odb) in the Data Source Name field.
6. Optionally, enter a description of the database driver that the data source connects to in the Description field.
7. Optionally, enter your database user ID in the User Name field.

8. Enter the Service Name for the Oracle Server engine in the Server field. The Service Name identifies the Oracle Database instance (*ocsid*) that you want to access.

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

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

   Option: Translation

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

   Option: Performance

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

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

   Option: Customization

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

10. Click OK to add the data source.

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

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

4.3.2.6 Set Parameters in the `spx.ini` File for Development and Testing

Parameters in the `spx.ini` file determine connectivity and product behavior for development and testing. See the *Oracle Configurator and SellingPoint Administration Guide* for more information about these and deployment parameters. Parameters

affecting Oracle Configurator Developer must be set for development and test. No `spx.ini` file is required for deployments of an Oracle Configurator window.

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

- Oracle Configurator Developer
- Test instances of an Oracle Configurator window from within Oracle Configurator Developer (Test/Debug)

Throughout this section, references to the test configurator mean test instances of an Oracle Configurator window launched from within Oracle Configurator Developer (Test/Debug).

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

Minimally, you must edit the `spx.ini` file and update the [DSN] entries by adding the ODBC DSN(s) you created for your Oracle Configurator schema. The entries then appear in the Oracle Configurator Developer list of available data sources when you log in to Oracle Configurator Developer. You must create the Oracle Configurator schema DSN yourself, following the instructions in Section 4.3.2.5, "Create DSNs and DBOwners" on page 4-10; the `spx.ini` entries will not work until you create the DSN.

You must update the following spx.ini entries to use Oracle Configurator Developer and test configurator.

Table 4–1 Spx.ini Parameters for Development and Testing

Section	Parameter	Description
Merlin	DBOwner	Specifies the default username of the owner of the Oracle Configurator schema that the `spx.ini` file accesses when starting up Oracle Configurator Developer. Users log into Oracle Configurator Developer with this schema name and the password.

Table 4–1 Spx.ini Parameters for Development and Testing

Section	Parameter	Description
DSN	DBOwner	The DSN of the Oracle Configurator schema used with Oracle Configurator Developer must be listed here. Furthermore, a section must be added for the DSN of the server DBOwner by which users will access the server Oracle Configurator schema. There may be several DSNs listed. You may have a DSN listed for an Oracle Applications database available for use by Oracle Configurator Developer to test user interfaces launching a DHTML window or a Java applet. In order to use the Oracle Applications login functionality, the value for DBowner here should be the same as the FNDNAM parameter value in the Oracle Applications environment file.

	JdbcUrl	If you are using Oracle Configurator Developer to test user interfaces launching a DHTML window, you must specify thin client usage by adding this parameter using the format: JDBC:ORACLE:Thin:@*host*:*port*:*sid*, where *host* is the name of the local machine, *port* is the port where your service is running, and *sid* is your server name.
		
		You must specify the JDBC connection URL in the section corresponding to the current data source. Example:
		
		`[Test11]`
		
		`JdbcUrl=jdbc:oracle:thin:@host:1521 :Test11`

Table 4–1 Spx.ini Parameters for Development and Testing

Section	Parameter	Description
Test	Launch	Sets the type of environment to launch when using the Test/Debug button in Oracle Configurator Developer.Launch=2 specifies the Dynamic HTML in a browser. When Launch=2 is specified, the parameter InitServletURL must also be set to specify the URL of the servlet generating the Dynamic HTML in a browser. Launch=3 specifies the Configurator Java Applet. When Launch=3 is specified, the parameter InitServletURL must also be set to specify the URL of the servlet generating the Configurator Java Applet. See the *Oracle Configurator and SellingPoint Administration Guide* for additional parameters that must be set for the Oracle Configurator Developer test environment.

4.4 Test Your Oracle Configurator Developer Installation

Run Oracle Configurator Developer to test the installation and connectivity. Before you run Oracle Configurator Developer, read the *Oracle Configurator Release Notes*, especially the section "Before You Start".

> **Note:** For **optimum performance** while running Oracle Configurator Developer, close all other applications and turn off scheduled applications such as virus scans, disk optimizations, and background processes. For example, be sure MS Find Fast's automatic index updating is turned off.

Using the base `spx.ini` file for development, start Oracle Configurator Developer. Log in as the DBOwner or an imported/added user listed in the CZ_END_USERS table.

To run the test configurator from Oracle Configurator Developer, execute the Generate Active Model and Generate Active UI commands, then click Test.

Using the edited `spx.ini` file, start the Oracle Configurator Developer. Oracle Configurator Developer looks for the `spx.ini` file in the `/Windows/ or /Winnt` directory on the local hard drive.

To run Oracle Configurator Developer, click the Windows **Start** button. Select **Programs** > **Oracle Configurator**. Depending on the installation option you selected, you have one or more of the following program options available:

- Help
- Manuals
- Oracle Configurator Developer
- Uninstall Oracle Configurator Developer

Warning: Before beginning to implement your configurator project, attend training in Oracle Configurator Developer, read the *Oracle Configurator Release Notes*, and read the *Oracle Configurator Developer User's Guide* or Help, especially the Introduction.

---

1. In the Oracle Configurator Developer login screen, enter your **username** and **password**.
2. Click the **Datasource** button. From the list, select the DSN for the database you would like to use. The DSNs in the list are specified in the `spx.ini` file (see *Oracle SellingPoint Configurator Administration Guide* for details).
3. Click **OK**.
4. Enter a new project name and description in the New Project dialog. If the database you chose contains existing Projects, go to the Existing tab and choose from the list.
5. Click **OK**.

Glossary of Terms

This glossary for Oracle Configurator is followed by a Glossary of Acronyms

Active Model

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

Active User Interface

The part of Oracle Configurator runtime architecture that provides the graphical views necessary to create configurations interactively. Interfaces with the Active Model and data to give users access to customer requirements gathering, product selection, and customer-centric extensions.

Application Architecture

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

Architecture

The software structure of a system. Architecture affects how a system is used, maintained, extended, and changed. See also Application Architecture.

Beta

An external release, delivered as an installable application, and subject to system, validation, and acceptance testing. Specially selected and prepared end users may participate in beta testing.

Bill of Material

A list of component items associated with a parent item (assembly) and information about how each item relates to the parent item.

BOM

See Bill of Material.

BOM Item

The nodes imported into the Oracle Configurator Developer Model that correspond to an Oracle BOM.

BOM Model

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

BOM OptionClass

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

BOM StandardItem

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

Boolean Expression

An element of a component in the Model that has two options: true or false.

Bug

See Defect.

Build

A specific instance of an application during its construction. A build must have an install early in the project so that application implementers can unit test their latest work in the context of the entire available application.

CIO

See Oracle Configuration Interface Object.

Client

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

Comparison Rule

An Oracle Configurator Developer rule type to establish a relationship that determines the selection state of a logical item (option, boolean feature, or list-of-options feature) based on a comparison of two numeric values (numeric features, totals, resources, option counts, or numeric constants). The numeric values being compared can be computed or they can be discrete intervals in a continuous numeric input.

Compatibility Rule

An Oracle Configurator Developer rule type to establish a relationship among features in the Model that specifies the allowable combinations of options. See also, Property-based Compatibility Rule.

Compatibility Table

A type of compatibility relationship where the allowable combination of options are explicitly enumerated.

Component

Represents a configurable element in a product. An element of the Model structure, typically containing features. May correspond to one screen of selections in an Oracle runtime configurator.

Component Set

An element of the Model that contains a number of components of the same type, where each component of the set is independently configured.

Configuration

A specific set of specifications for a product, resulting from selections made in an Oracle runtime configurator.

Configuration Model

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

Configuration Rules

The Oracle Configurator Developer logic rules and numeric rules available for defining configurations.

Configurator

The part of an application that provides custom configuration capabilities.

Constraint Rule

An Oracle Configurator Developer rule type to create a logical relationship among features and options. See also Rules.

Contributes to

An Oracle Configurator Developer numeric rule type for accumulating a total value.

Consumes from

An Oracle Configurator Developer numeric rule type for specifying the quantity of a resource used.

CRM

Customer Relationship Management. The aspect of the enterprise that involves contact with customers, from lead generation to support services.

Customer

The person or persons for whom products are configured by end users of the Oracle Configurator or other ERP and CRM applications.

Customer-centric Extensions

See Customer-centric Views.

Customer-centric Views

Optional extensions to core functionality that supplement product selection with rules for pre-selection, validation, and intelligent views. View capabilities include generative geometry, drawings, sketches and schematics, charts, performance analyses, and ROI calculations.

Customer Requirements

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

Data Import

Populating the Oracle Configurator schema with enterprise data from ERP or legacy systems via import tables.

Data Integration Object

Data Integration Object. A server in the runtime application that creates and manages the interface between the client (usually a user interface like the Active User Interface) and the Oracle Configurator schema.

Data Maintenance Environment

The environment in which the Oracle runtime configurator data is maintained.

Data Replication

The activity of downloading and uploading configuration, quote, and order data between the Oracle Configurator schema on the enterprise server and Oracle Configurator Mobile Database on end-user mobile laptop PCs. See also Data Synchronization.

Datasource

A programmatic reference to a database. Referred to by a datasource name, or DSN.

Data Synchronization

A process for matching the data in the Oracle Configurator schema and the data available to client processes such as the Oracle SellingPoint application. See also Data Replication.

Default

The automatic selection of an option based on the pre-selection rules or the selection of another option.

Defaults

An Oracle Configurator Developer logic rule to determine the logic state of features or options in a default relation to other features and options. For instance, if you set A to True by selecting it, B becomes true (selected) if it is available (not false) and can be set to True without contradicting a non-default rule or a previous default setting for B.

Defect

A failure in a product to satisfy the users' requirements. Defects are prioritized as critical, major, or minor, and fixes range from corrections or workarounds to enhancements. Also known as a “bug”.

Defect Tracking

A system of identifying defects for managing additional tests, testing, and approval for release to users.

Deliverable

A work product that is specified for review and delivery.

Demonstration

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

Design Chart

An Oracle Configurator Developer rule type for defining advanced Explicit Compatibilities interactively in a chart view.

Design Review

A technical review that focuses on application or system design.

Developer

The tool (Oracle Configurator Developer) used to create configuration models. The person who uses Oracle Configurator Developer to create a configurator. See also Implementer

DIO

See Data Integration Object.

End User

The ultimate user of the Oracle runtime configurator. The types of end users vary by project but may include salespeople or distributors, administrative office staff, marketing personnel, order entry personnel, product engineers, or customers directly accessing the application via web or kiosk.

Enterprise

The systems and resources of a business.

Environment

The arena in which software tools are used, such as operating system, applications, and server processes.

ERP

Enterprise Resource Planning. A software system and process that provides automation for the customer's back-room operations, including order processing.

Excludes

An Oracle Configurator Developer rule type for determining the logic state of features or options in an excluding relation to other features and options. For instance, if you set A to True, B becomes false, since it is not allowed when A is true. If you set A to False, there is no effect on B, meaning it could be true, false, or unknown.

Extended Functionality

A release after delivery of core functionality that extends that core functionality with customer-centric views, more complex proposal generation, discounting, quoting, and expanded integration with ERP, CRM, and third-party software.

Feature

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

Functional Companion

An object associated with a component that supplies methods that can be used to initialize, validate and generate customer-centric views and outputs for the configuration.

Functional Specification

Document describing the functionality of the application based on user requirements.

Incremental Construction

The process of organizing the construction of the application into builds, where each build is designed to meet a specified portion of the overall requirements and is unit tested.

Implementation

The stage in a project between defining the problem by selecting a configuration technology vendor, such as Oracle, and deploying the completed sales configuration application. The Implementation stage includes gathering requirements, defining test cases, designing the application, constructing and testing the application, and delivering it to users.

Implementer

The person who uses Oracle Configurator Developer to build the model structure, rules, and UI customizations that make up an Oracle runtime configurator.

Implies

An Oracle Configurator Developer logic rule type that determines the logic state of features or options in an implied relation to other features and options. For instance, if you set A to True by selecting it, B becomes true, since selecting A implies that B is also selected. If you set A to False by deselecting it, there is no effect on B, meaning it could be true false or unknown based on other relations B participates in. And if you set B to True by selecting it, there is no effect on A, meaning it could be true false or unknown based on other relations A participates in. But if you set B to False by deselecting it, the relation of A implies B is preserved only by having A be false (deselected) as well.

Import Tables

Tables mirroring the Oracle Configurator schema Item Master structure, but without integrity constraints. Import Tables allow batch population of the Oracle Configurator schema Item Master. Import Tables are used in conjunction with extractions from Oracle Applications or legacy data to create, update, or delete records in the Oracle Configurator schema Item Master.

Install

A program that sets up the local machine and installs the application for testing and use.

Integration

The process of combining multiple software components and making them work together.

Integration Testing

Testing the interaction among software programs that have been integrated into an application or system.

Intelligent Views

Configuration output, such as reports, graphs, schematics, and diagrams, that help to illustrate the value proposition of what is being sold.

Item Master

A table in the Oracle Configurator schema containing data used to structure the product. Data in the item master is either entered manually or imported from Oracle Applications or legacy data.

Item Type

A table in the Oracle Configurator schema containing data used to classify the product data in the item master table.

Log File

A file containing errors, warnings and other information output by the running application.

Logic Rules

Logic rules directly or indirectly set the logical state (true, false, or unknown) of features and options in the Model.

There are four (4) primary logic rules: Implies, Requires, Excludes, and Negates. Each of these rules takes a list of features or options as operands. See also Logic, Implies, Requires, Excludes, and Negates.

Maintainability

The characteristic of a product or process to allow straightforward maintenance, alteration, and extension. Maintainability must be built into the product or process from inception.

Maintenance

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

Maintenance Guide

A guide for maintaining a specific application or system. The maintenance guide covers all aspects of maintenance described in the generic Maintenance Plan.

Maintenance Plan

A document that outlines what is required for successful maintenance, and who is responsible for all the actions and deliverables of carrying out maintenance on a system.

MDUI

See Model-driven UI.

Mobile Database

See Oracle Configurator Mobile Database.

Model

The entire hierarchical “tree” view of all the data required for configurations, including model structure, variables such as resources and totals, and elements in support of intermediary rules. May consist of BOM Items.

Model-driven UI

The graphical views of the model structure and rules generated by the Active UI to present end users with interactive product selection based on configuration models.

Model Structure

Hierarchical, “tree” view of data in terms of product elements (Models, Products Components, Features, Options, BOM Models, BOM OptionClasses, BOM StandardItems, Resources, and Totals). May include reusable components.

MRP

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

Negates

An Oracle Configurator Developer logic rule type that determines the logic state of features or options in a negating relation to other features and options. For instance, if you set one item in the relationship to True, the other item must be false. And if you set one item to False, the other item must be true.

Node

The place in a Model occupied by a component, feature, option or variable, BOM Model, BOM OptionClass, or BOM StandardItem.

Numeric Rules

Rules that are used to set the global parameters specified in product structuring. See also, Contributes to and Consumes from.

OC

See Oracle Configurator.

Opportunity

The workspace in the Oracle SellingPoint application and Oracle Sales Online in which products, systems, and/or services are configured, quotes and proposals are generated, and orders are submitted.

Option

An element of the Model. A choice for the value of an enumerated feature.

A logical selection made by the end user when configuring a component.

Oracle Configurator

The product family consisting of development tools and runtime applications such as Oracle Configurator schema, Oracle Configurator Developer, Oracle Configurator window, and Oracle SellingPoint application. Also the Oracle runtime configurator variously packaged for use in networked, mobile, or web deployments.

Oracle Configurator Architecture

The application runtime architecture consists of the Active User Interface, the Active Model, and the Oracle Configurator schema or Oracle Configurator Mobile Database. The application development architecture consists of Oracle Configurator Developer and the Oracle Configurator schema, with test instances of an Oracle runtime configurator.

Oracle Configurator Developer

The suite of tools in the Oracle Configurator product family for constructing and maintaining configurators.

Oracle Configuration Interface Object (CIO)

A server in the runtime application that creates and manages the interface between the client (usually a user interface like the Active User Interface) and the underlying representation of model structure and rules in the Active Model.

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

Oracle Configurator Mobile Database

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

Oracle Configurator Schema

The implementation version of the standard Oracle runtime configurator data-warehousing schema that manages data for the configuration model. The implementation schema includes all the data required for the runtime system as well as specific tables used during the construction of the configurator.

Oracle SellingPoint Application

The test application generated by Oracle Configurator Developer. Also a full configuration environment with opportunity management, quotes, and proposals for networked or mobile deployments.

Output

The output generated by a configurator, such as quotes, proposals, bills of material (BOM), and customer-centric views.

PDM

Product Data Management. A software system that manages the version control of product data.

Populator

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

Pre-selection

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

A process that is implemented to select the initial element(s) of the configuration.

Principal Design Consultant

Member of the project team responsible for architecting the design of the application.

Product

Whatever is subjected to configuration and is the output of the application.

The root element of the Model.

Product Family

A collection of products or product lines, which are organized as a group by a provider or manufacturer.

Project

The workspace in Oracle Configurator Developer in which configurators are constructed

Project Manager

A member of the project team who is responsible for directing the project during implementation.

Project Plan

A document that outlines the logistics of successfully implementing the project, including the schedule.

Property

A named value associated with an object in the Model or the item master. A set of properties may be associated with an item type.

Property-based Compatibility Rule

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

Prototype

A construction technique in which a preliminary version of the application, or part of the application, is built to facilitate user feedback, to prove feasibility or examine other implementation issues.

Reference

The use of a reusable component within the Model. Not implemented in Release 4.2.6 or before.

Regression Test

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

Requires

An Oracle Configurator Developer logic rule type that determines the logic state of features or options in a requirement relation to other features and options. For instance, if you set one item in the relationship to True, the other item is required to be true as well. And if you set one item to False, the other item must be false as well.

Resource

Staff or materials available or needed within an enterprise.

A variable in the Model used to maintain the balance of features not consuming more of a specific resource than has been provided by other features.

Reusable Component

A component that is referenced from multiple locations in the Model. Not implemented in Release 4.2.6 or before.

Reusability

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

Rules

Also called business rules or configuration rules. Constraints applied among elements of the product to ensure that defined relationships are preserved during configuration. Elements of the product are components, features, and options. Rules express logic, numeric parameters, implicit compatibility, or explicit compatibility. Rules are used to provide pre-selection and validation capability in an application.

See also Logic Rules and Numeric Rules.

Runtime

The environment and context in which applications are run or used, rather than developed.

Sales Configuration

A part of the sales process to which configuration technology has been applied in order to increase sales effectiveness and decrease order errors. Commonly identifies needs assessment and product configuration.

Server

Centrally located software processes or hardware, shared by clients.

Solution

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

System

The hardware and software components and infrastructure integrated to satisfy functional and performance requirements.

Test Case

A description of inputs, execution instructions, and expected results, which are created for the purpose of determining whether a specific software feature works correctly or a specific requirement has been met.

Total

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

Undetermined

The logic state that is neither true nor false, but unknown at the time a logic rule is executed. This logic state is also referred to as Available, especially when considered from the point of view of the Oracle runtime configurator end user.

Unit Test

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

Update

Moving a production configurator to a new version of configuration model.

Upgrade

Moving the configurator to a new release of Oracle Configurator.

User

The person using the Oracle Configurator Developer tools and methods to build an Oracle runtime configurator. See also end user.

User Interface

The visible part of the application, including menus, dialog boxes, and other on-screen elements. The part of a system where the user interacts with the software.

User Requirements

A description of what the Oracle Configurator or Oracle SellingPoint application is expected to do from the end user's perspective.

User's Guide

Documentation on using the application or configurator to solve the intended problem.

Validation

Tests that ensure that the configured components will meet specific performance or acceptance criteria.

A type of functional companion that is implemented to ensure that the configured components will meet specific performance or acceptance criteria.

Variable

Parts of the Model that are represented by Totals, Resources, or numeric Features.

Verification

Tests that check whether the result agrees with the specification.

Glossary of Acronyms

API

Application Programming Interface

ATP

Available to Promise

BOM

Bill of Material

CIO

Configuration Interface Object

CM

Configuration Management

COM

Component Object Model

CRM

Customer Relationship Management

DBMS

Database Management System

DCOM

Distributed Component Object Modeling

DHTML

Dynamic Hypertext Markup Language

DIO

Data Integration Object

DLL

Dynamically Linked Library

DXF

Drawing Exchange Format (AutoCAD drawings)

ECO

Engineering Change Order

ERM

Enterprise Relationship Management

ERP

Enterprise Resource Planning

ESD

Electronic Software Distribution

ESP

External Service Provider

ESS

Enterprise Selling System

HT

High Tech

HTML

Hypertext Markup Language

IP

Industrial Products

IS
Information Services

ISS
Interactive Selling System

ISV
Independent Software Vendor

LAN
Local Area Network

MAPI
Messaging Application Programming Interface

MC/S
Mobile Client/Server System

MDUI
Model-Driven User Interface

MES
Marketing Encyclopedia System (Catalog)

MIS
Management Information Systems

MRP
Manufacturing Resource Planning

MS
Microsoft

OC
Oracle Configurator

OCX
Object Control File, OLE custom controls

ODBC
Open Database Connectivity

OLE
Object linking and embedding

OMS
Opportunity Management System

OOD
Object-Oriented Design

ORB
Object Request Broker

PDM
Product Data Management

PIA
Project Impact Assessment

POS
Point of Sale

QA
Quality Assurance

RAD
Rapid Application Development

RDBMS
Relational Database Management System

RFQ
Request for Quote

ROI
Return on Investment

SAS
Sales Analysis System

SCM
Supply Chain Management

SCS
Sales Configuration System

SE
Sales Engineer

SFA
Sales Force Automation

SI
System Integrator

SOT
Strategic Options Theory

SQA
Software Quality Assurance

SQL
Structured Query Language

TERM
Technology-Enabled Relationship Management

TES
Technology-Enabled Selling

UI
User Interface

VAR
Value-Added Reseller

VB

Microsoft Visual Basic

WAN

Wide Area Network

WIP

Work In Progress

Index

Symbols

$OA_HTML, 2–2
$OA_MEDIA, 2–3

A

aliases, 2–4
Apache, 2–1
  installation location, 2–2
apache_install
  installation location, 2–2
ApacheJServ.jar, 2–6, 2–12
apps.zip, 2–5, 2–12

C

CLASSPATH, 2–10
classpath, 2–5
  for custom web application, 2–5
client/server environment
  diagram, 4–4
  requirements for Oracle Configurator Developer, 4–6
  requirements for Oracle Configurator window, 4–6
  setup for Oracle Configurator Developer, 4–7
config.jar, 2–6, 2–12, 2–16
Configurator
  see also Oracle Configurator
  setup, 4–7
custom web application
  classpath, 2–5
cz3rdpty.jar, 2–5, 2–12
cz.activemodel
  property for UI Servlet, 2–14
cz.frameset.allocations.top
  property for UI Servlet, 2–19
cz.html.source.display
  property for UI Servlet, 2–18
cz.html.source.treeview
  property for UI Servlet, 2–18
cz.scrolling.treeview
  property for UI Servlet, 2–19
cz.uimanager.logpath
  property for UI Servlet, 2–18
cz.uiservlet.applet.tmp
  property for UI Servlet, 2–18
cz.uiservlet.jdbcdriver
  property for UI Servlet, 2–17
cz.uiservlet.logfilename
  property for UI Servlet, 2–17
cz.uiservlet.proxyscript
  property for UI Servlet, 2–16
cz.uiservlet.proxyurl
  property for UI Servlet, 2–16
cz.uiservlet.sourcefile
  property for UI Servlet, 2–16
cz.uiservlet.stylesheet.applet
  property for UI Servlet, 2–15
cz.uiservlet.stylesheet.applet.client
  property for UI Servlet, 2–15
cz.uiservlet.stylesheet.applet.server
  property for UI Servlet, 2–15
cz.uiservlet.stylesheet.dhtml
  property for UI Servlet, 2–15
cz.uiservlet.templateurl.ie
  property for UI Servlet, 2–14

cz.uiservlet.url
property for UI Servlet, 2–16

D

database
see also Oracle Configurator schema
connectivity, 4–8
DBOwner
in spx.ini, 4–12
in spx.ini DSN section, 4–13
in spx.ini Merlin section, 4–12
parameter for development and testing, 4–6
Developer
installation, 4–6
login, 4–14
running, 4–14
DSN
creation, 4–10

E

end user
responsibilities, 4–5

F

FND_TOP
property for UI Servlet, 2–19
Functional Companions, 2–6

G

Generate Active Model
before Test, 4–14
Generate Active UI
before Test, 4–14

H

heap size, 2–5, 2–9
html_vpath
internet server parameter, 2–2
HTTPD, 2–8
httpd.conf, 2–3

I

iAS
See Oracle Internet Application Server
ias_install
installation location, 2–2
initArgs file, 2–7
InitServletURL
parameter in the spx.ini file, 4–14
internet
configure server, 2–1
server parameters, 2–1

J

Java
interpreter, 2–5, 2–7
jdbc111.zip, 2–5, 2–6, 2–12
JdbcUrl
in spx.ini DSN section, 4–13
JDK
supported level, 2–9
jsdk.jar, 2–6, 2–12
JServ, 2–1, 2–8
installation location, 2–2
jserv_install
installation location, 2–2
jserv.conf, 2–4
jserv.log, 2–6
jserv.properties, 2–5

L

Launch
in spx.ini Test section, 4–14
LD_LIBRARY_PATH, 2–6
listening port
for JServ, 2–4
for web server, 2–4

M

media_vpath
internet server parameter, 2–3
memory requirements, 2–5
mount point

for servlet zone, 2–4

O

OA_HTML, 2–2
OA_MEDIA, 2–3
ODBC, see DSN
Oracle Configurator window
see also UI Servlet
running, 4–14
spx.ini file requirements, 4–12
Oracle Internet Application Server, 2–1
installation location, 2–2
Oracle Net8 Easy Config
for database connectivity, 4–9
Oracle SellingPoint application
login, 4–14
Oracle8 Client
installation, 4–7
requirement, 4–5
Oracle8i Enterprise Edition
requirement, 4–5

R

return_url servlets, 2–6

S

security risk, 2–10
server
kinds of, 4–4
servlet_vpath
internet server parameter, 2–3
servlets.default.initArgs, 2–7
spx.ini file
example, 4–11
location, 4–15
parameters for development and testing, 4–12

T

Test
button in Developer, 4–14
thin client usage
see JdbcUrl, 4–13
timeout, 2–4

U

UI Servlet
installation, 2–1
specifying URL, 2–16
URL
see also InitServletURL, 4–14
for UI Servlet, 2–16
Oracle Configurator window requirement, 4–6, 4–14
user
end user creation, 4–5

W

wrapper.bin.parameters, 2–7
wrapper.classpath, 2–5
wrapper.env, 2–6

X

xmlparserv2.zip, 2–5, 2–12

Z

zone.properties, 2–6