

# Oracle® Configurator

Installation Guide

Release 11*i*

April 2002

Part No. A85421-06

This document describes how to install and configure Oracle Configurator.

**ORACLE®**

Part No. A85421-06

Copyright © 1999, 2002 Oracle Corporation. All rights reserved.

Primary Author: Stephen R. Damiani

Contributing Author: Tina Brand

Contributor: Timothy Abbot, Manoj Jose, Adam Schwartz

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

Program Documentation is licensed for use solely to support the deployment of the Programs and not for any other purpose.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

**Restricted Rights Notice** Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and JInitiator, Oracle8, Oracle8i, Oracle9i, PL/SQL, SQL\*Net, SQL\*Plus, and SellingPoint are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

---

---

# Contents

<b>Send Us Your Comments .....</b>	<b>xi</b>
<b>Preface.....</b>	<b>xiii</b>
Intended Audience .....	xiii
Documentation Accessibility .....	xiv
Structure.....	xiv
Related Documents.....	xv
Conventions.....	xv
Product Support.....	xvi
<b>1 Installing Oracle Configurator</b>	
1.1 System Requirements .....	1-2
1.2 Supported Platforms.....	1-2
1.3 Required Patches .....	1-2
1.4 Completing the Oracle Configurator Installation.....	1-3
1.4.1 Modify Database Configuration File Parameters.....	1-4
1.4.2 Set Profile Options .....	1-5
1.4.2.1 ASO:Configurator URL.....	1-6
1.4.2.2 ASO:Enable Configure Model Item .....	1-7
1.4.2.3 BOM:Configurator URL of UI Manager .....	1-7
1.4.2.4 CZ:Automatically Validate on Exit .....	1-7
1.4.2.5 CZ: Populate Decimal Quantity Flags .....	1-8
1.4.2.6 CZ:Publication Lookup Mode .....	1-8

1.4.2.7	CZ:Publication Usage .....	1-9
1.4.2.8	CZ:Use Simple Configurator .....	1-9
1.4.2.9	GMA: Default Language .....	1-9
1.4.2.10	ICX: Language .....	1-9
1.4.2.11	OM:Use Configurator .....	1-9
1.4.3	Set up Oracle Applications Responsibilities and Users.....	1-10
1.4.4	Verify and Test Web Server Configuration .....	1-10
1.5	Upgrading to a New Release .....	1-10
1.6	Migrating From a Standalone Schema .....	1-11
1.6.1	Migration Prerequisites .....	1-11
1.6.2	Migrating to Release 11i .....	1-12
1.7	Installation and Setup Considerations for Multiple Language Support.....	1-15
1.7.1	Installation Tasks Specific to MLS .....	1-15
1.7.2	Workstation Fonts for MLS.....	1-16
1.7.3	Configuring Oracle Configurator Developer for MLS.....	1-16
1.7.4	Configuring JInitiator for MLS.....	1-16
1.7.5	Configuring Browsers for MLS .....	1-17
1.7.5.1	Internet Explorer .....	1-17
1.7.5.2	Netscape Navigator .....	1-17

## 2 Oracle Configurator Servlet Considerations

2.1	Configuring Apache and JServ.....	2-2
2.1.1	Verifying httpd.conf.....	2-4
2.1.2	Verifying jserv.conf.....	2-4
2.1.3	Verifying jserv.properties.....	2-5
2.1.4	Verifying zone.properties .....	2-8
2.1.5	Verifying OC Servlet Properties.....	2-8
2.2	Load Balancing.....	2-8
2.2.1	Load Balancing Procedure .....	2-9
2.2.2	Configuration for Multiple Properties Files .....	2-13
2.3	Implementing Oracle Configurator with Secure Sockets Layer .....	2-13
2.3.1	Editing jserv.properties for Secure Sockets Layer .....	2-14
2.3.2	Verifying AltBatchValidateURL.....	2-15
2.3.3	Enabling the Oracle Configurator Client for Secure Sockets Layer .....	2-16
2.3.3.1	DHTML Setup .....	2-16

2.3.3.2	Java Applet Setup .....	2-16
2.4	Oracle Configurator Servlet Properties.....	2-17
2.4.1	Setting Parameters for Apache.....	2-17
2.4.1.1	Syntax and Context for Setting Parameters .....	2-17
2.4.1.2	Usage for Setting Parameters .....	2-18
2.4.2	Descriptions of Oracle Configurator Servlet Properties.....	2-20

### **3 Troubleshooting Servlet Installation**

3.1	Miscellaneous.....	3-1
3.2	Checking the Response of the UI Servlet.....	3-1
3.3	Checking Your Model in the Runtime Oracle Configurator.....	3-3
3.4	Checking the Operation of the Apache Internet Server.....	3-5

### **4 Installing Oracle Configurator Developer**

4.1	The Oracle Configurator Developer Software .....	4-1
4.2	Installing Oracle Configurator Developer .....	4-2
4.3	Installing Oracle Configurator Developer Patches .....	4-3
4.4	Oracle Configurator Developer Client/Server Environments .....	4-4
4.4.1	Server Environment for Oracle Configurator Developer.....	4-4
4.4.1.1	Establish Users and Responsibilities.....	4-5
4.4.1.2	Users and Responsibilities for Accessing Runtime Oracle Configurators .....	4-6
4.4.1.3	Running Oracle Configurator Developer Using a Terminal Server .....	4-6
4.4.2	Client Environment for Oracle Configurator Developer .....	4-7
4.4.2.1	Oracle8 Client Installation .....	4-7
4.4.2.2	Runtime Oracle Configurator Client Requirements .....	4-8
4.4.2.3	Set Up Oracle Configurator Developer .....	4-8
4.4.2.4	Enable the Client for Database Connectivity .....	4-9
4.4.2.5	Create DSNs and DBOwners .....	4-10
4.4.2.6	Set Parameters in the spx.ini File for Development and Testing.....	4-12
4.5	Parameterized Startup of Oracle Configurator Developer .....	4-15
4.5.1	Syntax.....	4-15
4.5.2	Usage.....	4-16
4.5.3	Logging.....	4-17
4.6	Test Your Oracle Configurator Developer Installation.....	4-17

## **Glossary of Terms and Acronyms**

### **Index**

## List of Examples

2-1	Shell Script for Starting JServ in Manual Mode (in Oracle Applications).....	2-11
2-2	The jserv.properties File System Parameters for SSL.....	2-14
3-1	HTML Test Page for Invoking the DHTML runtime Oracle Configurator .....	3-3
3-2	Hello.java Test Class .....	3-6

## List of Tables

1-1	Configurator Developer Requirements.....	1-2
1-2	Profile Options for Oracle Configurator .....	1-5
1-3	CZ: Populate Decimal Quantity Flags Profile Option.....	1-8
2-1	Textual Placeholders for Configuration Files .....	2-2
2-2	Properties for the Oracle Configurator Servlet .....	2-20
2-3	Pricing Switch Values .....	2-22
2-4	Pricing Switch Settings .....	2-23
3-1	Test Strings and Results.....	3-2
3-2	Library Path Variable Names by Operating System .....	3-10
4-1	Oracle Configurator Developer Software .....	4-1
4-2	Spx.ini Parameters for Development and Testing.....	4-13
4-3	Startup Parameters for Oracle Configurator Developer.....	4-16
4-4	Logging Levels for Oracle Configurator Developer.....	4-17

## List of Figures

3-1	A Model, in the Runtime Oracle Configurator .....	3-5
4-1	Server Configuration for OC Server .....	4-5
4-2	Client/Server Data Communication Architecture .....	4-9



---

---

# Send Us Your Comments

## **Oracle Configurator Installation Guide, Release 11*i***

**Part No. A85421-06**

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 title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: [czdoc\\_us@oracle.com](mailto:czdoc_us@oracle.com)
- FAX: 781-238-9898 Attn: Oracle Configurator Documentation
- Postal service:  
Oracle Corporation  
Oracle Configurator Documentation  
10 Van de Graaff Drive  
Burlington MA 01803-5146  
USA

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

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 runtime Oracle Configurators.

## Intended 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 Oracle8i 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.

## Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle Corporation is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

**Accessibility of Code Examples in Documentation** JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

**Accessibility of Links to External Web Sites in Documentation** This documentation may contain links to Web sites of other companies or organizations that Oracle Corporation does not own or control. Oracle Corporation neither evaluates nor makes any representations regarding the accessibility of these Web sites.

## Structure

This manual contains the following chapters and appendixes:

- [Chapter 1, "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.
- [Chapter 2, "Oracle Configurator 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.
- [Chapter 3, "Troubleshooting Servlet Installation"](#) provides suggestions for resolving problems that may arise when installing the Oracle Configurator servlet.
- [Chapter 4, "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 runtime Oracle Configurator.
- ["Glossary of Terms and Acronyms"](#) contains definitions that you may need while working with Oracle Configurator.

## Related Documents

For more information, see the documentation for your release of Oracle Applications, Oracle8*i* 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 table below lists other conventions that 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.
.	
.	

Convention	Meaning
...	Horizontal ellipsis points in statements or commands mean that parts of the statement or command not directly related to the example have been omitted
<b>boldface text</b>	Boldface type in text indicates a new term, a term defined in the glossary, specific keys, and labels of user interface objects. Boldface type also indicates a menu, command, or option, especially within procedures
<i>italics</i>	Italic type in text, tables, or code examples indicates user-supplied text. Replace these placeholders with a specific value or string.
[ ]	Brackets enclose optional clauses from which you can choose one or none.
>	The left bracket alone represents the MS DOS prompt.
\$	The dollar sign represents the DIGITAL Command Language prompt in Windows and the Bourne shell prompt in Digital UNIX.
%	The per cent sign alone represents the UNIX prompt.
name ( )	In text other than code examples, the names of programming language methods and functions are shown with trailing parentheses. The parentheses are always shown as empty. For the actual argument or parameter list, see the reference documentation. This convention is <i>not</i> used in code examples.

## Product Support

The mission of the Oracle Support Services organization is to help you resolve any issues or questions that you have regarding Oracle Configurator Developer and Oracle Configurator.

To report issues that are not mission-critical, submit a Technical Assistance Request (TAR) using Metalink, Oracle's customer support Web site, at:

<http://metalink.oracle.com/>

Log into your Metalink account and navigate to the Configurator TAR template:

1. Choose the **TARs** link in the left menu.
2. Click on **Create a TAR**.
3. Fill in or choose a profile.
4. In the same form:

- a. Choose **Product**: Oracle Configurator or Oracle Configurator Developer
  - b. Choose **Type of Problem**: Oracle Configurator Generic Issue template
5. Provide the information requested in the iTAR template.

You can also find product-specific documentation and other useful information using Metalink.

For a complete listing of available Oracle Support Services and phone numbers, see:

<http://www.oracle.com/support>



---

# 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 runtime Oracle Configurator (the end-user environment). The Oracle Configurator schema and the runtime Oracle Configurator are installed with Oracle Applications Release 11*i* using Oracle Rapid Install. You install Oracle Configurator Developer 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 the same time. You select the product you want to install, and Oracle Rapid Install automatically selects and installs any dependent products. 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 runtime Oracle Configurator add-on in Oracle Order Management, Telesales, iStore, Sales Online, or Order Capture. It also describes the installation tasks for installing and running Oracle Configurator Developer to create a runtime Oracle Configurator 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*.

## 1.1 System Requirements

### Oracle Configurator Developer

[Table 1-1](#) lists the requirements for Oracle Configurator Developer.

**Table 1-1 Configurator Developer Requirements**

---

**Development Workstation**

---

Processor:  
Pentium II 300Mhz or equivalent

Memory:  
256Mb minimum  
512Mb or higher recommended

Disk Space: 40Mb Free

OS: Windows 95/98, Windows NT 4.0, and  
Windows 2000

Oracle8 Client

MS Word 97 (for generating project reports)

**Data Server**

Oracle8i Enterprise Edition

---

## 1.2 Supported Platforms

The Oracle Configurator Developer is supported on Windows 95/98, Windows NT (Intel), and Windows 2000. The Oracle Configurator schema and the runtime Oracle Configurator are supported on all of the same platforms as Oracle Applications.

## 1.3 Required Patches

After installing Oracle Applications, you must run `adpatch` to apply the latest patches to your Oracle Applications Release 11i environment. For the latest required patches, contact Oracle Support or go to Metalink at <http://metalink.oracle.com> and click on Patches. See *Maintaining Oracle Applications* for information about applying patches.

If you have applied an Oracle Applications patch, you may also be required to apply an Oracle Configurator Developer patch on your client machine. For more

information about Oracle Configurator Developer patches, see [Section 4.3, "Installing Oracle Configurator Developer Patches"](#) on page 4-3.

## 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 invoke a runtime Oracle Configurator from 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 1-4.
3. Set profile options that enable you to run the runtime Oracle Configurator within Oracle Applications. See [Section 1.4.2, "Set Profile Options"](#) on page 1-5.
4. Configure the servlet on your Internet server. See [Section 2.1, "Configuring Apache and JServ"](#) on page 2-2.
5. (Optional, but recommended) Load balance the servlet to support multiple simultaneous users of the runtime Oracle Configurator. 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 1-10.
7. If you are not installing Oracle Configurator Developer to develop, maintain, or customize the runtime Oracle Configurator, complete your installation by verifying and testing your Web server configuration. See [Section 1.4.4, "Verify and Test Web Server Configuration"](#) on page 1-10.

If you are installing Oracle Configurator Developer to develop, maintain, or customize the runtime Oracle Configurator, complete the following additional steps.

8. Install Oracle Configurator Developer to develop, maintain, or customize the runtime Oracle Configurator. See [Section 4.2, "Installing Oracle Configurator Developer"](#) on page 4-2.

9. Set up your server machine and establish Oracle Configurator Developer users and responsibilities. See [Section 4.4.1, "Server Environment for Oracle Configurator Developer"](#) on page 4-4.
10. Set up your client machine. See [Section 4.4.2, "Client Environment for Oracle Configurator Developer"](#) on page 4-7.
11. Establish data connectivity between the client machine and the Oracle Configurator schema on the server machine. See [Section 4.4.2.4, "Enable the Client for Database Connectivity"](#) on page 4-9.
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 4.4.2.6, "Set Parameters in the `spx.ini` File for Development and Testing"](#) on page 4-12 and the *Oracle Configurator Implementation Guide* for more information about these parameters.

### 1.4.1 Modify Database Configuration File Parameters

To invoke an Oracle Configurator from any Oracle Applications product, the database configuration file must be modified for your installation. This file is called `DatabaseHostname_DatabaseSID.dbc` and is located in `$FND_TOP/secure`.

1. Update the database configuration file for the database instance to include batch validate username and password parameters.

When you install or upgrade Oracle Applications, the database configuration file must include batch validate username and password parameters to use the runtime Oracle Configurator. If you do not add this information, end users may see a blank (empty) Configurator window at runtime.

For example:

```
BATCH_VALIDATE_USER=valid Oracle Applications username (for example, jsmith)
BATCH_VALIDATE_PWD=password for the above username
```

2. Oracle Configurator Developer uses thin drivers. To use Oracle Configurator Developer you must ensure that the thin driver entries in the database configuration file are not commented out.

For example:

```
APPS_JDBC_DRIVER_TYPE=THIN
```

3. Uncomment the following line and replace `host_name` with the appropriate value:

DB\_HOST=host\_name

4. Uncomment the following line and replace `port_number` with the appropriate value:

DB\_PORT=port\_number

5. Uncomment the following line and replace `db_name` with the appropriate value:

DB\_NAME=db\_name

## 1.4.2 Set Profile Options

To utilize some Oracle Configurator Developer functionality or run the runtime Oracle Configurator within other Oracle Applications such as Order Management, you must set some profile options. The profile options that affect Oracle Configurator are listed in [Table 1-2](#) on page 1-5 and are described in the sections that follow.

For more information about setting profile options, see the *Oracle Applications User's Guide*.

**Table 1-2 Profile Options for Oracle Configurator**

Profile Option	System Administrator					Requirements Default Value
	User	User	Resp	App	Site	
ASO:Configurator URL	X	X	X	X	X	Required with iStore, Telesales, SalesOnLine
ASO:Enable Configure Model Item	X	X	X	X	X	Required with iStore, Telesales, SalesOnLine
BOM:Configurator URL of UI Manager	X	X	X	X	X	Required with Order Management, iStore, Telesales, and SalesOnLine

**Table 1–2 (Cont.) Profile Options for Oracle Configurator**

Profile Option	User	System Administrator					Requirements	Default Value
CZ:Automatically Validate on Exit						X	Optional	Always Validate on Exit
CZ: Populate Decimal Quantity Flag	0	0	0	0	X		Optional	No
CZ:Publication Lookup Mode	X	X	X	X	X		Optional	Production
CZ:Publication Usage	X	X	X	X	X		Optional	Any Usage
CZ:Use Simple Configurator						X	Required with Order Management	
GMA: Default Language	X	X	X	X	X		Required	
ICX: Language	X	0	0	X	X		Required	Required
OM:Use Configurator	0	0	0	0	X		Required with Order Management	No
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.						

### 1.4.2.1 ASO:Configurator URL

This profile option indicates the location where the Oracle Configurator Servlet resides and allows iStore, Telesales, or SalesOnLine to find the Oracle Configurator Servlet URL. The Oracle Configurator Servlet URL is the location where the configurator servlet resides. This URL is set up by the installer of the servlet.

The URL is the same as the Java property `cz.uiservlet.url` defined for your Oracle Configurator 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/configurator/oracle.apps.cz.servlet.UiServlet
```

See [Section 2.4.2, "Descriptions of Oracle Configurator Servlet Properties"](#) on page 2-20.

#### 1.4.2.2 ASO:Enable Configure Model Item

This profile option enables *iStore*, *Telesales*, or *SalesOnLine* to configure models created using Configurator Developer. Valid values for this profile option are *Yes* and *No*.

#### 1.4.2.3 BOM:Configurator URL of UI Manager

This profile option allows the calling application to find the Oracle Configurator Servlet URL. The Oracle Configurator Servlet URL is the location where the Oracle Configurator Servlet resides. This URL is set up by the installer of the servlet.

---

---

**Note:** Setting this profile option is not required for installations of a runtime Oracle Configurator running in a custom Web application. The developer of the hosting application for the runtime Oracle Configurator must specify the URL of the Oracle Configurator Servlet, and then post the initialization message to that URL.

---

---

The BOM:Configurator URL of UI Manager profile option indicates the location where the Oracle Configurator Servlet resides. The URL is the same as the Java property `cz.uiservlet.url` defined for your Oracle Configurator Servlet (see [Section 2.4.2, "Descriptions of Oracle Configurator Servlet Properties"](#) on page 2-20).

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 *880n*. For example:

```
http://appsmachine.appsdomain:8800/configurator/oracle.apps.cz.servlet.UiServlet
```

#### 1.4.2.4 CZ:Automatically Validate on Exit

This profile option controls validation behavior of the runtime Configurator window when an end user clicks *Done* to end the configuration session. The default value is *Always Validate on Exit*.

### 1.4.2.5 CZ: Populate Decimal Quantity Flags

Use this profile option to control whether BOM Standard Items that accept decimal quantities are imported into Oracle Configurator will allow end users to enter a decimal quantity in a runtime Oracle Configurator. This option can be set at the Site level only. [Table 1–3](#) describes the effect of setting this profile option.

**Table 1–3 CZ: Populate Decimal Quantity Flags Profile Option**

Value	Effect
No	This is the default value. All items are imported as allowing only integer input, regardless of how they were defined in Oracle Inventory.
Yes	All items are imported as allowing either decimal or integer input, depending on how they were defined in Oracle Inventory.

After you set this profile option to Yes, and import, refresh, and republish your configured models, you must then restart the Oracle Configurator Servlet for configurations to use the new setting. See [Chapter 2, "Oracle Configurator Servlet Considerations"](#) on page 2-1

If your host application does not support input of decimal quantities, it is recommended that you set the value of this profile option to No and import all BOM Standard Items as allowing only integer values.

If your sales order system does support input of decimal quantities, set the value of this profile option to Yes and then import new BOM Models or refresh and republish existing models to use the new setting.

This profile option affects the behavior of the Import and Import Refresh concurrent programs. The internal name of this profile option is CZ\_IMP\_DECIMAL\_QTY\_FLAG. See the *Oracle Configurator Implementation Guide* for details.

### 1.4.2.6 CZ:Publication Lookup Mode

Hosting applications may include a publication mode parameter in the session initialization message to select the publication (model and UI) to display in the Configurator window. If the initialization message does not provide this parameter, the Oracle Configurator Servlet uses the value of CZ:Publication Lookup Mode. You can set this profile option to either Production or Test at the site level only. The default is Production.

### 1.4.2.7 CZ:Publication Usage

An Oracle Configurator Developer user specifies one or more Usages when publishing a configuration model to control its availability when accessed by hosting applications such as iStore or Order Capture. The hosting application optionally includes the value of this parameter in the session initialization message to select the UI and model to display in a Configurator window. This profile option provides a default Usage to select a model publication when no Usage is specified in the session initialization message. Valid values for this profile option are all Usage names defined in Oracle Configurator Developer.

The default value of this profile option is Any Usage, which does not limit the availability of publications based on Usages. For more information about Publishing, see the *Oracle Configurator Developer User's Guide*.

### 1.4.2.8 CZ:Use Simple Configurator

If you are using Oracle Order Management, this profile option must be set to No to a runtime Oracle Configurator window. If you are not using Order Management, this profile option is not used.

### 1.4.2.9 GMA: Default Language

The default value for this user profile option is set at the Site level by Oracle Rapid Install and is the base language of your Oracle Applications instance. The value of this profile option must be the same as the language in use on the workstation from which the user logs in to Oracle Applications. This profile option is relevant if you are using Multiple Language Support (MLS). See [Section 1.7.3, "Configuring Oracle Configurator Developer for MLS"](#) on page 1-16.

### 1.4.2.10 ICX: Language

This user profile option is set at the Site level by Oracle Rapid Install. Its value is the base language of your Oracle Applications instance, which is stored in FND\_LANGUAGES.LANGUAGE\_CODE.

### 1.4.2.11 OM:Use Configurator

This profile option supports usage of a runtime Oracle Configurator to configure products from Oracle Order Management. Valid values for this profile option are Yes or No. Set this profile option to Yes to use the runtime Oracle Configurator to configure items. Set this option to No to use the Order Management Options window to create sales orders by selecting options from an ATO/PTO model. See

the *Oracle Order Management User's Guide* for more information about the Options window.

### 1.4.3 Set up Oracle Applications Responsibilities and Users

Any Oracle Configurator user must also be defined in Oracle Applications and be assigned a Configurator responsibility (either the Configurator Administrator or Configurator Developer). For more information about assigning responsibilities, see *Installing Oracle Applications*.

### 1.4.4 Verify and Test Web Server Configuration

To use Oracle Configurator your Web server must be up and running. To verify that Web server is up and running properly, see the documentation for your specific Web server.

## 1.5 Upgrading to a New Release

If you are upgrading from a previous version of Oracle Configurator Release 11i, run `adpatch` to apply the latest patches to your Oracle Applications Release 11i environment. For the latest required patches, go to Metalink at <http://metalink.oracle.com> and click on **Technical Libraries > Alerts > Configurator**. See *Maintaining Oracle Applications* for information about applying these patches. Be sure to read the *Oracle Configurator Release Notes* for important information about the new version of Configurator Developer.

If you are upgrading to Patchset E and you use a custom program to import data into the CZ schema, refer to the *Oracle Configurator Implementation Guide* for information about a new field added to the CZ\_PS\_NODES table that can affect the display of items in the DHTML Summary screen.

If you are upgrading to Patchset G, see the Patchset G *Oracle Configurator Release Notes* for information regarding the how Numeric Rules that use the division (/) operator have changed.

If you are migrating from a previous version of an Oracle SellingPoint schema, see [Section 1.6, "Migrating From a Standalone Schema"](#) on page 1-11.

## 1.6 Migrating From a Standalone Schema

To migrate from a standalone Oracle SellingPoint schema to the Oracle Configurator schema of the Oracle Applications Release 11*i* database, the schema versions in your source and target databases must be the same. The scripts described in this section enable you to migrate data only from schema 14e of Oracle SellingPoint to schema 14e of Oracle Applications Release 11*i*. Only Release 11*i* versions 11.5.3 and 11.5.4 use schema 14e; if these versions are not available, contact Oracle Support.

Note that Oracle Applications Release 11*i*, version 11.5.5 and above use schema 15*i*. If your Oracle Applications installation uses schema 15*i* and you need to migrate from Oracle SellingPoint, contact Oracle Support for more information.

See [Section 1.6.1, "Migration Prerequisites"](#) on page 1-11 for more information.

---



---

**Warning:** Data migration is a "one-time" process. Once migration is complete, do not repeat the process or use the migration scripts to refresh data in the Oracle Applications database.

---



---

### 1.6.1 Migration Prerequisites

Before migrating data from your current Oracle SellingPoint standalone schema to Oracle Applications Release 11.5.3 or 11.5.4, you must:

- Ensure that both your source (Oracle SellingPoint) and target (Oracle Applications 11*i*) schemas are at version 14e. To do this, use SQL\*Plus to connect to your Oracle SellingPoint schema select the version settings from the CZ\_DB\_SETTINGS table.

For example:

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

The result should be MAJOR\_VERSION = 14, MINOR\_VERSION = e.

- Because Oracle SellingPoint release 4.2.2 and later use schema version 14c, you must upgrade your Oracle SellingPoint schema to 14e before migrating. To do this, perform the following:

- a. Using SQL\*Plus, navigate to the `/DBAdmin/Server` directory on either the Oracle Configurator Developer CD or the location where Oracle Configurator Developer is installed.
- b. Run the script `14c_to_14e.sql`. For example:

```
SQL> @14c_to_14e
```

- (optional) Run the PURGE utility on your standalone schema instance if you do not want to migrate logically deleted data. See the documentation for your current standalone version of Oracle SellingPoint for information about the PURGE utility.
- Perform a clean Oracle Rapid Install of the base Oracle Applications 11i database (schema 14e, Release 11.5.3 or 11.5.4) and ensure that the Oracle Configurator schema is *not* populated. (All tables in the Oracle Configurator schema are prefixed with `CZ_`.)
- Determine the location of your source (Oracle SellingPoint) and target (Oracle Applications 11i) databases.
- Determine the SID, hostname, and listener port number of the source database.

## 1.6.2 Migrating to Release 11i

Use the following procedure to migrate data from Oracle SellingPoint schema 14e to Oracle Configurator schema 14e in the base Release 11i Oracle Applications database (11.5.3 or 11.5.4). The migration process will fail if the Oracle SellingPoint and Oracle Applications schemas are not the same.

Before migrating, be sure you have performed all of the steps listed in [Section 1.6.1, "Migration Prerequisites"](#) on page 1-11.

---

---

**Warning:** There is no data processing allowed during data migration. Therefore, before migrating to Release 11i be sure you have met all of the prerequisites and that no developers or end users are connected to: your standalone schema; the Oracle Applications 11i database; Oracle Configurator Developer; a test deployment of the Oracle SellingPoint application or a runtime Oracle Configurator.

---

---

1. Using SQL\*Plus, connect to your Oracle Applications 11i schema as the Oracle Configurator user `CZ`. This user has `RESOURCE` privileges.

For example:

```
SQL> conn cz/cz@appssid
```

In this example, *appssid* is the name of the Oracle8i Enterprise Edition instance on which the Oracle Applications 11i database is installed.

2. Navigate to the `/DBAdmin/Migration` directory on either the Oracle Configurator Developer CD or the location where Oracle Configurator Developer is installed.

For example:

- a. Start SQL\*Plus
  - b. Choose **File > Open**
  - c. Navigate to the `/DBAdmin/Migration` directory
  - d. Click **Cancel** (although the SQL prompt does not change, you are now in the `/DBAdmin/Migration` directory)
3. Set up your migration environment by running the script `CZ_MIGRATE_SETUP.sql`. (Do not run any Oracle Configurator SQL\*Plus scripts from SQL Worksheet.)

For example:

```
SQL> @cz_migrate_setup
```

4. Enter the username for the source (standalone) database schema.
5. Enter the password for the source (standalone) database schema.
6. Enter the name of the database link to be created. The migration script uses the database link to move data between the source and target schemas. If `GLOBAL_NAMES` is set to true, the link name must match the global name of the database to which you are creating the link. Otherwise, you can enter any name you want. See your database administrator for more information.
7. 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 schema, then press Enter for the two prompts that follow.

If the source schema is *not* located on the same database instance as the target schema, enter the SID, hostname, and the listener port number for the remote database where the source schema is located.

8. The script creates the migration packages, a database link, and a log file of the process called `cz_migrate_setup.log`. It also compares the tables in the source and target schemas. The table comparison verifies whether:

- like columns exist
- data types match
- nullable fields match

Note that mismatch messages appear during the table comparison due to the following known schema differences:

- The `CZ_DB_SETTINGS` table contains two additional records, `DBLinkName` and `MigrationStatus`, in the `11i` schema. These fields are used for migration purposes.
- `CZ_XFR_TABLES` contains many additional records in the `11i` schema which are used for migration purposes.
- `CZ_DB_LOGS` is not migrated.
- The `CHAR` data type used in the standalone schema is replaced with the `VARCHAR2` data type in the `11i` schema.
- The `CZ_EXP_TMP_LINES` table exists in the `11i` schema only.
- All `CZ_IMP_xx` (import) tables may exist only in the `11i` schema.
- The `PROPERTY_VALUE_NUM` column of the `CZ_PSNODE_PROPCOMPAT_GENS` table exists in the `4.2.2` schema only.
- All nullable fields in the standalone tables `CZ_PRICING_STRUCTURE`, `CZ_PSNODE_PROPCOMPAT_GENS`, and `CZ_RULE_FOLDERS` are *not* nullable in the `11i` schema.
- The `CZ_END_USERS` table is migrated, but the users are not defined as Oracle Applications users. To define end users as Oracle Applications users, see *Installing Oracle Applications*.

---

---

**Note:** A truncation error may occur during migration if a column data type is the same in both schemas but the length is different.

---

---

9. Run `CZ_RUN_MIGRATE.sql` to migrate your Oracle SellingPoint data to the Oracle Configurator schema in the Oracle Applications Release `11i` database.

For example:

```
SQL> @cz_run_migrate
```

The script copies all data from the source schema to the target schema and displays a status message when the migration is complete.

---

---

**Note:** The migration script uses queries such as "Insert into xx as Select \* from yy" to copy data for most tables. Therefore, a rollback segment error can occur for large volume tables if the rollback segment for your database is set too low. (The appropriate setting for the rollback segment depends on the size of your database and the volume of data it contains.) If this occurs, increase the size of your rollback segment and then re-run `CZ_RUN_MIGRATE.sql`.

---

---

10. Apply the latest Oracle Configurator patch from Metalink. See [Section 1.3, "Required Patches"](#) on page 1-2 for more information about obtaining and applying patches.

## 1.7 Installation and Setup Considerations for Multiple Language Support

Multiple Language Support (MLS) enables you to create a model and one or more user interfaces in your base language and then display the runtime UI in any language in which you do business. For information about using this feature in Configurator Developer, see the *Oracle Configurator Developer User's Guide*.

### 1.7.1 Installation Tasks Specific to MLS

Consult *Oracle Applications Concepts* for background information on selecting languages, character sets, and territory values. Then consult *Installing Oracle Applications* (11.5.3) and follow the step on "Select NLS settings" in either Chapter 2, "Single-node Installations" or Chapter 3, "Multi-node Installations".

Finally, follow the steps in the section "Set up National Language Support (NLS)" in Chapter 5, "Finishing Your Installation". This task involves consulting the Oracle Applications NLS Release Notes for Release 11*i*, which can be obtained through Metalink at <http://metalink.oracle.com> (follow the links for **Apps 11*i* information > NLS > 11*i* NLS Release Notes**).

## 1.7.2 Workstation Fonts for MLS

The following procedures configure your workstation for using fonts compatible with MLS.

On a workstation running Windows NT or Windows 2000, if you follow the normal procedure for installing Windows, then the fonts for other languages are installed by default.

If you need to change the input locale for entering text in the character set of a desired language, go to the Windows Control Panel and select **Regional Settings** (Windows NT) or **Regional Options** (Windows 2000). On the **Input Locales** tab, you can add or remove input locales as needed. For details on other setup tasks, please refer to the Windows 2000 Help, by searching on the keyword: "regional settings" (Windows NT) or "regional options" (Windows 2000).

---

---

**Warning:** It is possible to change the input locale on your local machine after logging in to Configurator Developer and then modify existing models. However, Oracle corporation strongly recommends that you do not do this as it can corrupt model data.

---

---

## 1.7.3 Configuring Oracle Configurator Developer for MLS

The language in which you log in to Oracle Configurator Developer must match the base language of the Oracle Applications database instance to which you are connecting. There is no option for running Configurator Developer in a different language than the one used by the database. See [Section 1.4.2.9, "GMA: Default Language"](#) on page 1-9.

In the Windows Control Panel, the Regional Settings (or Regional Options) must specify on the Number tab that the decimal symbol to be used when displaying numbers is set to "." (the period or dot character), and that the digit grouping symbol must be set to some character that is not the one set for the decimal symbol. You will not be able to log on to Oracle Configurator Developer if these settings are not configured correctly.

## 1.7.4 Configuring JInitiator for MLS

If you are using the Java Applet user interface, you should use version 1.1.8.7 of Oracle JInitiator. See [Table 4-2, "Spx.ini Parameters for Development and Testing"](#) on page 4-13 for more details.

## 1.7.5 Configuring Browsers for MLS

The following procedures configure your browser for using fonts compatible with MLS.

### 1.7.5.1 Internet Explorer

If your browser is Microsoft Internet Explorer 5.0 or higher, begin by visiting a Web site that uses the fonts appropriate to the language you want to use. Before you reach this site, a message box will appear, asking you whether you want to download a font driver for the language. Click **Yes**, and Internet Explorer will download and install it automatically. Then restart Internet Explorer. Then choose **View > Encoding > More**, and the specific character set you want to use for the language you specified.

### 1.7.5.2 Netscape Navigator

If your browser is Netscape Navigator 4.6 or higher, and is an English language version with foreign language fonts installed, use the following procedure. First, choose **Edit > Preferences > Navigator > Languages**, and the languages that you wish to use at runtime. Then choose **View > Character Set** and the specific character set you wish to use for the desired language. Then choose **View > Character Set > Set Default Character Set**. Then choose **Edit > Preferences > Appearance > Fonts** and select the desired code set from the For the Encoding list (such as "Japanese"). In the option group for respecting document-specified fonts, choose the option that uses your default font setting, and ignores the document-specified fonts.



---

---

# Oracle Configurator Servlet Considerations

To run a runtime Oracle Configurator (Java applet or DHTML), you must have the Oracle Configurator Servlet (or "OC Servlet") installed on your internet server. Installing the OC 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. See [Section 1, "Installing Oracle Configurator"](#) on page 1-1.
2. Configuring Apache and JServ to work with the OC Servlet, by verifying (and modifying, if necessary) the Apache configuration files. See [Section 2.1, "Configuring Apache and JServ"](#) on page 2-2.
3. Verifying or modifying Java system properties for the OC Servlet. See [Section 2.1.5, "Verifying OC Servlet Properties"](#) on page 2-8, and [Section 2.4, "Oracle Configurator Servlet Properties"](#) on page 2-17.
4. Load-balancing your Apache Web server. See [Section 2.2, "Load Balancing"](#) on page 2-8.

You may want to consult the *Apache 1.3 User's Guide* and Apache Web site (<http://java.apache.org>) when installing, configuring, or load balancing the OC Servlet.

---

---

**Note:** These instructions assume that you are installing on the Solaris™ Operating Environment platform, unless noted otherwise.

---

---

---



---

**Note:** You should run the Oracle Configurator Servlet under the production version of the latest version of JDK 1.3 for your platform. The production version runs significantly faster than the reference version. See [Section 2.1.3, "Verifying jserv.properties"](#) on page 2-5 for details on verifying this setting.

---



---

## 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 OC 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:
  - [Section 2.1.1, "Verifying httpd.conf"](#) on page 2-4
  - [Section 2.1.2, "Verifying jserv.conf"](#) on page 2-4
  - [Section 2.1.3, "Verifying jserv.properties"](#) on page 2-5
  - [Section 2.1.4, "Verifying zone.properties"](#) on page 2-8
  - [Section 2.1.5, "Verifying OC Servlet Properties"](#) on page 2-8
- For a description the OC Servlet properties that are used in these files, see [Section 2.4](#) on page 2-17.

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
<code>ias_install</code>	<code>/d01/oracle/viscomn/util/apache/1.3.9/Apache</code>	The directory in which you install iAS, using Oracle Rapid Install.

**Table 2–1 (Cont.) Textual Placeholders for Configuration Files**

Placeholder	Example Values	Comment
<code>apache_install</code>	<code>ias_install/Apache</code> (for example, <code>/d01/oracle/viscomm/util/apache/1.3.9/Apache/</code> )	The directory in which you install Apache, as part of iAS, using Oracle Rapid Install.
<code>jserv_install</code>	<code>ias_install/Jserv</code> (for example, <code>/d01/oracle/viscomm/util/apache/1.3.9/Apache/Jserv</code> )	The directory in which you install JServ, as part of iAS, using Oracle Rapid Install.
<code>hostname</code>	<code>www.mysite.com</code>	The name of the host machine.
<code>portnum</code>	<code>8802</code>	The port number used by the Apache listener, which is specified by <i>Port</i> in <code>httpd.conf</code> .
<code>html_vpath</code>	<code>OA_HTML</code>  <code>apache_install/htdocs/html</code>	When running under Oracle Applications, the location pointed to by <code>\$APPL_TOP/html</code> .  Otherwise, a directory located under <code>/htdocs</code> , which is specified by <code>DocumentRoot</code> in <code>httpd.conf</code> .
<code>media_vpath</code>	<code>OA_MEDIA</code>  <code>apache_install/htdocs/media</code>	When running under Oracle Applications, the location pointed to by <code>\$APPL_TOP/media</code> .  Otherwise, a directory located under <code>/htdocs</code> , which is specified by <code>DocumentRoot</code> in <code>httpd.conf</code> .
<code>servlet_vpath</code>	This should always be: configurator	A mounting location specified by <code>ApJServMount</code> in <code>jserv.conf</code> .

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_$APPL_TOP
```

your configuration file should contain:

```
/d01/oracle/visappl
```

rather than:

```
$APPL_TOP
```

### 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/ "$APPL_TOP/html"

Alias /OA_MEDIA/ "$APPL_TOP/html"
```

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 OC 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
Oracle Rapid Install
```

### 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 OC Servlet zone, and the virtual path for the zone:

```
ApJServMount /name_of_zone /zone_property_filename
```

Where *zone\_property\_filename* is the name of the zone's properties file, without the extension `.properties`. The default name is `root`, which would point to the file `zone.properties`.

This setting is required, in order to determine the *servlet\_vpath*:

```
ApJServMount /configurator /root
```

This causes the OC Servlet to use the file `zone.properties`.

### 2.1.3 Verifying `jserv.properties`

The configuration file `jserv.properties` sets Java system properties on the Java Virtual Machine (JVM) that runs the JServ servlet engine. These system property values will be available to the OC Servlet when it starts up.

By default, Oracle Rapid Install places `jserv.properties` in *jserv\_install/etc*.

Verify that the JServ engine is using the production version (as opposed to the reference version) of the JDK. For example:

```
wrapper.bin=/local/java/jdk1.3p/bin/java
```

Oracle strongly recommends that you run Java 1.3 or higher. However, if you are using an earlier version (such as 1.2.2), ensure that the JServ engine is using native (rather than "green") threads. (Version 1.3 uses native threads by default.) You can verify this by checking that the following parameter is the first parameter specified in `jserv.properties`:

```
wrapper.bin.parameters=-native
```

On platforms that are running Java with green threads and have no native thread implementation, specify the `Xss8m` option so that you use an 8 MB native stack. To specify this option, verify that the following parameter is the first parameter specified in `jserv.properties`:

```
wrapper.bin.parameters=-Xss8m
```

---

---

**Note:** If your operating system is HP-UX, you must use JDK version 1.2.2.08 or higher.

---

---

Verify that a single entry like one of the following examples is added to set the maximum heap size. (Oracle recommends at least 600MB, but the required heap size may vary depending on your configuration.)

```
wrapper.bin.parameters=-mx600m
```

```
wrapper.bin.parameters=-mx1000m
```

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

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

```
wrapper.classpath=local_value_of_${SOA_JAVA}/apps.zip
```

```
wrapper.classpath=local_value_of_${SOA_JAVA}/jdbc12.zip
```

```
wrapper.classpath=local_value_of_${SOA_JAVA}/xmlparserv2.zip
```

---

---

**Note:** The second entry above assumes you are using JDK version 1.2 or above, which requires `jdbc12.zip` in the classpath. If you are using JDK version 1.1.8 or before, enter `jdbc11.zip` instead.

---

---

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 they must be in the OC Servlet's classpath. For example, if you install your Functional Companion classes in `jserv_install/servlets`, then add the following parameter:

```
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=library_path=local_value_of_${CZ_TOP}/bin
```

The name of `library_path` varies by platform. For Solaris™ Operating Environment, it is `LD_LIBRARY_PATH`. For HP, it is `SHLIBPATH`. For Windows NT, it is `%PATH%`. The `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

The configuration file `zone.properties` sets Java servlet properties on the OC Servlet that are not available to the JVM running the JServ engine. This file is read after `jserv.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 that are often modified and recompiled, so that they can be reloaded.

```
repositories=where_modified_classes_are_reloaded_from
```

**Example:**

```
repositories=jserv_install/servlets
```

Verify that certain Java property values are being set, by being passed as initial arguments to the Java interpreter. See [Section 2.1.5, "Verifying OC Servlet Properties"](#) on page 2-8 for details.

## 2.1.5 Verifying OC Servlet Properties

Verify that certain Java property values are being set for the OC Servlet, by being passed as arguments to the Java interpreter. By default, Oracle Rapid Install creates servlet property settings in `zone.properties`.

You can also set other Java servlet properties to control the behavior of the OC Servlet, such as:

```
servlets.default.initArgs=cz.scrolling.treeview=auto
```

See [Section 2.4, "Oracle Configurator Servlet Properties"](#) on page 2-17 for details on setting these properties and a description of each.

## 2.2 Load Balancing

Load balancing means distributing processing and communications activity evenly across a computer network so that no single device is overwhelmed. Load balancing is especially important for networks where it is difficult to predict the number of requests that will be issued to a server.

For more information about load balancing, visit the Apache Web site 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 OC Servlet by your end users.

---

---

**Note:** Internet Application Server (iAS) version 1.0.2.2 or later can be set up to automatically load balance server processes using a process manager. If you are using version 1.0.2.2 or later, refer to the corresponding release of the Oracle 9i Application Server documentation for information about setting up the `mod_oprocmgr` module. Otherwise, refer to the information provided in this section.

---

---

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. Only the second approach is outlined here, and assumes a single Web listener.

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 (using a script such as the one in [Example 2-1](#) on page 2-11).

The Solaris™ Operating Environment shell in which you run JServ should set the maximum number of file descriptors to 1024. For example:

```
ulimit -n 1024
```

You should set the **maximum heap size** for the Java Virtual Machine to an optimal value, such as 1000MB. Do this by passing a runtime parameter, like this:

```
-mx2097151000
```

```
or
```

```
-mx1000m
```

---

---

**Note:** Oracle corporation recommends a minimum of 600MB for the JVM and a maximum of 1500 MB. (The maximum allowable heap size varies by operating system. For example, the maximum allowed for Windows NT is 1000 MB.)

---

---

## 2.2.1 Load Balancing Procedure

Here is one possible procedure for setting up load balancing:

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

```
ApJServManual on
```

For more information about ApJServManual, see [Section 3.4, "Checking the Operation of the Apache Internet Server"](#) on page 3-5.

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

```
ApJServMount /configurator balance://set1/zone_name
```

In this example, `set1` is your arbitrary name for the set of JServ instances that you are spawning, and `zone_name` is the name that you specified for your zone's properties file in `jserv.properties`.

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` 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 start up in manual mode. To do this you should use a shell script. An example script is shown in [Example 2-1](#) on page 2-11.

Be aware that none of the system properties set in `jserv.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`).

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

```
bindaddress=*
```

---



---

**Warning: 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.

- In `jserv.properties`, change `port` to

```
port=portnum_jserv
```

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

- In `jserv.properties`, you may optionally want to change the `root.properties` setting to point to a different zone properties file for each servlet, in order to read a different set of runtime parameters (for example, to send your logs to a different directory for each servlet). See [Section 2.2.2, "Configuration for Multiple Properties Files"](#) on page 2-13 for details.
- Start up your JServ instances by running the script in [Example 2-1](#).
- Start up Apache:

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

### **Example 2-1 Shell Script for Starting JServ in Manual Mode (in Oracle Applications)**

```
#!/bin/sh

ulimit -n 1024

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

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

cmdlineargs="-Dcz.uimanager.logpath=/d01/oracle/viscomm/util/apache/1.3.9/Apache/Jserv/logs/ \
```

```
-native \  
-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
```

If you are setting OC Servlet parameters for each JServ instance (similar to how parameters are passed in `jserv.properties`), rather than for the entire servlet zone (by setting the parameters in `zone.properties`), then you need to pass the same parameters to each JServ instance. To do this, modify the `cmdlineargs` in the script in [Example 2-1](#) to include the parameters as Java system property settings. The lines would look like this example:

```
...  
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"  
...  
...
```

Notice that the settings for `cz.uimanager.logpath` and `mx` are the same as in [Example 2-1](#), since you would not be setting those in `zone.properties`.

---

---

**Note:** The script in [Example 2-1](#) is for the OC Servlet running inside Oracle Applications Release 11*i*. If you are running the OC 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 [Section 2.1.3, "Verifying jserv.properties"](#) on page 2-5 for details.

---

---

## 2.2.2 Configuration for Multiple Properties Files

If you are following step 7 on page 2-11, edit the configuration files as follows, for each JServ instance that you are creating.

1. Create another version of `jserv.properties` (for example, `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 in step 2 below:

```
filename.properties=file_location/filename.properties
```

Example:

```
oc_configurator.properties=/d01/oracle/viscomn/util/apache/1.3.9/Apache/Jserv/  
etc/oc_configurator.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 on page 2-11:

```
port=portnum_jserv2
```

2. Create another version of `zone.properties` (for example, `zone2.properties`). In this alternate version, point to different log files. For example:

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

3. Create another version of the script in [Example 2-1](#). In this alternate version, change the lines indicated in the example by the comment "# for 2nd JServ instance...".

## 2.3 Implementing Oracle Configurator with Secure Sockets Layer

**Secure Sockets Layer (SSL)** is a protocol that creates a secure connection between a client and a server machine and enables you to safely transmit private documents over the Internet. SSL uses a public **key** to encrypt data that is transferred over the SSL connection. (A key is a password or table needed to decipher encoded data.)

To set up Oracle Configurator to run in SSL mode, perform the following:

1. Install Internet Application Server (iAS) version 1.0.2.2 or later and configure it to run SSL.  
For more information, consult the *Apache 1.3 User's Guide* and the Apache-SSL Web site at <http://www.apache-ssl.org>.
2. Install Oracle Configurator version 16-38 or later. (To upgrade to version 16-38, apply patch 2102442.) See [Chapter 1, "Installing Oracle Configurator"](#).
3. Edit the Oracle Configurator system parameters in `jserv.properties` file to use the SSL server and port. This step is required for SSL to support HTML hosting applications such as *iStore*. See [Section 2.3.1, "Editing jserv.properties for Secure Sockets Layer"](#) on page 2-14.
4. Verify that `AltBatchValidateURL` is present and set correctly in the `ORAAPPS_INTEGRATE` section of the `CZ_DB_SETTINGS` table. See [Section 2.3.2, "Verifying AltBatchValidateURL"](#) on page 2-15.
5. Set the profile option `BOM:Configurator URL` of UI Manager to the *secure* URL. This option should be set at the Site level, but it can also be set at the User level. For information about setting profile options, see the *Oracle Applications User's Guide*.  
You can test this URL using the same method described in [Section 2.3.2, "Verifying AltBatchValidateURL"](#) on page 2-15.
6. Set up your Web browser and Jinitiator to support SSL. See [Section 2.3.3, "Enabling the Oracle Configurator Client for Secure Sockets Layer"](#) on page 2-16.

### 2.3.1 Editing jserv.properties for Secure Sockets Layer

To support SSL, all of the Oracle Configurator system parameters in the `jserv.properties` file must use the secure "https" protocol, and must include the name and port number of your secure server. (All URLs that require an SSL connection start with "https" instead of "http".)

In [Example 2-2](#), `myservername` is the name of the *iAS* server and 443 is the secure server port number.

**Example 2-2 The jserv.properties File System Parameters for SSL**

```
wrapper.bin.parameters=-Dcz.uiservlet.templateURL=https://myse  
rvername.com:443/OA_HTML/US/czFraNS.htm
```

```

wrapper.bin.parameters=-Dcz.uiservlet.URL=https://myservername
.com:443/configurator/oracle.apps.cz.servlet.UiServlet

wrapper.bin.parameters=-Dcz.uiservlet.proxyscript=https://myse
rvername.com:443/OA_HTML/czProxy.js

wrapper.bin.parameters=-Dcz.html.source.treeview=https://myser
vername.com:443/OA_HTML/cztree.htm

wrapper.bin.parameters=-Dcz.html.source.display=https://myserv
ername.com:443/OA_HTML/czdisp.htm

wrapper.bin.parameters=-Dcz.uiservlet.blaftemplateURL=https://
myservername.com:443/OA_HTML/US/czBlafTemplate.htm

wrapper.bin.parameters=-Dcz.uiservlet.formtemplateURL=https://
myservername.com:443/OA_HTML/US/czFormTemplate.htm

wrapper.bin.parameters=-Dcz.html.source.formtreeview=https://m
yservername.com:443/OA_HTML/czFormTree.htm

```

---



---

**Note:** Each parameter should be defined on a single line in `jserv.properties`.

---



---

### 2.3.2 Verifying AltBatchValidateURL

The `AltBatchValidateURL` setting allows the batch validation process to bypass the URL that would normally be used for batch validation. This is needed if Oracle Configurator uses SSL. The value of `AltBatchValidateURL` must be your non-secure URL. Since the batch validation process communicates between the database and Web server, it is not necessary for this communication to use SSL. The batch validation process runs, for example, when booking an order in Oracle Order Management.

Enter the following command in SQL\*Plus to determine the value of `AltBatchValidateURL`:

```

SELECT * FROM cz_db_settings WHERE setting_id =
'AltBatchValidateURL';

```

The value returned is the same as the Java property `cz.uiservlet.url` for the non-secure URL. For example:

```

http://servername.com:8808/configurator/oracle.apps.cz.servlet
.UiServlet

```

You can test this URL by entering it in a Web browser followed by "?test=version". The result should be the build and schema version of Oracle Configurator running on the Apache server. For example:

```
http://servername.com:8808/configurator/oracle.apps.cz.servlet
.UiServlet?test=version
```

For more information, see "[cz.uiservlet.url](#)" on page 2-36.

## 2.3.3 Enabling the Oracle Configurator Client for Secure Sockets Layer

You can run an Oracle Configurator in SSL mode using either a DHTML or a Java applet UI, but some setup is required.

### 2.3.3.1 DHTML Setup

Because a DHTML UI runs in a Web browser, you must ensure that your browser supports SSL. To enable your browser to support SSL, perform the following:

- In Netscape Navigator, choose **Communicator > Tools > Security Info**, then click **Navigator**. Select from the available options to enable SSL.
- In Internet Explorer, choose **Tools > Internet Options**, then select the **Advanced** tab. Scroll down to view the **Security** section, and select the appropriate options to enable SSL.

### 2.3.3.2 Java Applet Setup

To use SSL with the Java applet, you must:

- Install JInitiator version 1.1.8.3 or later (prior versions do not support SSL)
- Set up the root certificate

If you know the root certificate, go the JInitiator security directory (for example, C:\Program Files\Oracle\JInitiator 1.1.8.13\lib\security), open the file certdb.txt, and add the root certificate to the end of the file.

If you do not know the root certificate, perform the following:

1. Using Internet Explorer, navigate to your secure URL.
2. Choose **File > Properties**.
3. Click **Certificates**.
4. Select the **Certification Path** tab, then select the top entry for the Certification path.

5. Click **View Certificate**.
6. Select the **Details** tab, then click Copy to File.
7. Click **Next**, then choose "Base64 encoded X.509 (.CER)".
8. Click **Next**, then enter a filename to export the certificate (for example, test.cer).
9. Click **Next**, then **Finish**. Close the Certificate and Properties windows by clicking **OK**.
10. Using a text editor, open the file you saved in step 8.
11. Copy the certificate, then paste its contents at the end of the certdb.txt file.
12. Save your work.

---

---

**Note:** Oracle corporation strongly recommends that you also have your server's SSL certificate signed by one of the following Certificate Authorities (CAs): VeriSign Inc.; RSA Data Security Inc.; GTE Corporation; GTE CyberTrust Solutions Inc. If your certificate is not signed by one of these CAs, you may receive errors when running the Java applet in SSL mode.

---

---

## 2.4 Oracle Configurator Servlet Properties

You can control certain behaviors of the runtime Oracle Configurator by passing parameter values to custom properties of the OC Servlet. These properties are passed as Java system properties to the JVM in which the process for the OC Servlet is running.

### 2.4.1 Setting Parameters for Apache

This section describes setting up parameters for your Apache Web Server.

#### 2.4.1.1 Syntax and Context for Setting Parameters

When using Apache, you can set the parameters that determine the properties for a Java servlet in either of these locations:

- In `zone.properties`, as global initialization parameters for the servlet, using this syntax:

```
servlets.default.initArgs=property_name=property_value
```

**Example:**

```
servlets.default.initArgs=cz.uimanager.logpath=/d01/oracle/viscomm/util/apache/1.3.9/Apache/Jserv/logs/
```

- In `jserv.properties`, as system execution parameters for the JServ JVM, using this syntax:

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

**Example:**

```
wrapper.bin.parameters=-Dcz.uimanager.logpath=/d01/oracle/viscomm/util/apache/1.3.9/Apache/Jserv/logs/
```

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

```
java -Dproperty_name=property_value
```

---

---

**Note:** The method recommended for use with the OC Servlet is the one using `servlets.default.initArgs`, in `zone.properties`.

---

---

### 2.4.1.2 Usage for Setting Parameters

- For most installations, it is not necessary to add properties to the `zone.properties` file. By default, the Oracle Configurator Servlet derives all required configuration information from the hosting application's session request and uses default values for other required properties. If you do add properties to `zone.properties`, these properties take precedence over information derived from the session request. Deriving configuration information from the session request makes it easier to run the OC Servlet on a different host (server machine), if required.

Although it is not necessary to add properties to the `zone.properties` file, you may want to define the following properties if, for example, you want to deploy additional servlets that display HTML pages in different languages:

[cz.uiservlet.blaftemplateurl](#) on page 2-31

[cz.uiservlet.formtemplateurl](#) on page 2-32

[cz.uiservlet.templateurl](#) on page 2-36

When adding properties to the configuration files, you can specify the property's location either by entering a fully qualified path or a relative URL. The recommended method is using a relative URL, as this method enables the OC Servlet to gather its configuration information from the session request. For an example of how each property listed above can be defined using a relative URL, see [Section 2.4.2, "Descriptions of Oracle Configurator Servlet Properties"](#) on page 2-20.

- If you set parameters in `zone.properties`, then you are setting a single set of properties for all of the servlets that are running in that servlet zone. By default, all servlets run in a single zone. By setting the properties once, in a single location, you can greatly simplify the maintenance of the multiple startup scripts required for load balancing. The properties are only available to the servlets.

If you set parameters in `jserv.properties`, then you can vary the settings for each of multiple JServ instances that you may be running. This would be the case when you are load balancing (as described in [Section 2.2, "Load Balancing"](#) on page 2-8). The settings will be used by each servlet that starts up in the JVM provided by that JServ instance. The properties are available not only to the servlets, but to any other processes in the same JVM.

- The use of `servlets.default.initArgs` parameters replaces the use of a separate `.initArgs` file.
- If a parameter for the OC Servlet is omitted from an Apache configuration file, then its default value is used by the runtime Oracle Configurator. You can omit all OC Servlet parameters, without triggering an error.
- You can leave obsolete OC Servlet parameters in your configuration files, without triggering an error. You can also set unimplemented parameters without triggering an error, if you observe the rules for syntax. You may wish to do this for testing purposes, to observe which parameters have been passed to the OC Servlet.
- You can view the parameters that have been passed, along with the other Java system properties set for the OC Servlet, in the servlet log file. See the description of [cz.uiservlet.logfilename](#) on page 2-33 for details.
- If you omit all the parameters that produce log files, then logging will be turned off. These parameters are:

[cz.uimanager.logpath](#) on page 2-25

[cz.uiservlet.logfilename](#) on page 2-33

- If you are not starting up Apache in manual mode, you can set some parameters in `zone.properties` and other parameters in `jserv.properties` without triggering an error, but this is not recommended.

---

**Note:** Oracle Configurator gives precedence to parameters that are set in `zone.properties` as `initArgs` parameters. If you set a parameter with the same name in both `zone.properties` and `jserv.properties`, then only the one in `zone.properties` is used. This behavior overrides the normal behavior of Apache, in which precedence is given to the first setting of a parameter, and `jserv.properties` is read first, effectively giving precedence to the parameters set in that file.

---

## 2.4.2 Descriptions of Oracle Configurator Servlet Properties

The properties of the OC Servlet for which you can set Apache configuration parameters are listed in [Table 2-2](#) on page 2-20. The columns “Applet” and “DHTML” indicate the type of runtime Oracle Configurator with which the property can be used. For additional details on using some of these properties with the DHTML runtime Oracle Configurator, see the *Oracle Configurator Custom Web Deployment Guide*.

**Table 2-2 Properties for the Oracle Configurator Servlet**

Property Name	Description	Applet	DHTML
Properties used by both Applet and DHTML windows:			
<a href="#">cz.activemodel</a>	on page 2-21	Y	Y
<a href="#">cz.activemodel.lazyloadlistprice</a>	on page 2-23	Y	Y
<a href="#">cz.uimanager.logpath</a>	on page 2-25	Y	Y
<a href="#">cz.uiserver.check_heartbeat_timeout</a>	on page 2-26	Y	Y
<a href="#">cz.uiserver.ciolog</a>	on page 2-27	Y	Y
<a href="#">cz.uiserver.heartbeat_interval</a>	on page 2-28	Y	Y
<a href="#">cz.uiserver.lazyload</a>	on page 2-28	Y	Y
<a href="#">cz.uiserver.poll_timeout_applet</a>	on page 2-30	Y	Y
<a href="#">cz.uiserver.poll_timeout_applet_to_dhtml</a>	on page 2-30	Y	Y
<a href="#">cz.uiservlet.dio_share</a>	on page 2-31	Y	Y

**Table 2–2 (Cont.) Properties for the Oracle Configurator Servlet**

Property Name	Description	Applet	DHTML
<a href="#">cz.uiservlet.jdbcdriver</a>	on page 2-33	Y	Y
<a href="#">cz.uiservlet.logfilename</a>	on page 2-33	Y	Y
<a href="#">cz.uiservlet.name</a>	on page 2-34	Y	Y
<a href="#">cz.uiservlet.pre_load_filename</a>	on page 2-34	Y	Y
<a href="#">cz.uiservlet.url</a>	on page 2-36	Y	Y
<a href="#">cz.uiservlet.versionfuncsavail</a>	on page 2-36	Y	Y
Properties used only by Applet window:			
<a href="#">cz.uiserver.media.folder</a>	on page 2-29	Y	N
<a href="#">cz.uiservlet.applet.summary</a>	on page 2-31	Y	N
Properties used only by DHTML window:			
<a href="#">cz.html.source.display</a>	on page 2-24	N	Y
<a href="#">cz.html.source.formtreeview</a>	on page 2-24	N	Y
<a href="#">cz.html.source.treeview</a>	on page 2-24	N	Y
<a href="#">cz.scrolling.treeview</a>	on page 2-25	N	Y
<a href="#">cz.uiserver.lazyload</a>	on page 2-28	N	Y
<a href="#">cz.uiserver.lfalse_is_not_available</a>	on page 2-29	N	Y
<a href="#">cz.uiserver.summary.itemcolumn</a>	on page 2-30	N	Y
<a href="#">cz.uiservlet.blafplateurl</a>	on page 2-31	N	Y
<a href="#">cz.uiservlet.formtemplateurl</a>	on page 2-32	N	Y
<a href="#">cz.uiservlet.proxyscript</a>	on page 2-35	N	Y
<a href="#">cz.uiservlet.templateurl</a>	on page 2-36	N	Y

**cz.activemodel**

Controls the type of prices to be displayed. See the section on pricing in the *Oracle Configurator Implementation Guide* for more details on controlling pricing.

The syntax for this setting is:

```
cz.activemodel=/switch|
```

Where *switch* is one of the values listed in [Table 2-3](#).

**Table 2-3 Pricing Switch Values**

Switch	Effect
lp	Fetch List Prices from the database.
dp	Fetch Discounted Prices from the database.
atp	Fetch ATP data from the database.
nolp	Do not fetch List Prices from the database.
nodp	Do not fetch Discounted Prices from the database.
noatp	Do not fetch ATP data from the database.

You can set more than one switch. The syntax for this setting is:

```
cz.activemodel=/switch1|/switch2|
```

Note that each switch is separated by the pipe character ( | ), and that the pipe character is required at the end of the property setting.

If you set contradictory switches, then only the first switch is respected. Example:

```
cz.activemodel=/lp|/nolp|
```

The default pricing behavior for the OC Servlet is that all price fetching is turned off. You can produce this behavior by omitting the `cz.activemodel` property from the OC Servlet's configuration files.

If one of the switches that disable fetching has been set, the end user will receive an error message when pricing is requested in the runtime Oracle Configurator.

Be aware that in Dynamic HTML in a browser, the display of columns for Discount Price and Extended Price is controlled by the initialization message. If you pass pricing parameters, these columns will be displayed.

---



---

**Caution:** Oracle also uses the `cz.activemodel` property internally to write debugging output to a file pathname. If you are using the property to write debugging output, you must ensure that the pathname is specified explicitly, to avoid conflicts with the pricing switches. For example, if your output path is `/lp`, then you must write the property value as `/lp/` to avoid conflict with the pricing switch `/lp`. If `cz.activemodel` is not specified, the debugging output is directed to [cz.uimanager.logpath](#), so be sure that you observe this caution for that property as well.

---



---

## Examples of Setting Pricing Switches

The following table lists examples of setting pricing switches, and their effects.

**Table 2–4 Pricing Switch Settings**

Setting	Effect
<code>cz.activemodel=/lp </code>	List Prices will be fetched and displayed.
<code>cz.activemodel=/nolp </code>	List Pricing is turned off. List Prices will not be fetched or displayed
<code>cz.activemodel=/lp /dp </code>	List Prices and Discounted Prices will both be fetched and displayed.
<code>cz.activemodel=/nolp /nodp </code>	No List Prices or Discounted Prices will be fetched.

### `cz.activemodel.lazyloadlistprice`

Controls the incremental fetching of list price data from the database. The fetching of list prices can impose a significant performance cost, especially if your Model is complex. See the *Oracle Configurator Implementation Guide* for more background on pricing in Oracle Configurator.

If you set this property to `true`, then list prices are only fetched for the items on the page that is currently being displayed. When the runtime Oracle Configurator is initialized, list prices are fetched only for the first screen displayed. Thereafter, list prices are fetched only when a new page is displayed.

If you set this property to `false`, then list prices are fetched for all the items in your Model, regardless of the page on which they appear.

The default value is `true`.

**Syntax:**

```
cz.activemodel.lazyloadlistprice=[true|false]
```

**Example:**

```
cz.activemodel.lazyloadlistprice=true
```

### **cz.html.source.display**

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

**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.html.source.formtreeview**

The tree template rendered when the look and feel is for the Form type. This corresponds to the "Oracle Forms Look" set in Oracle Configurator Developer.

**Syntax:**

```
cz.html.source.formtreeview=http://host:port/html_vpath/czFormTree.htm
```

**Example:**

```
cz.html.source.formtreeview=http://www.mysite.com:8802/OA_HTML/czFormTree.htm
```

### **cz.html.source.treeview**

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

**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.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
```

**cz.uimanager.logpath**

Sets the path into which the UI Server will write a log file only when it handles an exception and exits. Exception logs are named with the convention:

```
cz-exc_date_time_processid.txt
```

(Normal log file names and locations are controlled by the property [cz.uiservlet.logfilename](#) on page 2-33.)

Syntax:

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

A trailing slash character must be added to the value.

Example for the Solaris™ Operating Environment:

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

Example for Windows NT:

```
cz.uimanager.logpath=D:\orant\OSP\OSP\log\
```

---

---

**Caution:** The `cz.uimanager.logpath` property is also used internally by Oracle to write debugging output to a file pathname, in the event that the `cz.activemodel` property is not specified. See the [Caution](#) on page 2-23 for the [cz.activemodel](#) property for an explanation.

---

---

**cz.uiserver.check\_heartbeat\_timeout**

Controls the timeout for the UI Server's checking of "heartbeat" events. (See "[The Heartbeat Mechanism and Guided Selling](#)" on page 2-26 for a description of heartbeat events.) If the UI Server doesn't receive any heartbeats from the DHTML client browser after this time value, then the UI Server will end itself and the guided selling session in the DHTML user interface is terminated, with a "terminate" message being sent back to the Applet client. The default value is 30,000 milliseconds.

If loading a large configuration model on to the DHTML client, this property should be changed to a value close to the time it takes to load the model. For example, if the configuration model takes 60 seconds to load, the value of this property should be set to approximately 60000 milliseconds.

Syntax:

```
cz.uiserver.check_heartbeat_timeout=milliseconds
```

Example:

```
cz.uiserver.check_heartbeat_timeout=30000
```

**The Heartbeat Mechanism and Guided Selling**

You can enable the use of the DHTML User Interface to provide guided buying and selling in Oracle Order Management, which normally uses the Java applet User Interface. (See the *Oracle Configurator Developer User's Guide* for details on how to enable the DHTML User Interface for Order Management.)

In this situation, an Oracle Configurator Applet client runs under Order Management, but is not visible to the end user. The Applet client is not able to directly determine the status of the end user's DHTML client browser, so it is not able to know if the client browser has crashed or been closed prematurely. To handle this problem, Oracle Configurator uses a "heartbeat" mechanism, in which:

1. The DHTML client (the browser) sends heartbeat events to a DHTML session running in the UI Server. Continued heartbeats indicate that the DHTML client is still "alive". A cessation of heartbeats indicate that the DHTML client has terminated. This cessation is detected by the DHTML session.
2. The Applet client polls an Applet session running in the UI Server, to check whether the UI Server has received a termination message from the DHTML session.
3. If the heartbeats received by the DHTML session cease to satisfy a specified frequency, then the UI Server sends the termination message to the Applet

session, which is being polled by the Applet client running under Order Management.

The servlet parameters that govern the operation of the heartbeat mechanism are:

<code>cz.uiserver.check_heartbeat_timeout</code>	on page 2-26
<code>cz.uiserver.heartbeat_interval</code>	on page 2-28
<code>cz.uiserver.poll_timeout_applet</code>	on page 2-30
<code>cz.uiserver.poll_timeout_applet_to_dhtml</code>	on page 2-30

The value for all of these heartbeat parameters must be greater than zero, and must be less than the timeout value for the Web listener. (For Apache, this listener timeout value is specified by the setting for `Timeout` in `httpd.conf`.)

To use guided selling in Order Management, you must also cause the JServ engine to use native threads. (See [Section 2.1.3, "Verifying jserv.properties"](#) on page 2-5.)

### **cz.uiserver.ciolog**

Sets the absolute path and file name for the CIO activity log file.

(Normal log file names and locations are controlled by the property [cz.uiservlet.logfilename](#) on page 2-33.)

Syntax:

```
cz.uiserver.ciolog=absolute_path/filename
```

Example:

```
cz.uiserver.ciolog=jserv_install/logs/server_logs/cio_log.txt
```

### **cz.uiserver.displayconsequences**

If this parameter is set to `true`, then the OC Servlet displays the consequences of overriding a contradiction during a configuration session. If it is set to `false`, a contradiction message appears, but it does not describe the changes that will be made to the configuration if the user chooses to override the contradiction.

Syntax:

```
cz.uiserver.displayconsequences=[true|false]
```

Example:

```
cz.uiserver.displayconsequences=true
```

### **cz.uiserver.heartbeat\_interval**

Controls the frequency at which the heartbeat is sent from the client browser to the UI Server. The default value is 10,000 milliseconds.

See "[The Heartbeat Mechanism and Guided Selling](#)" on page 2-26 for background.

Syntax:

```
cz.uiserver.heartbeat_interval=milliseconds
```

Example:

```
cz.uiserver.heartbeat_interval=10000
```

### **cz.uiserver.lazyload**

Controls whether screens in the DHTML user interface are loaded in a "lazy" or "eager" manner. Lazy loading means that each screen is loaded as the end user navigates to it. Eager loading means that all of the screens are loaded when the configuration session is initialized.

Setting this parameter to `true` causes the OC Servlet to use lazy loading. Setting this parameter to `false` causes the OC Servlet to use eager loading.

The default value is `true`.

The benefit of lazy loading is that the initialization of the configuration session is faster, although the first visit to any screen is potentially slower. However, after a screen is visited once, subsequent visits to that screen are fast. This behavior may be the more effective if end users make numerous visits to a screen.

The benefit of eager loading is that the initialization of the configuration session is faster, although the first visit to any screen is potentially slower. However, after a screen is visited once, subsequent visits to that screen are fast.

Choose the value of this property depending on application screen sizes and usage. If a configuration session only visits each screen once, and a fast initial load is not critically important, it may be appropriate to set this property to `false`.

Syntax:

```
cz.uiserver.lazyload=[true|false]
```

Example:

```
cz.uiserver.lazyload=true
```

---

---

**Note:** You should consider the interaction of the behavior of this property with that provided by [cz.uiservlet.dio\\_share](#) on page 2-31 and [cz.uiservlet.pre\\_load\\_filename](#) on page 2-34.

---

---

### **cz.uiserver.lfalse\_is\_not\_available**

Controls the appearance of Options in the DHTML user interface when they have been set logically false (LFALSE) by Oracle Configurator.

Normally, logically false Options are displayed as if they had never been selected, as if their logical state were actually UNKNOWN.

Setting this parameter to `true` causes logically false (LFALSE) Options to be displayed with a special icon (such as a red "X"). This displaying of icons can impose some performance cost, especially if your Model is complex, and has many Options.

Setting this parameter to `false` causes logically false (LFALSE) options to be displayed as if their logical state were actually UNKNOWN.

In the `false` case, you can use the "Hide unselectable Options" feature of the User Interface module in Oracle Configurator Developer to completely hide these options. For details on this feature, see the section of the *Oracle Configurator Developer User's Guide* on Dynamic Visibility.

The default value is `false`.

Syntax:

```
cz.uiserver.lfalse_is_not_available={true|false}
```

Example:

```
cz.uiserver.lfalse_is_not_available=true
```

### **cz.uiserver.media.folder**

Location of the directory containing image media for the OC 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.uiserver.poll\_timeout\_applet**

Sets the time after which the UI Server's Applet session tells the Applet client to poll back, to check whether the UI Server session is terminated. The suggested range is 30,000 to 60,000 milliseconds. The default value is 20,000 milliseconds.

See "[The Heartbeat Mechanism and Guided Selling](#)" on page 2-26 for background.

Syntax:

```
cz.uiserver.poll_timeout_applet=milliseconds
```

Example:

```
cz.uiserver.poll_timeout_applet=20000
```

### **cz.uiserver.poll\_timeout\_applet\_to\_dhtml**

Controls the timeout for the DHTML session when it is being checked by the applet session. The suggested range is 30,000 to 60,000 milliseconds. The default value is 30,000 milliseconds.

See "[The Heartbeat Mechanism and Guided Selling](#)" on page 2-26 for background.

Syntax:

```
cz.uiserver.poll_timeout_applet_to_dhtml=milliseconds
```

Example:

```
cz.uiserver.poll_timeout_applet_to_dhtml=30000
```

### **cz.uiserver.summary.itemcolumn**

Controls whether the Item column is displayed or hidden on the Summary screen of the DHTML user interface.

Setting this parameter to `hidden` causes the Item column to be hidden. Omitting this property, or setting it to any other value, causes the Item column to be displayed.

This property has special relevance for MLS deployments. Because the Item column displays names that cannot be translated out of the base language, you may wish to hide this column.

Syntax:

```
cz.uiserver.summary.itemcolumn=[hidden]
```

**Example:**

```
cz.uiserver.summary.itemcolumn=hidden
```

**cz.uiservlet.applet.summary**

Sets the indentation character used for the summary grid in the Java Applet.

**Syntax:**

```
cz.uiservlet.applet.summary=" "
```

**cz.uiservlet.blaftemplateurl**

The file rendered when the look and feel of the runtime UI is the BLAF (Browser Look And Feel) type. This corresponds to the "Oracle Web Look" option set when generating a new user interface in Oracle Configurator Developer.

This property is not required in `zone.properties`. You may want to add this property if, for example, you are defining a servlet that displays HTML pages in a different language. If you define this property, you can specify its location either by entering a fully qualified path or a relative URL. The recommended method is using a relative URL.

**Relative URL:**

```
cz.uiservlet.blaftemplateurl=/OA_HTML/US/czBlafTemplate.htm
```

**Syntax of fully qualified path:**

```
cz.uiservlet.blaftemplateurl=http://host:port/html_vpath/US/czBlafTemplate.htm
```

**Example of fully qualified path:**

```
cz.uiservlet.blaftemplateurl=http://www.mysite.com:8802/OA_HTML/US/czBlafTemplate.htm
```

**cz.uiservlet.dio\_share**

Controls whether the UI Server running inside the OC Servlet shares (caches) the Model in the DIO between configuration sessions.

Setting this property to `true` enables sharing the cached Model. Sharing the cached Model improves the loading performance of sessions after the first one for a given Model, but requires that the OC Servlet be restarted in order for the runtime Oracle Configurator to reflect recent changes to the Active Model. However, configuration sessions started with the Test button in Oracle Configurator

Developer ignore the cached Model and fetch the latest Model from the database, thus reflecting changes to the Active Model. This setting provides a convenience for model developers, while providing efficiency for runtime users. As a general rule, you should keep the property set to `true`.

Setting this property to `false` disables sharing of the cached Model for *all* configuration sessions on the same OC Servlet.

The default is `true`.

Syntax:

```
cz.uiservlet.dio_share=[true|false]
```

Example:

```
cz.uiservlet.dio_share=false
```

---

---

**Note:** This property provides a development convenience, when you disable model caching. However, this convenience counteracts the performance enhancement derived by preloading and caching a Model at servlet startup, by using [cz.uiservlet.pre\\_load\\_filename](#). You must also consider the interaction of this property with [cz.uiserver.lazyload](#) on page 2-28.

---

---

### **cz.uiservlet.formtemplateurl**

The file rendered when the look and feel is for the Form type. This corresponds to the "Oracle Forms Look" set when generating a new user interface in Oracle Configurator Developer. This property is optional.

This property is not required in `zone.properties`. You may want to add this property if, for example, you are defining a servlet that displays HTML pages in a different language. If you define this property, you can specify its location either by entering a fully qualified path or a relative URL. The recommended method is using a relative URL.

Relative URL:

```
cz.uiservlet.formtemplateurl=/OA_HTML/US/czFormTemplate.htm
```

Syntax of fully qualified path:

```
cz.uiservlet.formtemplateurl=http://host:port/html_vpath/US/czFormTemplate.htm
```

Example of fully qualified path:

```
cz.uiservlet.formtemplateurl=http://www.mysite.com:8802/OA_
HTML/US/czFormTemplate.htm
```

### **cz.uiservlet.jdbcdriver**

Sets the class name of the JDBC driver for connecting the OC Servlet to a database.

There is no default setting. You must set this parameter.

Syntax:

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

Example:

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

### **cz.uiservlet.logfilename**

The path and file name prefix for normal servlet and configuration session logging files. Do not specify a specific file name. The logs will automatically be prefixed with the prefix you specify. If this parameter is omitted, no logging files will be written.

The servlet log file records servlet startup and shutdown, the system and servlet properties passed to it, and the classpath used. It is named with the convention:

```
prefix-servlet_date_time_processid.txt
```

Example:

```
cz-servlet_03Jan01_1056AM_23193.txt
```

The configuration session log file records the initialization messages sent to the OC Servlet, the HTML template used, XML events, and JavaScript rendering. It is named with the convention:

```
prefix-session_servletname_sessionid_date_time_processid.txt
```

Example:

```
cz-session-myservlet_DHTML2_03Jan01_1058AM_30519.txt
```

The value of *servletname* is set by [cz.uiservlet.name](#) on page 2-34.

The output for *sessionid* is set by the UI Server. It is composed of an ID for the type of session (DHTML, APPLET, BatchSingle, or returnui) appended with a counter. Example: DHTML2.

(Exception log file names and locations are controlled by the property [cz.uiservlet.logfilename](#) on page 2-33.)

Syntax:

```
cz.uiservlet.logfilename=logging_dir/prefix
```

Examples for Solaris™ Operating Environment:

```
cz.uiservlet.logfilename=jserv_install/logs/server_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 session files written to the location specified by [cz.uiservlet.logfilename](#) on page 2-33.

Syntax:

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

Example:

```
cz.uiservlet.name=myservlet
```

### **cz.uiservlet.pre\_load\_filename**

Absolute path to a file containing an initialization message for the OC Servlet. This file is read if you are preloading the servlet. In order to preload a servlet with Apache, you must specify its class name as the value of the parameter `servlets.startup` in the file `zone.properties`. Example for setting `servlets.startup`:

```
servlets.startup=oracle.apps.cz.servlet.UiServlet
```

Syntax for this property:

```
cz.uiservlet.pre_load_filename=absolute_path_to_init_file
```

The contents of *init\_file* is a valid Oracle Configurator initialization message, the construction of which is described in detail in the *Oracle Configurator Custom*

*Web Deployment Guide.* Make sure that each initialization message in *init\_file* is completely on a single line, with no line breaks in the message text. There can be multiple initialization messages in the file, but each message must be on its own line.

Example for an initialization message:

```
<initialize>
<param name="database_id">dbc_filename</param>
<param name="gwyuid">applsypub/pub</param>
<param name="user">apps</param><param name="pwd">apps</param>
<param name="ui_type">DHTML</param>
<param name="context_org_id">5</param>
<param name="model_id">1234</param>
<param name="ui_type">calling_application_id">671</param>
</initialize>
```

Example for setting this property:

```
cz.uiservlet.pre_load_filename=/home/apache/init_msg.txt
```

---



---

**Note:** This property provides a performance enhancement, by caching a Model at servlet startup. However, this enhancement is counteracted if you disable model caching, by setting [cz.uiservlet.dio\\_share](#) to `false`. You must also consider the interaction of this property with [cz.uiserver.lazyload](#) on page 2-28.

---



---

### cz.uiservlet.proxyscript

URL of JavaScript source file for the Proxy Frame. See the *Oracle Configurator Custom Web Deployment Guide* for details about the Proxy Frame.

This property is not required in `zone.properties`. You may want to add this property if, for example, you are defining a servlet that displays HTML pages in a different language. If you define this property, you can specify its location either by entering a fully qualified path or a relative URL. The recommended method is using a relative URL.

Relative URL:

```
cz.uiservlet.proxyscript=/OA_HTML/czProxy.js
```

Syntax of fully qualified path:

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

Example of fully qualified path:

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

### **cz.uiservlet.templateurl**

URL of the HTML template for a DHTML client running in a Netscape browser. See the *Oracle Configurator Custom Web Deployment Guide* for details about HTML templates.

This property is not required in `zone.properties`. You may want to add this property if, for example, you are defining a servlet that displays HTML pages in a different language. If you define this property, you can specify its location either by entering a fully qualified path or a relative URL. The recommended method is using a relative URL. If this property is not set in `zone.properties`, the servlet uses the URL defined for [cz.uiservlet.blafteurl](#).

Relative URL:

```
cz.uiservlet.templateurl=/OA_HTML/US/czFraNS.htm
```

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.url**

URL of the Oracle Configurator Servlet.

Syntax:

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

Example:

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

### **cz.uiservlet.versionfuncsavail**

Use this property to determine whether the servlet responds to any message entered in a Web browser (for example, a test message). The default value of this property is `True`.

For more information, see [Section 3.2, "Checking the Response of the UI Servlet"](#) on page 3-1.

**Syntax:**

```
cz.uiservlet.versionFuncsAvail=[true|false]
```

**Example:**

```
cz.uiservlet.versionFuncsAvail=true
```



---

## 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, "Oracle Configurator 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 czjni in shared library path
```

- Make sure that your database configuration file includes the batch validate username and password parameters. See [Section 1.4.1, "Modify Database Configuration File Parameters"](#) on page 1-4.

### 3.2 Checking the Response of the UI Servlet

#### What you are checking

Does the UI Servlet respond to a test message?

#### The test

Invoke this URL in a Web browser:

```
http://hostname:portnum/configurator/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 in [Table 3-1](#) for different values of *test\_string*.

---



---

**Note:** The property `cz.uiservlet.versionfuncsavail` determines whether you can test the response of the servlet using a test string. See [Section 2.4.2, "Descriptions of Oracle Configurator Servlet Properties"](#) on page 2-20.

---



---

**Table 3-1 Test Strings and Results**

Value of <i>test_string</i>	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 <i>hostname</i> and <i>portnum</i> specified in your test URL.
<i>any_other_string</i>	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: 11.5.4.15.55
Expecting schema: 15i
```

**Example:**

```
http://www.mysite.com:8802/configurator/oracle.apps.cz.servlet.UiServlet?test=hello_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.
- There may be a basic problem with your configuration of Apache. See [Section 3.4, "Checking the Operation of the Apache Internet Server"](#) on page 3-5.
- `apps.zip` may not be in your class path.
- The JServ engine may not have been started up.

## 3.3 Checking Your Model in the Runtime Oracle Configurator

**What you are checking**

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

---



---

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

---



---

**The test**

You can test the behavior of the DHTML runtime Oracle Configurator 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 3-1](#), and the *Oracle Configurator Custom Web Deployment Guide* for examples of simple test pages.

**Example 3-1 HTML Test Page for Invoking the DHTML runtime Oracle Configurator**

```
<html>
<head>
<title>Minimal Configurator Test</title>
```

```
</head>
<body>
<form
action="http://www.mysite.com:8802/configurator/oracle.apps.cz.servlet.UiServlet
" method="post">
<input type="hidden" name="XMLmsg" value=
'<initialize>
  <param name="database_id">serv01_sid02</param>
  <param name="ui_type">DHTML</param>
  <param name="ui_def_id">3120</param>
</initialize>'>

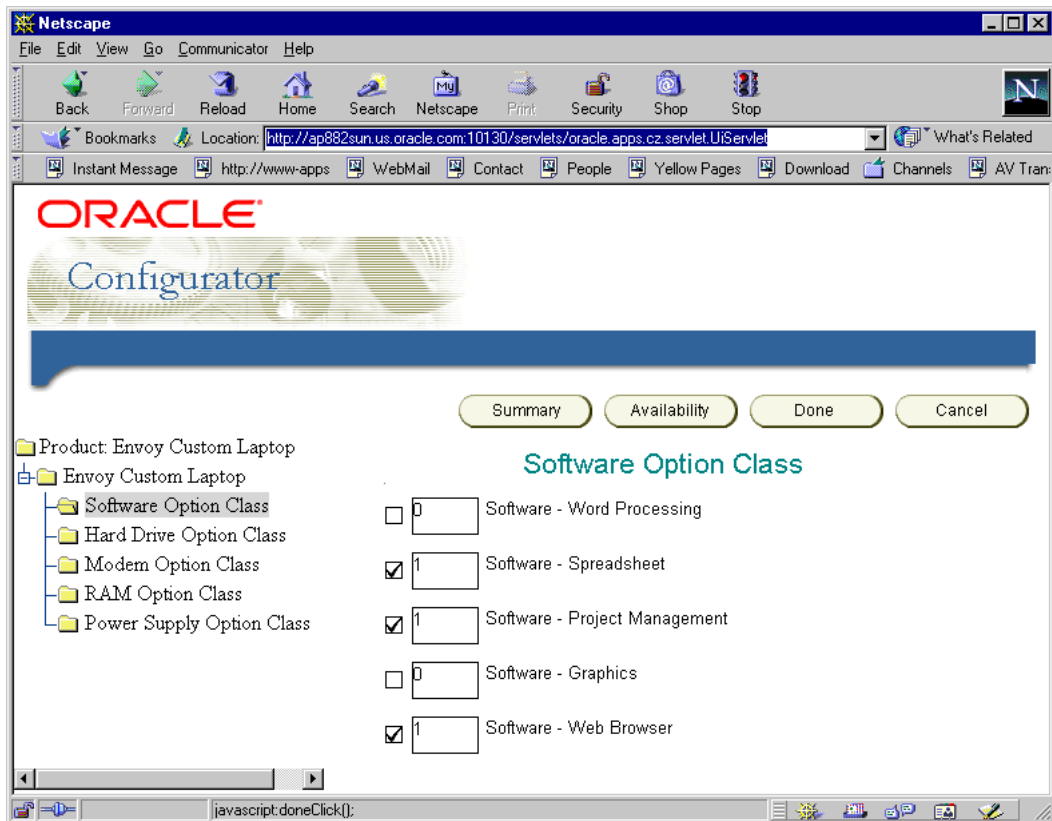
<p>Click the button to configure the model...
<input type="submit" value="Configure">
</form>
</body>
</html>
```

2. Ensure that you have the necessary database connectivity, and that your UI Servlet is installed and configured correctly. See [Chapter 2, "Oracle Configurator Servlet Considerations"](#).
3. Test the runtime Oracle Configurator 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 3-1](#), click the button to produce the User Interface.

See [Figure 3-1](#) on page 3-5 for a general example of the appearance of the DHTML runtime Oracle Configurator.

Figure 3-1 A Model, in the Runtime Oracle Configurator



## 3.4 Checking the Operation of the Apache Internet Server

### Test 1 - What you are checking

Does your Apache internet server respond at all?

### The test

1. Compile the code in [Example 3-2](#) into the file `Hello.class` in your servlets directory.
2. In a Web browser, invoke this URL:

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

where *hostname* is the server that you have installed on, *portnum* is the port number configured for the HTTP listener, and *servlet\_vpath* is the mounting location specified by `ApJServletMount` in the file `Jserv.conf` (this is usually configurator). For example:

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

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

**Example 3–2 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.

### Test 2 - What you are checking

Is the hostname and port valid?

#### The Test

In a Web browser, invoke this URL:

```
http://hostname:portnum/
```

If the test is successful, the "Oracle Applications Rapid Install Portal" page appears. This indicates that Apache is working and that the hostname and port are valid. If this test works, but the first test did not, then there could be problems with the JServ/JVM configuration. Check the mount point (ApJServMount directive) in `jserv.conf` and look for errors in the `mod_jserv_log` and `jserv_log` files. Additionally, confirm that Java is setup correctly (change directory to the Java interpreter path specified in `jserv.properties` and enter "java -version").

If this test returns an error, there are problems with Apache on the machine. Check the `error_log`, make sure `httpd.conf` (or `httpds.conf`) is set up properly, and make sure the hostname is a resolvable DNS entry.

### Test 3 - What you are checking

Does the `/OA_HTML/` alias exist and, if so, is it valid?

#### The Test

In a Web browser, invoke this URL:

```
http://hostname:portnum/OA_HTML/czxml2js.xml
```

Verify that a page can be pulled up (it will look like HTML source code). If this test fails, be sure that:

- The `/OA_HTML/` alias is set up correctly (in `httpd.conf` or `httpds.conf` or `oracle_apache.conf` or `apps.conf`)
- There is no `OA_HTML` directory under the `DocumentRoot` (`httpd.conf` or `httpds.conf`)

### Check Apache Configuration Files

If these tests do not resolve the problem, check the following Apache/JServ configuration files.

- ❑ In **JServ.conf**, check the settings of the following parameters and modify them as required:
  - **ApJServMount**: This parameter specifies the mount points for servlets zones. Be sure the following entry exists:

```
ApJServMount /configurator /root
```

See [Section 2.1.2, "Verifying jserv.conf"](#) on page 2-4.

- **ApJServVMTimeout**: This parameter indicates amount of time (in seconds) to give to the JVM to start up as well as the amount of time to wait to "ping" the JVM to see if it is active. For slow or heavily loaded machines, you may want to increase this value. The default is 10 seconds (used when the line is preceded by a "#" symbol).

If the `mod_jserv.log` shows the JVM continually being spawned, exited, and restarted, increase this value to 300, restart Apache, and test again. If the problem goes away, it means the server is too heavily loaded. This is most often seen on single tier systems.

For example:

```
ApJServVMTimeout 300
```

- **ApJServManual**: This determines whether Apache will automatically start Apache JServ and Java Virtual Machine (JVM). If this is commented out (line is preceded by a "#") or set to off, then Apache will automatically start Apache JServ using the parameters specified in `jserv.properties`.

If this parameter is set to on, then Apache JServ must be started manually via shell scripts. In this case, the parameters set in `jserv.properties` will *not* be used, and must be set in the scripts used to start the JVM. This parameter is set to on when load balancing is implemented. See [Section 2.2, "Load Balancing"](#) on page 2-8.

- In the **JServ.properties** file, check the following and make changes as required:
  - The path of the Java interpreter (java.exe or jre.exe) - Near the top of the file (just under "# The Java Virtual Machine interpreter") there is a line that begins with `wrapper.bin`. This must point to the full path of the Java executable (Java or JRE). This is usually `apache_install/jdk/bin/java`, where `apache_install` is replaced by the actual path.

For example:

```
wrapper.bin=/u09/oracle/test1ora/iAS/Apache/jdk/bin/java
```

Note that the JRE or JDK (Java) can be used in many cases, but in other cases only the JDK can be used (for example, Oracle CRM). Rapid Install installs the JDK on the Solaris™ Operating Environment, but it is a manual process on other platforms.

- You must add the following parameters if you just upgraded to version 16.x of Oracle Configurator Developer:
  - [cz.uiservlet.blaftemplateurl](#)
  - [cz.uiservlet.formtemplateurl](#)
  - [cz.html.source.formtreeview](#)

The value of `ApJServManual` determines *where* these parameters must be set:

- If `ApJServManual` is set to `off` (or the line is commented out) in `JServ.conf`, add the parameters to the `Jserv.properties` file.
- If `ApJServManual` is set to `on`, add the parameters to the `zone.properties` file.

For more information about these properties, see [Section 2.4, "Oracle Configurator Servlet Properties"](#) on page 2-17.

- Verify that the Oracle Applications profile option BOM:Configurator URL of UI Manager has the same value as the `cz.uiservlet.url` parameter. (All Configurator Developer parameters in `JServ.properties` begin with `wrapper.bin.parameters=-Dcz.`) Refer to the *Oracle Applications System Administrator's Guide* for information about setting profile options.
- The Shared Library Path - This variable points to the directory that contains the Configurator shared library(s). This path is always `$CZ_TOP/bin`. The shared library path variable name is platform-dependent. [Table 3-2](#) lists the library path name for each operating system.
- The CLASSPATH - See [Section 2.1.3, "Verifying jserv.properties"](#) on page 2-5.

**Table 3–2 Library Path Variable Names by Operating System**

<b>Operating System</b>	<b>Variable Name</b>
Solaris™ Operating Environment	LD_LIBRARY_PATH
HP	SHLIB_PATH
AIX	LIBPATH
NT	PATH

For example, if the operating system is the Solaris™ Operating Environment:

```
wrapper.env=LD_LIBRARY_PATH=/u012/oracle/crplappl/cz/11.5.0/bin
```

---

---

**Note:** After running Oracle Rapid Install, the shared object files may need to be relinked with `adrelink`.

---

---

---



---

## 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 Developer, uninstall all components before installing the new version.
- Ensure that you have release 10.7, 11.01, 11.0.2, 11.0.3, or 11*i* of Oracle Applications installed and configured.
- For all development installations, ensure that Oracle8*i* Enterprise Edition (or higher) is installed on the server machine and Oracle8 Client is installed on the client machine.
- For Web deployment or development testing using the DHTML or JAVA applet, install a supported internet server and the OC servlet. See "[System Requirements](#)" on page 1-2 and [Chapter 2, "Oracle Configurator Servlet Considerations"](#) for more information.

### 4.1 The Oracle Configurator Developer Software

[Table 4-1](#) lists the software included with Oracle Configurator Developer Release 11*i*.

**Table 4-1 Oracle Configurator Developer Software**

Product	Revision	Level	License
Oracle Configurator Developer Installation	11 <i>i</i>	Production	Yes
Oracle Configurator Developer	11 <i>i</i>	Production	Yes

**Table 4–1 Oracle Configurator Developer Software**

Product	Revision	Level	License
Oracle Configurator Documentation (doc folder)	11i	Production	No
SQL *Plus Migration Scripts (Migration folder)	11i	NA	No
Oracle Client	8.0.6	NA	No
Adobe Acrobat Reader	4.0	NA	No
DCOM98	1.0	NA	No
MDAC	2.1	NA	No

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

- The `Migration` folder contains SQL scripts that enable you to migrate from Oracle SellingPoint 4.2.2 or later to the Oracle Configurator schema version 14e (Release 11i of Oracle Applications, version 11.5.3 or 11.5.4). Note that your Oracle SellingPoint schema must be 14e before migrating. See [Section 1.6, "Migrating From a Standalone Schema"](#) on page 1-11.
- 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 Implementation 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 Installing Oracle Configurator Developer

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%\OC\).
5. If you are running Windows 95/98, a dialog box appears prompting you to install DCOM98 from the Oracle Configurator Developer compact disc.
6. 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**.
7. 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.
8. An spx.ini file was installed on your machine. This file contains default settings and 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 4.4.2.6, "Set Parameters in the spx.ini File for Development and Testing"](#) on page 4-12 and the *Oracle Configurator Implementation Guide* for details on setting parameters in the spx.ini file.

To run Oracle Configurator Developer and a runtime Oracle Configurator, see "[Test Your Oracle Configurator Developer Installation](#)" on page 4-17.

## 4.3 Installing Oracle Configurator Developer Patches

After your initial install of Oracle Configurator Developer is completed, you can apply the latest patch or an interim release to your Oracle Configurator Developer environment. To obtain the latest Oracle Configurator Developer patches or interim releases, create a shipping TAR requesting the desired version through Metalink at <http://metalink.oracle.com> or any other standard Oracle Support method.

Follow these steps to install Oracle Configurator Developer patch:

1. Exit all other applications.
2. Uninstall any previously installed versions of Oracle Configurator Developer.
3. Unzip the patch .zip file.

4. From the **Start** menu, run OCSetup.exe.
5. Follow steps 4. through 8. of the instructions in "[Installing Oracle Configurator Developer](#)" on page 4-2.
6. Stop and then restart the Apache Web server.

## 4.4 Oracle Configurator Developer Client/Server Environments

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.

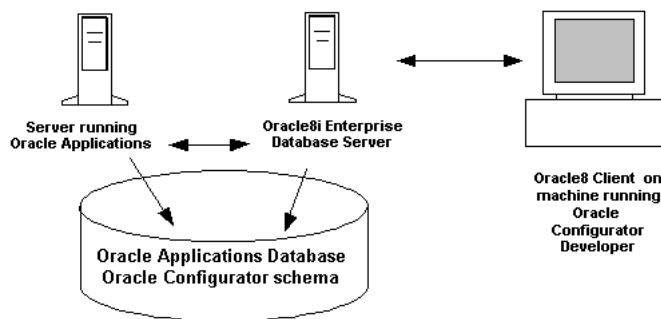
A runtime Oracle Configurator and Oracle Configurator Developer typically run on client machines connected to the Oracle Applications database server. A runtime Oracle Configurator embedded in other Oracle Applications, such as Order Management, runs on the application server machine where the calling application runs. A runtime Oracle Configurator embedded in other Oracle Applications (such as Order Management and *iStore*) or in a custom Web application runs on the internet server machine.

### 4.4.1 Server Environment for Oracle Configurator Developer

There are a number of servers included in the Oracle Configurator Developer environment:

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

[Figure 4-1](#) 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 database server for access by Oracle Configurator Developer and test runtime Oracle Configurators. This is 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 Oracle8i Enterprise Edition installed and the client machine(s) must have Oracle8 Client installed.

The database server setup includes the following tasks:

- Install Oracle8i Enterprise Edition first
- Install the Oracle Applications database
- Create Users and Responsibilities (see ["Establish Users and Responsibilities"](#) on page 4-5)
- Set Oracle Configurator Developer database settings (see [Section 4.4.2.6, "Set Parameters in the spx.ini File for Development and Testing"](#) on page 4-12)

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 Implementation Guide*.

#### 4.4.1.1 Establish Users and Responsibilities

An Oracle Configurator Developer user is established through Oracle Applications administration and resides in the Oracle Applications database. Each user must also be defined as a database user in the Oracle8i Enterprise Edition server database

running the Oracle Configurator schema and be assigned an Oracle Configurator responsibility (Configurator Administrator or Configurator Developer).

You can log in to Oracle Configurator Developer as an Oracle Applications user assigned either the Configurator Administrator or Configurator Developer responsibility. 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 [Section 4.4.2.6, "Set Parameters in the spx.ini File for Development and Testing"](#) on page 4-12. See the *Oracle Configurator Implementation Guide* for more information on these and other spx.ini parameters.

#### **4.4.1.2 Users and Responsibilities for Accessing Runtime Oracle Configurators**

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

#### **4.4.1.3 Running Oracle Configurator Developer Using a Terminal Server**

Because Configurator Developer is a database-intensive application, the recommended configuration is to run the application over a local area network (LAN). However, your organization may need to work over a remote or wide area network (WAN). To run Configurator Developer on a remote database server while maintaining an acceptable level of performance requires the use of terminal server emulation software.

This software provides remote access to a server desktop and can run on a number of client hardware devices, such as a personal computer. Using a terminal server program minimizes network bandwidth requirements between the database server and client (Configurator Developer) and enables multiple users to run the application simultaneously.

To install Configurator Developer to run on a terminal server, be sure that:

- No one else is logged in to the system during installation
- Installation is performed by the system administrator or a user with the Administrator responsibility
- The installation process is performed using the "Add and Remove Programs" option from the Windows Control Panel

To install Configurator Developer for use with terminal emulation software, perform the following:

1. Log in to the terminal server machine as the System Administrator, then choose **Start > Settings > Control Panel > Add/Remove Programs**.
2. If Oracle Configurator Developer appears in the list of installed applications, uninstall the application.
3. Click **Install**.
4. Insert the Oracle Configurator Developer compact disc or navigate to the location of the file OCSetup.exe, then continue with the installation.
5. When prompted, choose to enable the application for **All Users**.
6. If this is the initial Oracle Configurator Developer installation, click **Yes** when prompted to install the Microsoft Data Access Components (MDAC); otherwise, click No.
7. When the installation is complete, modify the default spx.ini file as required. See [Section 4.4.2.6, "Set Parameters in the spx.ini File for Development and Testing"](#) on page 4-12 and the *Oracle Configurator Implementation Guide* for details on setting parameters in the spx.ini file.
8. Copy the modified spx.ini file to each user's /Windows directory on the terminal server.

For example:

```
/user_folder/windows/spx.ini
```

## 4.4.2 Client Environment for Oracle Configurator Developer

Installing Oracle Configurator for an implementation, test, or maintenance environment consists of installing Oracle Configurator Developer. Oracle Configurator Developer runs on a client machine with Oracle8 Client for:

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

### 4.4.2.1 Oracle8 Client Installation

Oracle8 Client allows the Oracle Configurator Developer user to access an Oracle8i Enterprise Edition database from a client machine not running Oracle8i Enterprise Edition. Networked client machines must have Oracle8 Client installed if they access the Oracle8i Enterprise Edition Server database.

#### 4.4.2.2 Runtime Oracle Configurator Client Requirements

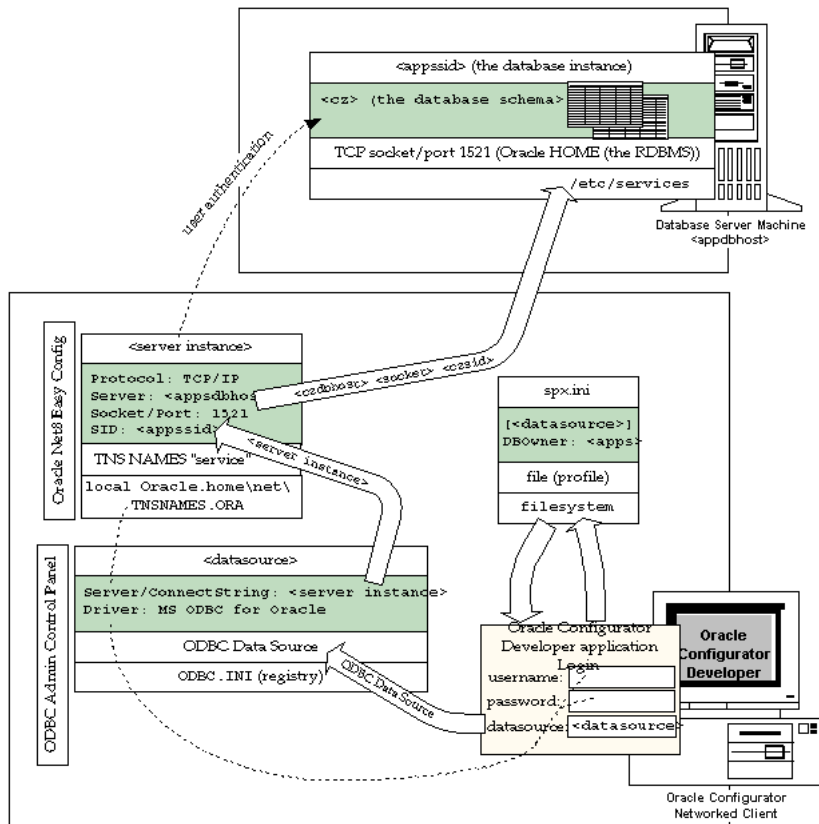
When unit testing the UI (Java applet or DHTML) from Oracle Configurator Developer using the Test button, the URL for the DHTML and Java applet Servlet UI and JDBC thin client parameter (for DHTML only) must be specified in the spx.ini file. See [Section 4.4.2.6, "Set Parameters in the spx.ini File for Development and Testing"](#) on page 4-12.

#### 4.4.2.3 Set Up Oracle Configurator Developer

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:

- Oracle8 Client is installed on the client machine that is running Oracle Configurator Developer. See ["Oracle8 Client Installation"](#) on page 4-7.
- The client machine is configured to connect to the database server (Oracle Net8 Easy Config). See ["Enable the Client for Database Connectivity"](#) on page 4-9.
- A datasource name (DSN) for the Oracle Configurator schema on the server machine is defined in ODBC Administrator on the client machine. See ["Create DSNs and DBOwners"](#) on page 4-10.
- The user logged into Oracle Configurator Developer 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 4.4.2.6, "Set Parameters in the spx.ini File for Development and Testing"](#) on page 4-12.
- Test your Oracle Configurator Developer installation. See ["Test Your Oracle Configurator Developer Installation"](#) on page 4-17.

[Figure 4-2](#) 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**

#### 4.4.2.4 Enable the Client for Database Connectivity

Each machine running Oracle Configurator Developer must be configured to connect to an Oracle8i 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 Oracle8 Client. To establish data connectivity on the client machine, you need to know the following parameters:

- The name and of the server database instance or system identifier (SID) where the Oracle Configurator schema is located
- The host name and port number of the database server
- The username and password for the Oracle Configurator schema

You must set these parameters using Oracle Net8 Easy Config to establish data connectivity. To establish data connectivity, perform the following:

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 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.

#### 4.4.2.5 Create DSNs and DBOwners

Create ODBC datasource names (DSNs) for each Oracle8i Enterprise Edition server that you need for a development, test, or maintenance installation.

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

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

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 window.
2. Select the **System DSN** tab.
3. Click **Add**. This opens the Create New Data Source dialog.
4. Select **Microsoft ODBC for Oracle** for setting a server DSN, then click **Finish**.
5. The Microsoft ODBC for Oracle Setup dialog appears. Enter the name of the database you want to access (including the extension, such as .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, but this is usually not necessary. These options include:

Translation: Click **Select** to choose a loaded data translator. The default is No Translator.

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.

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.

#### 4.4.2.6 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 in Oracle Configurator Developer. An `spx.ini` file is not required for deployments of a runtime Oracle Configurator.

When installing Oracle Configurator Developer, a default `spx.ini` file is copied to the `Winnt` directory (for Windows NT machines) or the `Windows` directory (for Windows 95/98 machines).

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

- Oracle Configurator Developer
- Test instances of a runtime Oracle Configurator from within Oracle Configurator Developer (Test)

Throughout this section, references to the test configurator mean test instances of a runtime Oracle Configurator launched from within Oracle Configurator Developer (Test).

Oracle Configurator Developer requires 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 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 "[Create DSNs and DBOwners](#)" on page 4-10; the spx.ini entries will not work until you create the DSN.

For each data source identified in the [DSN] section in the spx.ini file, you can optionally set values for the test parameters that differ from one data source to the next. For more information, see the *Oracle Configurator Implementation Guide*.

You must minimally update the following spx.ini entries to start up and use Oracle Configurator Developer and a test configurator. See the *Oracle Configurator Implementation Guide* for more information about these and deployment parameters.

**Table 4-2 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.
DSN		List of datasource names of the databases used for development, testing, or maintenance.  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.
<i>dsn</i>	DBOwner	The datasource name of the Oracle Configurator schema used with Oracle Configurator Developer must be listed here. the DBOwner in this section is the username used to access the Oracle Configurator schema.  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.

**Table 4–2 (Cont.) Spx.ini Parameters for Development and Testing**

Section	Parameter	Description
	JdbcUrl	<p>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. Use this format</p> <pre>jdbc:oracle:thin:@host:port:sid</pre> <p>In this example, <code>host</code> is the name of the local machine, <code>port</code> is the port where your service is running, and <code>sid</code> is your server name.</p> <p>You must specify the JDBC connection URL in the section corresponding to the current data source. Example:</p> <pre>[Test11] JdbcUrl=jdbc:oracle:thin:@host:1521:Test11</pre>
	gwyuid	The public Oracle gateway username that grants access to the Oracle Applications log on window. Gwyuid and gwypass should be the same as the default username and password in your Oracle Applications environment file.
	gwypass	The public Oracle gateway password that grants access to the Oracle Applications log on window.
	Launch	<p>Sets the type of environment to launch when using the Test button in Oracle Configurator Developer. <code>Launch=1</code> specifies the Dynamic HTML in a browser.</p> <p>When <code>Launch=1</code> is specified, the parameter <code>InitServletURL</code> must also be set to specify the URL of the servlet generating the Dynamic HTML in a browser.</p> <p><code>Launch=2</code> specifies the Java Applet. When <code>Launch=2</code> is specified, the parameter <code>InitServletURL</code> must also be set to specify the URL of the servlet generating the Java Applet. See the <i>Oracle Configurator Implementation Guide</i> for additional parameters that must be set for the Oracle Configurator Developer test environment.</p>
	InitCodeBaseUrl	The URL of the servlet generating the Java Applet for testing.
	InitServletUrl	The URL of the location of the class or Java archive files for testing with the Java Applet.

**Table 4–2 (Cont.) Spx.ini Parameters for Development and Testing**

Section	Parameter	Description
	jInitVersion	The version of Jinitiator that is installed on the client machine running Oracle Configurator Developer. The default version is 1.1.8.7.
	jInitClassID	The Jinitiator class identifier. If you modify the value of jInitVersion in your spx.ini file, this value must be changed accordingly. To determine the correct value for this parameter, see /ORACLE_HOME/jinitversion/doc/jinit-version.txt on the client machine running Configurator Developer .
	APPLID	The client application identifier. Used with FND_USER_ID to generate icx_session_ticket information for the database session.
	RESPID	Associated with the APPLID, this identifies the Oracle Configurator user's responsibility. This parameter is also used to generate icx_session_ticket information for the database session.

## 4.5 Parameterized Startup of Oracle Configurator Developer

You can also start up Oracle Configurator Developer using predefined parameters. Using this method you provide preset values for the mandatory login parameters (user, password, data source name), and bypass the Configurator Developer login screen where you would normally enter them. You can also control logging with a parameter.

### 4.5.1 Syntax

The syntax for the parameterized startup of Oracle Configurator Developer is:

```
[dev_path]\SPDeveloper /usr=username /pwd=password /dsn=datasource
[/log(n)+=filename]
```

The order of the parameters is not significant. The parameters are not case-sensitive.

Example:

```
SPDeveloper /usr=apps /pwd=apps /dsn=vis01 /log(2)+=oc_log.out
```

The startup parameters are defined in [Table 4–3, "Startup Parameters for Oracle Configurator Developer"](#) on page 4-16.

**Table 4–3 Startup Parameters for Oracle Configurator Developer**

Parameter	Value	Definition
	<i>dev_path</i>	The path to the directory in which OCD is installed. Typically, this is:  [ORACLE_HOME]\OSP\Developer  This is not a formal startup parameter, but may be necessary if you are starting up from a directory other than the one in which OCD is installed.
/usr	<i>username</i>	These parameters are the same as those that you define for starting OCD from the Windows Start menu. For details, see <a href="#">Section 4.4.2, "Client Environment for Oracle Configurator Developer"</a> on page 4-7.
/pwd	<i>password</i>	
/dsn	<i>datasource</i>	
/log		This parameter is optional. Any combination of its following variant forms may be used.
/log( <i>n</i> )	<i>n</i>	<i>n</i> is a number indicating the logging level, as listed in the Value column in <a href="#">Table 4-4</a> on page 4-17.
/log+	+	+ makes the log window visible.
/log= <i>fil</i>	<i>filename</i>	<i>filename</i> is the name of a log output file. By default, the file is written to <i>dev_path</i> . You can provide a path to write it to another location. This value overrides the default value specified on the Log tab of the Configurator Developer Options window (Tools > Options).
	<i>ename</i>	

## 4.5.2 Usage

You can enter the parameterized startup command in several locations:

- Directly on the command line in a Windows command prompt window
- In a Windows batch file
- In the Target field of a Windows shortcut to OCD, which will probably require providing a value for *dev\_path*, such as:

```
C:\orant\OSP\Developer\SPDeveloper /usr=apps /pwd=apps /dsn=vis01
/log(2)+=oc_log.out
```

### 4.5.3 Logging

You can determine the level of logging messages that are written to the Log Messages window and the log file, by setting the value of `n` in the `/log(n)` parameter. See [Table 4-3, "Startup Parameters for Oracle Configurator Developer"](#) on page 4-16.

The logging levels are listed in [Table 4-4](#), in order of increasing severity. Selecting a logging level includes that level and *all higher levels* in the log. For example, selecting DetailTrace includes all possible logging messages; selecting Error includes Error, DataDamaged, and Fatal logging messages. The Level Names correspond to the options on the Settings > Report Settings submenu of the Log Messages window. The Log Messages window is described in the *Oracle Configurator Developer User's Guide*.

**Table 4-4 Logging Levels for Oracle Configurator Developer**

Type	Value	Level Name
Information	-10	DetailTrace (the default)
	-9	Info
Warning	0	Empty/EOF
	1	Notification
	2	Warning
Serious	10	Error
	11	Data Damaged
	12	Fatal

## 4.6 Test Your Oracle Configurator Developer Installation

Run Oracle Configurator Developer to test the installation and connectivity. Before you do this, read the *Oracle Configurator Release Notes*, especially the section called "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.

---

---

---

---

**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**.

---

---

To test your Oracle Configurator Developer installation:

1. Start Oracle Configurator Developer and log in as the DBOwner.
2. To run Oracle Configurator Developer, click the Windows **Start** button, then choose **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
3. Log into Oracle Configurator Developer using the username specified in the spx.ini file (DBOwner) or a valid Oracle Applications username, and a password that has been assigned an Oracle Configurator responsibility (Configurator Administrator or Configurator Developer).
4. From the **Datasource** list, select the DSN for the database you want to use. The DSNs in the list are specified in the spx.ini file (see *Oracle Configurator Implementation Guide* for details).
5. Click **OK**. The Repository window appears.
6. To create a new Model, choose **Create > New Model**, then enter a name and description in the New Model dialog. If the database you chose contains existing Models, open the Model you want to modify by double clicking on it.

For information about constructing models and building configuration rules, see the *Oracle Configurator Developer User's Guide*.

7. To run the test configurator from Oracle Configurator Developer:
  - a. In the Model module, choose **Tools > Generate Active Model**.
  - b. In the UI module, choose **Create > New User Interface**.
  - c. Click the **Test** button.



---

---

# Glossary of Terms and Acronyms

This glossary contains definitions that you may need while working with Oracle Configurator.

## **Active Model**

The compiled structure and rules of a **configuration model** that is loaded into memory on the Web server at **configuration session** initialization and used by the **Oracle Configurator engine** to validate runtime selections. The Active Model must be generated either in **Oracle Configurator Developer** or programmatically in order to access the configuration model at **runtime**.

## **API**

Application Programming Interface

## **applet**

A Java application running inside a Web browser. *See also* **Java** and **servlet**.

## **application architecture**

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

## **architecture**

*See* **application architecture**.

## **ATO**

Assemble to Order

**ATP**

Available to Promise

**attribute**

The defining characteristic of something. Models have attributes such as Effectivity Sets and Usage. Components, Features, and Options have attributes such as Name, Description, and Effectivity.

**benchmark**

Represents performance data collected during **runtime** tests under various conditions that emulate expected and extreme use of the product.

**beta**

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

**bill of material**

A list of Items associated with a parent Item, such as an assembly, and information about how each Item relates to that parent Item.

**Bills of Material**

The application in Oracle Applications in which you define a **bill of material**.

**BOM**

See **bill of material**.

**BOM item**

The **node** imported into **Oracle Configurator Developer** that corresponds to an Oracle **Bills of Material** item. Can be a **BOM Model**, **BOM Option Class node**, or **BOM Standard Item node**.

**BOM Model**

A model that you import from Oracle **Bills of Material** into **Oracle Configurator Developer**. When you import a BOM Model, effective dates, **ATO** rules, and other data are also imported into Configurator Developer. In Configurator Developer, you can extend the structure of the BOM Model, but you cannot modify the BOM Model itself or any of its **attributes**.

**BOM Model node**

The imported **node** in **Oracle Configurator Developer** that corresponds to a **BOM Model** created in Oracle **Bills of Material**.

**BOM Option Class node**

The imported **node** in **Oracle Configurator Developer** that corresponds to a BOM Option Class created in Oracle **Bills of Material**.

**BOM Standard Item node**

The imported **node** in **Oracle Configurator Developer** that corresponds to a BOM Standard Item created in Oracle **Bills of Material**.

**Boolean Feature**

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 program 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 (CIO)**.

**client**

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

**Comparison rule**

An **Oracle Configurator Developer** rule type that establishes a relationship to determine 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 that establishes a relationship among **Features** in the Model to control the allowable combinations of **Options**. *See also, Property-based Compatibility rule.*

**Compatibility Table**

A kind of Explicit Compatibility rule. For example, a type of compatibility relationship where the allowable combination of **Options** are explicitly enumerated.

**component**

A piece of something or a configurable element in a **model** such as a **BOM Model, Model**, or **Component**.

**Component**

An element of the **model structure**, typically containing **Features**, that is configurable and instantiable. An **Oracle Configurator Developer** node type that represents a configurable element of a **Model**. Corresponds to one UI screen of selections in a runtime **Oracle Configurator**.

**Component Set**

An element of the **Model** that contains a number of instantiated **Components** of the same type, where each Component of the set is independently configured.

**concurrent manager**

A process manager that coordinates the **concurrent processes** generated by **users' concurrent requests**. An Oracle Applications product group can have several concurrent managers.

**concurrent process**

A task that can be scheduled and is run by a **concurrent manager**. A concurrent process runs simultaneously with interactive functions and other concurrent processes.

**concurrent processing facility**

An Oracle Applications facility that runs time-consuming, non-interactive tasks in the background.

**concurrent program**

Executable code (usually written in SQL\*Plus or Pro\*C) that performs the function(s) of a requested task. Concurrent programs are stored procedures that

perform actions such as generating reports and copying data to and from a database.

**concurrent request**

A user-initiated request issued to the concurrent processing facility to submit a non-interactive task, such as running a report.

**configuration**

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

**configuration attribute**

A characteristic of an **item** that is defined in the **host application** (outside of its inventory of items), in the **Model**, or captured during a **configuration session**. Configuration attributes are inputs from or outputs to the host application at initialization and termination of the configuration session, respectively.

**Configuration Interface Object**

See **Oracle Configuration Interface Object (CIO)**.

**configuration model**

Represents all possible configurations of the available **options**, and consists of **model structure** and **rules**. It also commonly includes **User Interface** definitions and **Functional Companions**. A configuration model is usually accessed in a **runtime Oracle Configurator window**. See also **model**.

**configuration rules**

The **Oracle Configurator Developer Logic rules**, **Compatibility rules**, **Comparison rules**, **Numeric rules**, and **Design Charts** available for defining **configurations**. See also **rules**.

**configuration session**

The time from launching or invoking to exiting **Oracle Configurator**, during which **end users** make selections to configure an orderable product. A configuration session is limited to one **configuration model** that is loaded when the session is initialized.

**configurator**

The part of an application that provides custom configuration capabilities. Commonly, a window that can be launched from a hosting application so **end users**

can make selections resulting in valid **configurations**. *Compare* **Oracle Configurator**.

### **connectivity**

The connection between **client** and database **server** that allows data communication.

The connection across components of a model that allows modeling such products as networks and material processing systems.

### **Connector**

The **node** in the **model structure** that enables an **end user** at **runtime** to connect the Connector node's parent to a referenced **Model**.

### **context**

The surrounding text or conditions of something.

Determines which context-sensitive segments of a flexfield in the Oracle Applications database are available to an application or **user**. Used in defining **configuration attributes**.

### **Contributes to**

A relation used to create a specific type of **Numeric rule** that accumulates a total value. *See also* **Total**.

### **Consumes from**

A relation used to create a specific type of **Numeric rule** that decrementing a total value, such as specifying the quantity of a **Resource** used.

### **count**

The number or quantity of something, such as selected **options**. *Compare* **instance**.

### **CRM**

*See* **Customer Relationship Management**

### **CTO**

Configure to Order

**customer**

The person for whom products are configured by **end users** of the **Oracle Configurator** or other **ERP** and **CRM** applications. Also the end users themselves directly accessing **Oracle Configurator** in a Web store or kiosk.

**customer-centric extensions**

See **customer-centric views**.

**customer-centric views**

Optional extensions to core functionality that supplement configuration activities with **rules** for **preselection**, **validation**, and **intelligent views**. View capabilities include generative geometry, drawings, sketches and schematics, charts, performance analyses, and **ROI** calculations.

**Customer Relationship Management**

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

**customer requirements**

The needs of the customer that serve as the basis for determining the configuration of products, **systems**, and services. Also called needs assessment. See **guided buying or selling**.

**CZ**

The product shortname for **Oracle Configurator** in Oracle Applications.

**data import**

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

**Data Integration Object**

Also known as the DIO, the Data Integration Object is a **server** in the **runtime** application that creates and manages the interface between the **client** (usually a **user interface**) and the **Oracle Configurator schema**.

**data maintenance environment**

The environment in which the runtime **Oracle Configurator** data is maintained.

**data source**

A programmatic reference to a database. Referred to by a data source name (DSN).

**DBMS**

Database Management System

**default**

A predefined value. In a **configuration**, the automatic selection of an **option** based on the **preselection** rules or the selection of another option.

**Defaults rule**

An **Oracle Configurator Developer** Logic rule that determines the logic state of **Features** or **Options** in a default relation to other Features and Options. For example, if A Defaults B, and you select A, B becomes Logic True (selected) if it is available (not Logic False).

**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 table view.

**design review**

A technical review that focuses on application or **system** design.

**developer**

The person who uses **Oracle Configurator Developer** to create a **configurator**. *See also **implementer** and **user**.*

**Developer**

The tool (**Oracle Configurator Developer**) used to create **configuration models**.

**DHTML**

Dynamic Hypertext Markup Language

**DIO**

*See **Data Integration Object**.*

**distributed computing**

Running various **components** of a **system** on separate machines in one network, such as the database on a database **server** machine and the application software on a Web server machine.

**DLL**

Dynamically Linked Library

**DSN**

*See **data source**.*

**element**

Any entity within a **model**, such as **Options**, **Totals**, **Resources**, UI controls, and **components**.

**end user**

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

**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 rule**

An **Oracle Configurator Developer** Logic rule determines the logic state of **Features** or **Options** in an excluding relation to other Features and Options. For example, if A Excludes B, and if you select A, B becomes Logic False, since it is not allowed when A is true (either User or Logic True). If you deselect A (set to User False), there is no effect on B, meaning it could be User or Logic True, User or Logic False, or **Unknown**. See **Negates rule**.

**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**

A characteristic of something, or a configurable element of a **component** at **runtime**.

**Feature**

An element of the **model structure**. Features can either have a value (numeric or Boolean) or enumerated **Options**.

**Functional Companion**

An extension to the **configuration model** beyond what can be implemented in Configurator Developer.

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.

**guided buying or selling**

Needs assessment questions in the **runtime** UI to guide and facilitate the configuration process. Also, the **model structure** that defines these questions. Typically, guided selling questions trigger **configuration rules** that automatically select some product **options** and exclude others based on the **end user's** responses.

**host application**

An application within which **Oracle Configurator** is embedded as integrated functionality, such as Order Management or iStore.

**HTML**

Hypertext Markup Language

**ICX**

Inter-Cartridge Exchange

**implementation**

The stage in a project between defining the problem by selecting a configuration technology vendor, such as Oracle, and deploying the completed configuration application. The implementation stage includes gathering requirements, defining test cases, designing the application, constructing and testing the application, and delivering it to **end users**. *See also* **developer** and **user**.

**implementer**

The person who uses **Oracle Configurator Developer** to build the **model structure**, rules, and UI customizations that make up a **runtime** Oracle Configurator. Commonly also responsible for enabling the integration of **Oracle Configurator** in a **host application**.

**Implies rule**

An **Oracle Configurator Developer** Logic rule that determines the logic state of **Features** or **Options** in an implied relation to other Features and Options. For example, if A Implies B, and you select A, B becomes Logic True. If you deselect A (set to User False), there is no effect on B, meaning it could be User or Logic True, User or Logic False, or **Unknown**. *See* **Requires rule**.

**import server**

A database **instance** that serves as a source of data for **Oracle Configurator's** Populate, Refresh, and Synchronization concurrent processes. The import server is sometimes referred to as the remote server.

**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's Item Master. Import tables also store extractions from Oracle Applications or **legacy data** that create, update, or delete records in the Oracle Configurator schema **Item Master**.

**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**.

**install program**

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

**Instance**

An **Oracle Configurator Developer** attribute of a **component's node** that specifies a minimum and maximum value. *See also* **instance**.

**instance**

A **runtime** occurrence of a **component** in a configuration. *See also* **instantiate**. *Compare* **count**.

Also, the memory and processes of a database.

**instantiate**

To create an instance of something. Commonly, to create an **instance** of a **component** in the runtime **user interface** of a **configuration model**.

**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**. *Compare* **unit test**.

**intelligent views**

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

**IS**

Information Services

**item**

A product or part of a product that is in inventory and can be delivered to customers.

**Item**

A Model or part of a Model that is defined in the [Item Master](#). Also data defined in Oracle Inventory.

**Item Master**

Data stored to structure the Model. Data in the Item Master is either entered manually in [Oracle Configurator Developer](#) or imported from Oracle Applications or a legacy system.

**Item Type**

Data used to classify the Items in the Item Master. Item Catalogs imported from Oracle Inventory are Item Types in [Oracle Configurator Developer](#).

**Java**

An object-oriented programming language commonly used in internet applications, where Java applications run inside Web browsers and [servers](#). *See also* [applet](#) and [servlet](#).

**LAN**

Local Area Network

**LCE**

Logical Configuration Engine. *Compare* [Active Model](#).

**legacy data**

Data that cannot be imported without creating custom extraction programs.

**load**

Storing the **configuration model** data in the **Oracle Configurator Servlet** on the Web server. Also, the time it takes to initialize and display a configuration model if it is not preloaded.

The burden of transactions on a **system**, commonly caused by the ratio of **user** connections to CPUs or available memory.

**log file**

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

**Logic rules**

Logic rules directly or indirectly set the logical state (User or Logic True, User or Logic False, or **Unknown**) of **Features** and **Options** in the Model.

There are four primary Logic rule relations: Implies, Requires, Excludes, and Negates. Each of these rules takes a list of Features or Options as operands. *See also* **Implies rule**, **Requires rule**, **Excludes rule**, and **Negates rule**.

**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 **defect** fixes, procedure changes, infrastructure adjustments, data replication schedules, and so on.

**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. Includes both imported **BOM Models** and Models created in Configurator Developer. May consist of BOM Option Classes and BOM Standard Items.

**model**

A generic term for data representing products. A model contains **elements** that correspond to **items**. Elements may be **components** of other objects used to define

products. A **configuration model** is a specific kind of model whose elements can be configured by accessing an **Oracle Configurator window**.

### **model-driven UI**

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

### **model structure**

Hierarchical "tree" view of data composed of **elements** (**Models**, **Components**, **Features**, **Options**, **BOM Models**, **BOM Option Class nodes**, **BOM Standard Item nodes**, **Resources**, and **Totals**). May include reusable **components** (**References**).

### **MS**

Microsoft Corporation

### **Negates rule**

A type of **Oracle Configurator Developer** Logic rule that determines the logic state of **Features** or **Options** in a negating relation to other Features and Options. For example, if one **option** in the relationship is selected, the other option must be Logic False (not selected). Similarly, if you deselect one option in the relationship, the other option must be Logic True (selected). *See* **Excludes rule**.

### **node**

The icon or location in a **Model** tree in **Oracle Configurator Developer** that represents a **Component**, **Feature**, **Option** or variable (**Total** or **Resource**), **Connector**, **Reference**, **BOM Model**, **BOM Option Class node**, or **BOM Standard Item node**.

### **Numeric rule**

An **Oracle Configurator Developer** rule type that express constraint among model elements in terms of numeric relationships. *See also*, **Contributes to** and **Consumes from**.

### **OC**

*See* **Oracle Configurator**.

### **ODBC**

Open Database Connectivity. A database access method that uses drivers to translate an application's data queries into **DBMS** commands.

## OCD

See [Oracle Configurator Developer](#).

## opportunity

The workspace in Oracle Sales Online in which products, [systems](#), and services are configured, quotes and proposals are generated, and orders are submitted.

## option

A logical selection made by the [end user](#) when configuring a [component](#).

## Option

An element of the [Model](#). A choice for the value of an enumerated [Feature](#).

## Oracle Configuration Interface Object (CIO)

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

The CIO is the [API](#) that supports creating and navigating the Model, querying and modifying selection states, and saving and restoring [configurations](#).

## Oracle Configurator

The product consisting of development tools and [runtime](#) applications such as the [Oracle Configurator schema](#), [Oracle Configurator Developer](#), and runtime Oracle Configurator. Also the runtime Oracle Configurator variously packaged for use in networked or Web deployments.

## Oracle Configurator architecture

The three-tier [runtime](#) architecture consists of the [User Interface](#), the [Active Model](#), and the [Oracle Configurator schema](#). The application development architecture consists of [Oracle Configurator Developer](#) and the Oracle Configurator schema, with test instances of a runtime [Oracle Configurator](#).

## Oracle Configurator Developer

The suite of tools in the [Oracle Configurator](#) product for constructing and maintaining [configurators](#).

## Oracle Configurator engine

Also [LCE](#). Compare [Active Model](#).

### **Oracle Configurator schema**

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

### **Oracle Configurator Servlet**

Vehicle for **Oracle Configurator** containing the UI Server.

### **Oracle Configurator window**

The **user interface** that is launched by accessing a **configuration model** and used by **end users** to make the selections of a **configuration**.

### **Oracle SellingPoint Application**

No longer available or supported.

### **output**

The output generated by a **configurator**, such as quotes, proposals, and **customer-centric views**.

### **performance**

The operation of a product, measured in throughput and other data.

### **Populator**

An entity in **Oracle Configurator Developer** that creates **Component**, **Feature**, and **Option nodes** from information in the **Item Master**.

### **preselection**

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**.

### **product**

Whatever is ordered and delivered to customers, such as the output of having configured something based on a model. Products include intangible entities such as services or contracts.

**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 a **node** in the **Model** or the **Item Master**. A set of Properties may be associated with an Item Type. After importing a BOM Model, Oracle Inventory Catalog Descriptive Elements are Properties in **Oracle Configurator Developer**.

**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, prove feasibility, or examine other implementation issues.

**PTO**

Pick to Order

**publication**

A unique deployment of a **configuration model** (and optionally a **user interface**) that enables a developer to control its availability from hosting applications such as Oracle Order Management or iStore. Multiple publications can exist for the same configuration model, but each publication corresponds to only one **Model** and **User Interface**.

**publishing**

The process of creating a **publication** record in **Oracle Configurator Developer**, which includes specifying applicability parameters to control **runtime** availability and running an Oracle Applications concurrent process to copy data to a specific database.

**QA**

Quality Assurance

**RAD**

Rapid Application Development

**RDBMS**

Relational Database Management System

**reference**

The ability to reuse an existing **Model** or **Component** within the structure of another Model (for example, as a subassembly).

**Reference**

An **Oracle Configurator Developer** node type that denotes a **reference** to another **Model**.

**regression test**

An automated test that ensures the newest **build** still meets previously tested requirements and functionality. *See also* **incremental construction**.

**Requires rule**

An **Oracle Configurator Developer** Logic rule that determines the logic state of **Features** or **Options** in a requirement relation to other Features and Options. For example, if A Requires B, and if you select A, B is set to Logic True (selected). Similarly, if you deselect A, B is set to Logic False (deselected). *See* **Implies rule**.

**resource**

Staff or equipment available or needed within an enterprise.

**Resource**

A variable in the **Model** used to keep track of a quantity or supply, such as the amount of memory in a computer. The value of a Resource can be positive or zero, and can have an Initial Value setting. An error message appears at **runtime** when the value of a Resource becomes negative, which indicates it has been over-consumed. Use **Numeric rules** to contribute to and consume from a Resource.

Also a specific node type in **Oracle Configurator Developer**. *See also* **node**.

**reusable component**

See [reference](#) and [model structure](#).

**reusability**

The extent to and ease with which parts of a [system](#) can be put to use in other systems.

**RFQ**

Request for Quote

**ROI**

Return on Investment

**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 provide [preselection](#) and [validation](#) capability in [Oracle Configurator](#).

See also [Comparison rule](#), [Compatibility rule](#), [Design Chart](#), [Logic rules](#) and [Numeric rule](#).

**runtime**

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

The environment in which an [implementer](#) (tester), [end user](#), or [customer](#) configures a product whose model was developed in [Oracle Configurator Developer](#). See also [configuration session](#).

**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 [customer requirements](#) and product configuration.

**schema**

The tables and objects of a data model that serve a particular product or business process. See [Oracle Configurator schema](#).

**SCM**

Supply Chain Management

**server**

Centrally located software processes or hardware, shared by **clients**.

**servlet**

A Java application running inside a Web server. *See also* **Java**, **applet**, and **Oracle Configurator Servlet**.

**SFA**

Sales Force Automation

**solution**

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

**SQA**

Software Quality Assurance

**SQL**

Structured Query Language

**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 that are created to determine 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.

Also a specific node type in **Oracle Configurator Developer**. *See also* **node**.

**UI**

*See* **User Interface**.

### **Unknown**

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

### **unit test**

Execution of individual routines and modules by the application **implementer** or by an independent test consultant to find and resolve **defects** in the application. Compare **integration testing**.

### **update**

Moving to a new version of something, independent of software release. For instance, moving a production **configurator** to a new version of a **configuration model**, or changing a **configuration** independent of a model **update**.

### **upgrade**

Moving to a new release of **Oracle Configurator** or **Oracle Configurator Developer**.

### **user**

The person using a product or system. Used to describe the person using **Oracle Configurator Developer** tools and methods to build a **runtime Oracle Configurator**. Compare **end user**.

### **User Interface**

The part of **Oracle Configurator architecture runtime** architecture that is generated from the **model structure** and provides the graphical views necessary to create **configurations** interactively. Interacts with the **Active Model** and data to give **end users** access to customer requirements gathering, product selection, and **customer-centric views**.

### **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. Not necessarily generated in **Oracle Configurator Developer**.

### **user requirements**

A description of what the **configurator** 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 configured **components** will meet specific criteria set by an enterprise, such as that the components can be ordered or manufactured.

**Validation**

A type of **Functional Companion** that is implemented to ensure that the configured **components** will meet specific criteria.

**VAR**

Value-Added Reseller

**variable**

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

**VB**

Microsoft Visual Basic. Programming language in which portions of **Oracle Configurator Developer** are written.

**verification**

Tests that check whether the result agrees with the specification.

**WAN**

Wide Area Network

**Web**

The portion of the Internet that is the World Wide Web.

**WIP**

Work In Progress



## Symbols

---

SAPPL\_TOP, 2-3

## A

---

adpatch, 1-2, 1-10

Alias, 2-4

aliases, 2-4

AltBatchValidateURL, 2-15

Apache, 2-1

configuration files

ApJServManual, 3-8

ApJServMount, 3-8

ApJServVMTimeout, 3-8

httpd.conf, 2-4

jserv.conf, 2-4

jserv.properties, 2-5

zone.properties, 2-8

installation location, 2-3

parameters

Alias, 2-4

DocumentRoot, 2-4

Port, 2-4

repositories, 2-8

ServerRoot, 2-4

troubleshooting installation problems, 3-5

*apache\_install*

installation location, 2-3

ApacheJServ.jar, 2-7, 2-12

ApJServManual, 2-9, 3-8

ApJServMount, 3-8

ApJServVMTimeout, 3-8

apps.zip, 2-6, 2-12

ASO:Configurator URL

profile option, 1-6

ASO:Enable Configure Model Item

profile option, 1-7

## B

---

batch validation

AltBatchValidateURL, 2-15

BOM:Configurator URL of UI Manager

profile option, 1-7

## C

---

CLASSPATH, 2-10

classpath, 2-6

client environment

requirements for Oracle Configurator

Developer, 4-7

client/server environment

diagram, 4-5

requirements for Oracle Configurator

window, 4-8

config.jar, 2-12

configuration file

modifying, 1-4

CZ:Automatically Validate on Exit

profile option, 1-7

CZ:Publication Lookup Mode

profile option, 1-8

CZ:Publication Usage

profile option, 1-9

CZ:Use Simple Configurator

profile option, 1-9

- cz3rdpty.jar, 2-12
- cz.activemodel, 2-21
- cz.activemodel.lazyloadlistprice, 2-23
- cz.html.source.display, 2-24
- cz.html.source.formtreeview, 2-24
- cz.html.source.treeview, 2-24
- cz.scrolling.treeview, 2-25
- cz.uimanager.logpath, 2-25
- cz.uiserver.check\_heart\_beat\_timeout, 2-26
- cz.uiserver.ciolog, 2-27
- cz.uiserver.displayconsequences, 2-27
- cz.uiserver.heartbeat\_interval, 2-28
- cz.uiserver.lazyload, 2-28
- cz.uiserver.lfalse\_is\_not\_available, 2-29
- cz.uiserver.media.folder, 2-29
- cz.uiserver.poll\_timeout\_applet, 2-30
- cz.uiserver.poll\_timeout\_applet\_to\_dhtml, 2-30
- cz.uiserver.summary.itemcolumn, 2-30
- cz.uiservlet.applet.summary, 2-31
- cz.uiservlet.blaftemplateurl, 2-31
- cz.uiservlet.dio\_share, 2-31
- cz.uiservlet.formtemplateurl, 2-32
- cz.uiservlet.jdbcdriver, 2-33
- cz.uiservlet.logfilename, 2-33
- cz.uiservlet.name, 2-34
- cz.uiservlet.pre\_load\_filename, 2-34
- cz.uiservlet.proxyscript, 2-35
- cz.uiservlet.templateurl, 2-36
- cz.uiservlet.url, 2-36
- cz.uiservlet.versionfuncsavail, 2-36

## D

---

### database

- see also Oracle Configurator schema
- connectivity, 4-9
- modifying the database configuration file, 1-4
- setting the DB\_HOST
  - setting the DB\_PORT
  - setting the DB\_NAME, 1-4

### DB\_HOST

- specifying the database hostname, 1-4

### DB\_NAME, 1-4

### DB\_PORT, 1-4

### DBOwner

- creation, 4-10
- in spx.ini DSN section, 4-13
- in spx.ini Merlin section, 4-13
- parameter for development and testing, 4-8

### Developer

- installation, 4-7
- login, 4-17
- running, 4-17

### DocumentRoot, 2-4

### DSN

- creation, 4-10
- section in spx.ini file, 4-13

### dsn

- section in spx.ini file, 4-13, 4-14, 4-15

## E

---

### end user

- responsibilities, 4-6

## F

---

### Functional Companions, 2-7

## G

---

### Generate Active Model

- before Test, 4-19

### Generate Active UI

- before Test, 4-19

- green threads, 2-6

- Xss8m option, 2-6

### guided selling

- support for interface, 2-26

### gwyypass

- in spx.ini DSN section, 4-14

### gwyuid

- in spx.ini DSN section, 4-14

## H

---

- heap size, 2-6

- setting maximum heap size, 2-9

### heartbeat

- mechanism for interface management, 2-26

- Hello test class, 3-5

- host
  - determining host information, 3-2
  - test message, 3-2
- HTML template
  - Netscape, 2-36
- html\_vpath*
  - internet server parameter, 2-3
- HTTPD, 2-9
- httpd.conf, 2-4

## I

---

- iAS, 2-1
- ias\_install*
  - installation location, 2-2
- initArgs file, 2-19
- InitCodeBaseUrl
  - in spx.ini Test section, 4-14
- InitServletUrl
  - parameter in the spx.ini file, 4-14
- installing
  - Oracle Configurator Developer, 4-2
- internet
  - configure server, 2-2
  - server parameters, 2-2

## J

---

- Java
  - green threads, 2-6
  - interpreter, 2-8, 2-18
  - JDK
    - version, 2-2, 2-5
    - production, 2-5
  - native threads, 2-6
  - system properties, 2-8
  - Xss8m option, 2-6
- Java applet
  - client and server requirements, 4-8
- Java Virtual Machine, 2-5
- JDBC driver
  - modifying the database configuration file, 1-4
  - property for setting, 2-33
- jdbc111.zip, 2-6, 2-12
- JdbcUrl

- in spx.ini DSN section, 4-14
- JDK, 2-2, 2-5
- jsdk.jar, 2-7, 2-12
- JServ, 2-1, 2-9
  - installation location, 2-3
- jserv\_install*
  - installation location, 2-3
- jserv.conf, 2-4
- jserv.log, 2-7
- jserv.properties, 2-5
  - editing for Secure Sockets Layer, 2-14
- JVM (Java Virtual Machine), 2-10
  - maximum heap size, 2-9

## L

---

- Language
  - Multiple Language Support, 1-15
- Launch
  - in spx.ini Test section, 4-14
- LD\_LIBRARY\_PATH, 2-7
- listening port
  - for JServ, 2-5
  - for web server, 2-4
- load balancing, 2-9
  - procedure, 2-9
  - script, 2-13
  - vendor information, 2-8
- logs
  - configuration session log, 2-33
  - exception log, 2-25
  - servlet log, 2-33

## M

---

- maximum heap size
  - setting, 2-9
- media\_vpath*
  - internet server parameter, 2-3
- memory requirements, 2-6
- Merlin
  - section in spx.ini file, 4-13
- migrating
  - from standalone Oracle SellingPoint schema, 1-11

- prerequisites, 1-11
- running migration scripts, 1-12

## MLS

- see Multiple Language Support

- model caching, 2-35

- mount point

- for servlet zone, 2-5

- Multiple Language Support

- relevant servlet properties

- setup considerations, 1-15

## N

---

- native threads, 2-6

- required for guided selling, 2-27

## O

---

- OA\_HTML, 2-3, 2-4

- OA\_MEDIA, 2-3, 2-4

- OC Servlet

- allow test message property, 2-36

- installation, 2-1

- properties

- cz.activemodel, 2-21

- cz.activemodel.lazyloadlistprice, 2-23

- cz.html.source.display, 2-24

- cz.html.source.formtreeview, 2-24

- cz.html.source.treeview, 2-24

- cz.scrolling.treeview, 2-25

- cz.uimanager.logpath, 2-25

- cz.uiserver.check\_heart\_beat\_timeout, 2-26

- cz.uiserver.ciolog, 2-27

- cz.uiserver.displayconsequences, 2-27

- cz.uiserver.heartbeat\_interval, 2-28

- cz.uiserver.lazyload, 2-28

- cz.uiserver.lfalse\_is\_not\_available, 2-29

- cz.uiserver.media.folder, 2-29

- cz.uiserver.poll\_timeout\_applet, 2-30

- cz.uiserver.poll\_timeout\_applet\_to\_dhtml, 2-30

- cz.uiserver.summary.itemcolumn, 2-30

- cz.uiservlet.applet.summary, 2-31

- cz.uiservlet.blaftemplateurl, 2-31

- cz.uiservlet.dio\_share, 2-31

- cz.uiservlet.formtemplateurl, 2-32

- cz.uiservlet.jdbcdriver, 2-33

- cz.uiservlet.logfilename, 2-33

- cz.uiservlet.name, 2-34

- cz.uiservlet.pre\_load\_filename, 2-34

- cz.uiservlet.proxyscript, 2-35

- cz.uiservlet.templateurl, 2-36

- cz.uiservlet.url, 2-36

- cz.uiservlet.versionfuncsavail, 2-36

- specifying URL, 2-36

- ODBC, see DSN

- OM:Use Configurator

- profile option, 1-9

- operating systems

- system requirements, 1-2

- Oracle Configurator

- servlet

- see OC Servlet

- Oracle Configurator Developer

- installing, 4-2

- setup, 4-8

- Oracle Configurator schema

- graphic, 4-5

- migrating from a standalone schema, 1-11

- migrating from Release 11i, 1-12

- overview, 1-1

- version expected by UI Servlet, 3-2

- Oracle Internet Application Server, 2-1

- installation location, 2-2

- Oracle Net8 Easy Config

- for database connectivity, 4-9

- Oracle Rapid Install, 1-1, 1-3, 1-12, 2-1, 2-8

- Rapid Install

- see Oracle Rapid Install

- Oracle8 Client

- installation, 4-7

- requirement, 4-5

- Oracle8i Enterprise Edition

- requirement, 4-5

## P

---

- parameters

- Apache Web server parameters, 2-17

- configuring Apache and JServ, 2-2

- in httpd.conf, 2-4
- in jserv.conf, 2-4
- in jserv.properties, 2-5
- in the spx.ini file, 4-12
- in zone.properties, 2-8
- modify database configuration file
  - parameters, 1-4
- see also OC Servlet
- patches
  - adpatch, 1-2
  - for Oracle Configurator, 1-2
  - for Oracle Configurator Developer, 1-2, 4-3
  - tcpatch, 4-3
- performance
  - caching model, 2-32, 2-35
  - impact of pricing, 2-23
  - preloading servlet, 2-35
- platforms
  - supported operating systems and system requirements, 1-2
- Port, 2-4
- pricing
  - performance impact, 2-23
- Product Support, xvi
- production version of JDK, 2-5
- profile options
  - ASO:Configurator URL, 1-6
  - ASO:Enable Configure Model Item, 1-7
  - BOM:Configurator URL of UI Manager, 1-7
  - CZ:Automatically Validate on Exit, 1-7
  - CZ:Publication Lookup Mode, 1-8
  - CZ:Publication Usage, 1-9
  - CZ:Use Simple Configurator, 1-9
  - for Oracle Configurator, 1-5
  - OM:Use Configurator, 1-9
- Proxy Frame, 2-35

## R

---

- relative URL
  - defined, 2-19
  - for cz.uiservlet.blaftemplateurl, 2-31
  - for cz.uiservlet.formtemplateurl, 2-32
  - for cz.uiservlet.proxyscript, 2-35
  - for cz.uiservlet.templateurl, 2-36

- repositories, 2-8
- response
  - from UI Servlet, 3-1
- return\_url servlets, 2-7
- runtime Oracle Configurator
  - client requirements, 4-8
  - login, 4-17
  - running, 4-17
  - spx.ini file requirements, 4-12

## S

---

- secure
  - implementing Secure Sockets Layer, 2-13
- Secure Sockets Layer
  - editing jserv.properties, 2-14
  - setting up Oracle Configurator, 2-13
  - verifying AltBatchValidateURL, 2-15
- security risk, 2-11
- server
  - kinds of, 4-4
- server environment
  - requirements for Oracle Configurator Developer, 4-4
- ServerRoot, 2-4
- servlet
  - see OC Servlet
- servlet\_vpath*
  - internet server parameter, 2-3
- servlets.default.initArgs, 2-17
  - recommended for setting parameters, 2-18
- SHLIBPATH, 2-7
- spx.ini file
  - example, 4-12
  - parameters for development and testing, 4-12
- SSL
  - see Secure Sockets Layer
- Support, xvi

## T

---

- Terminal Server
  - installing Configurator Developer, 4-6
- Test
  - button in Developer, 4-14, 4-19

- test class, 3-5
- test message, 3-1
  - control for OC Servlet, 2-36
- test page, 3-3
- testing
  - Oracle Configurator Developer installation, 4-17
- thin client usage
  - see JdbcUrl, 4-14
- timeout
  - parameter in httpd.conf, 2-4
- Translations
  - Multiple Language Support, 1-15

## U

---

- upgrading
  - from previous 11i release, 1-10
- URL
  - see also InitServletURL, 4-14
  - for OC Servlet, 2-36
  - Oracle Configurator window requirement, 4-8, 4-14
  - see also relative URL
- user
  - end user creation, 4-6

## V

---

- validation
  - AltBatchValidateURL, 2-15
- version
  - determining build version, 3-2
  - property that accepts or rejects test message, 2-36
  - test message, 3-2

## W

---

- Windows
  - supported operating systems, 1-2
- wrapper.bin.parameters, 2-18
- wrapper.classpath, 2-6
- wrapper.env, 2-7

## X

---

- xmlparserv2.zip, 2-6, 2-12

## Z

---

- zone.properties, 2-8