Oracle® Product Configurator User's Guide

Release 11

March 1998

ORACLE®

Enabling the Information Age™

Oracle® Product Configurator User's Guide
Release 11

The part number for this volume is A57334–01.

Major Contributors: Bryan Dobson

Contributors: Kurt Thompson, Sharon Lee, Rajan Moses, Dana Lieberman

Contents

Preface

Welcome to the *Oracle® Product Configurator User's Guide, Release 11*.

This user's guide includes the information you need to work with Oracle Product Configurator effectively. It contains detailed information about the following:

- Overview and reference information
- Specific tasks you can accomplish using Oracle Product Configurator
- Oracle Product Configurator setup
- Oracle Product Configurator functions and features
- Oracle Product Configurator windows
- Oracle Product Configurator reports and processes

This preface explains how this user's guide is organized and introduces other sources of information that can help you.

About This User's Guide

This guide contains overviews as well as task and reference information about Oracle Product Configurator. This guide includes the following chapters:

- Chapter 1 provides an overview of the features of Product Configurator
- Chapter 2 provides setup information and describes how to define and assign Configuration Constraints and Attributes.

  Note: Implementation information and procedures are contained in this chapter.
- Chapter 3 describes how to use configurator systems and match and reserve functionality to configure products using Product Configurator.
- Chapter 4 describes Product Configurator reports.
- The appendix provides you with complete navigation paths to all windows in Product Configurator.

Audience for This Guide

This guide assumes you have a working knowledge of your business area's processes and tools. It also assumes you are familiar with Product Configurator. If you have never used Product Configurator, we suggest you attend one or more of the Product Configurator training classes available through World Wide Education. For more information about Product Configurator and Oracle training, see: Other Information Sources.

Do Not Use Database Tools to Modify Oracle Applications Data

Because Oracle Applications tables are interrelated, any change you make using Oracle Applications can update many tables at once. But when you modify Oracle Applications data using anything other than Oracle Applications, you may change a row in one table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle Applications.

When you use Oracle Applications to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications also keeps track of who changes information. If you enter information into database tables using database tools, you may store invalid information. You also lose the ability to track who has changed your information because SQL*Plus and other database tools do not keep a record of changes.

Consequently, we STRONGLY RECOMMEND that you never use SQL*Plus or any other tool to modify Oracle Applications data unless otherwise instructed.

Other Information Sources

Here are some other ways you can increase your knowledge and understanding of Product Configurator.

Online Documentation

All Oracle Applications documentation is available online on CD–ROM, except for technical reference manuals. There are two online

formats, HyperText Markup Language (HTML) and Adobe Acrobat (PDF).

All user's guides are available in HTML, Acrobat, and paper. Technical reference manuals are available in paper only. Other documentation is available in Acrobat and paper.

The *content* of the documentation does not differ from format to format. There may be slight differences due to publication standards, but such differences do not affect content. For example, page numbers and screen shots are not included in HTML.

The HTML documentation is available from all Oracle Applications windows. Each window is programmed to start your web browser and open a specific, context–sensitive section. Once any section of the HTML documentation is open, you can navigate freely throughout all Oracle Applications documentation. The HTML documentation also ships with Oracle Information Navigator (if your national language supports this tool), which enables you to search for words and phrases throughout the documentation set.

Related User's Guides

Product Configurator shares business and setup information with other Oracle Applications products. Therefore, you may want to refer to other user's guides when you set up and use Product Configurator.

If you do not have the hardcopy versions of these manuals, you can read them online using the Applications Library icon or Help menu command.

Oracle Applications User's Guide

This guide explains how to enter data, query, run reports, and navigate using the graphical user interface (GUI) available with this release of Product Configurator (and any other Oracle Applications products). This guide also includes information on setting user profiles, as well as running and reviewing reports and concurrent processes.

You can access this user's guide online by choosing "Getting Started with Oracle Applications" from any Oracle Applications help file.

Oracle Applications Demonstration User's Guide

This guide documents the functional storyline and product flows for Global Computers, a fictional manufacturer of personal computers products and services. As well as including product overviews, the

book contains detailed discussions and examples across each of the major product flows. Tables, illustrations, and charts summarize key flows and data elements.

Oracle Bills of Material User's Guide

This guide describes how to create various bills of materials to maximize efficiency, improve quality and lower cost for the most sophisticated manufacturing environments. By detailing integrated product structures and processes, flexible product and process definition, and configuration management, this guide enables you to manage product details within and across multiple manufacturing sites.

Oracle Engineering User's Guide

This guide enables your engineers to utilize the features of Oracle Engineering to quickly introduce and manage new designs into production. Specifically, this guide details how to quickly and accurately define the resources, materials and processes necessary to implement changes in product design.

Oracle Inventory User's Guide

This guide describes how to define items and item information, perform receiving and inventory transactions, maintain cost control, plan items, perform cycle counting and physical inventories, and set up Oracle Inventory.

Oracle Order Entry/Shipping User's Guide

This guide describes how to enter sales orders and returns, copy existing sales orders, schedule orders, release orders, plan departures and deliveries, confirm shipments, create price lists and discounts for orders, and create reports.

Oracle Service User's Guide

This guide describes how you can track service requests, maintain and repair customer products in your installed base, and bill your customers for services rendered. This guide also gives an overview of the workflows that Oracle Service provides.

Oracle Work in Process User's Guide

This guide describes how Oracle Work in Process provides a complete production management system. Specifically this guide describes how discrete, repetitive, assemble–to–order, project, flow, and mixed manufacturing environments are supported.

Reference Manuals

Oracle Automotive Implementation Manual

This manual describes the setup and implementation of the Oracle Applications used for the Oracle Automotive solution.

Oracle Manufacturing, Distribution, Sales and Service Open Interfaces Manual

This manual contains up–to–date information about integrating with other Oracle Manufacturing applications and with your other systems. This documentation includes open interfaces found in Oracle Manufacturing.

Oracle Applications Message Reference Manual

This manual describes all Oracle Applications messages. This manual is available in HTML format on the documentation CD–ROM for Release 11.

Oracle Project Manufacturing Implementation Manual

This manual describes the setup steps and implementation for Oracle Project Manufacturing.

Oracle Self–Service Web Applications Implementation Manual

This manual describes the setup steps for Oracle Self–Service Web Applications and the Web Applications dictionary.

Installation and System Administration

Oracle Alert User's Guide

This guide explains how to define periodic and event alerts to monitor the status of your Oracle Applications data.

Multiple Reporting Currencies in Oracle Applications

If you use the Multiple Reporting Currencies feature to record transactions in more than one currency, use this manual before implementing Product Configurator. This manual details additional steps and setup considerations for implementing Product Configurator with this feature.

Multiple Organizations in Oracle Applications

If you use the Oracle Applications Multiple Organization Support feature to use multiple sets of books for one Product Configurator installation, this guide describes all you need to know about setting up and using Product Configurator with this feature.

Oracle Applications Implementation Wizard User's Guide

If you are implementing more than one Oracle product, you can use the Oracle Applications Implementation Wizard to coordinate your setup activities. This guide describes how to use the wizard.

Oracle Applications Developer's Guide

This guide contains the coding standards followed by the Oracle Applications development staff. It describes the Oracle Application Object Library components needed to implement the Oracle Applications user interface described in the *Oracle Applications User Interface Standards*. It also provides information to help you build your custom Developer/2000 forms so that they integrate with Oracle Applications.

Oracle Applications Flexfields Guide

This guide provides flexfields planning, setup and reference information for the Product Configurator implementation team, as well as for users responsible for the ongoing maintenance of Oracle Applications product data. This manual also provides information on creating custom reports on flexfields data.

Oracle Applications Installation Manual for Windows Clients

This guide provides information you need to successfully install Oracle Financials, Oracle Public Sector Financials, Oracle Manufacturing, or Oracle Human Resources in your specific hardware and operating system software environment.

Oracle Applications Product Update Notes

If you are upgrading your Oracle Applications, refer to the product update notes appropriate to your update and product(s) to see summaries of new features as well as changes to database objects, profile options and seed data added for each new release.

Oracle Applications Upgrade Preparation Manual

This guide explains how to prepare your Oracle Applications products for an upgrade. It also contains information on completing the upgrade procedure for each product. Refer to this manual and the *Oracle Applications Installation Manual* when you plan to upgrade your products.

Oracle Applications System Administrator's Guide

This manual provides planning and reference information for the Product Configurator System Administrator.

Other Sources

Training

We offer a complete set of formal training courses to help you and your staff master Product Configurator and reach full productivity quickly. We organize these courses into functional learning paths, so you take only those courses appropriate to your job or area of responsibility.

You have a choice of educational environments. You can attend courses offered by Oracle Education Services at any one of our many Education Centers, or you can arrange for our trainers to teach at your facility. In addition, Oracle training professionals can tailor standard courses or develop custom courses to meet your needs. For example, you may want to use your organization structure, terminology, and data as examples in a customized training session delivered at your own facility.

Support

From on–site support to central support, our team of experienced professionals provides the help and information you need to keep Product Configurator working for you. This team includes your Technical Representative, Account Manager, and Oracle's large staff of consultants and support specialists with expertise in your business

area, managing an Oracle8 server, and your hardware and software environment.

About Oracle

Oracle Corporation develops and markets an integrated line of software products for database management, applications development, decision support, and office automation, as well as Oracle Applications, an integrated suite of more than 45 software modules for financial management, supply chain management, manufacturing, project systems, human resources and sales and service management.

Oracle products are available for mainframes, minicomputers, personal computers, network computers and personal digital assistants, allowing organizations to integrate different computers, different operating systems, different networks, and even different database management systems, into a single, unified computing and information resource.

Oracle is the world's leading supplier of software for information management, and the world's second largest software company. Oracle offers its database, tools, and applications products, along with related consulting, education, and support services, in over 140 countries around the world.

Thank You

Thank you for using Product Configurator and this user's guide.

We value your comments and feedback. At the end of this guide is a Reader's Comment Form you can use to explain what you like or dislike about Product Configurator or this user's guide. Mail your comments to the following address or call us directly at (650) 506–7000.

Oracle Applications Documentation Manager
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Or, send electronic mail to **appsdoc@us.oracle.com**.

CHAPTER

1

Overview of Oracle Product Configurator

This chapter provides you with an overview of Product Configurator and describes the benefits and features of the application.

Overview of Oracle Product Configurator

The Oracle Product Configurator:

- verifies product configurations
- automatically selects configuration options
- generates manufacturing bills of material from configurations

Using the Configurator eliminates configuration errors in Order Entry, reduces change order processing, and reduces rework in manufacturing.

The Configurator supports:

- Assemble–to–Order and Pick–to–Order models
- user–defined item attributes
- flexible configuration constraints
- integrated configuration validation
- automatic configuration completion
- automatic generation of valid manufacturing bills
- Enterprise Resource Planning (ERP) integration

ERP Integration

Oracle Product Configurator integrates with ERP using the following Oracle Manufacturing and Distribution products:

- Order Entry
- Bills of Material
- Inventory
- MRP
- Shipping
- Service
- Work in Process

See Also

Overview of Configuration Constraints: page: 2 – 11

Overview of Configuration Attributes: page: 2 – 4

Configuring a Product: page 3 – 2

CHAPTER

2 Setting Up

This chapter provides you with an overview of Configuration Attributes and Contraints. It also describes how to define and assign the attributes and constraints. Task descriptions provide you with detailed information about the following:

Setup Overview

Oracle Applications Implementation Wizard

If you are implementing more than one Oracle Applications product, you may want to use the Oracle Applications Implementation Wizard to coordinate your setup activities. The Implementation Wizard guides you through the setup steps for the applications you have installed, suggesting a logical sequence that satisfies cross–product implementation dependencies, and reduces redundant setup steps. The Wizard also identifies steps that can be completed independently, by several teams working in parallel, to help you manage your implementation process most efficiently.

You can use the Implementation Wizard as a resource center to see a graphical overview of setup steps, read online help for a setup activity, and open the appropriate setup window. You can also document your implementation, for further reference and review, by using the Wizard to record comments for each step. See: *Oracle Applications Implementation Wizard User's Guide* and *Oracle Applications System Administrator's Guide.*

Set Up Oracle Applications Technology

The setup steps in this chapter tell you how to implement the parts of Oracle Applications specific to Oracle [Product Name].

The Implementation Wizard guides you through the entire Oracle Applications setup, including system administration. However, if you do not use the Wizard, you need to complete several other setup steps, including:

- performing systemwide setup tasks such as configuring concurrent managers and printers
- managing data security, which includes setting up responsibilities to allow access to a specific set of business data and complete a specific set of transactions, and assigning individual users to one or more of these responsibilities

Also, if your product uses Oracle Workflow to, for example, manage the approval of business documents or to derive Accounting Flexfield values via the Account Generator, you need to set up Oracle Workflow. See: *Oracle Workflow Guide.*

See Also

Oracle Applications Implementation Wizard User's Guide

Oracle Applications System Administrator's Guide

Oracle Workflow Guide

Overview of Configuration Constraints: page 2 – 11

Overview of Configuration Attributes: page 2 – 4

Configurator Systems: page 3 – 6

Overview of Configuration Attributes

Configuration attributes are assigned to items in your product structure. When you define constraints, you reference these attributes and their values. When you configure products, attributes are evaluated by constraints to ensure valid configurations.

You can define an unlimited number of configuration attributes. This enables you to create the names and types of attributes necessary for your products.

You can optionally create regions to ease the maintenance of attribute values.

See Also

Overview of Configuration Constraints: page 2 – 11

Defining Attributes: page 2 – 6

Defining Attribute Value Lists: page 2 – 5

Defining Regions: page 2 – 8

Defining Region Items: page 2 – 9

Associating Attributes with Regions: page 2 – 9

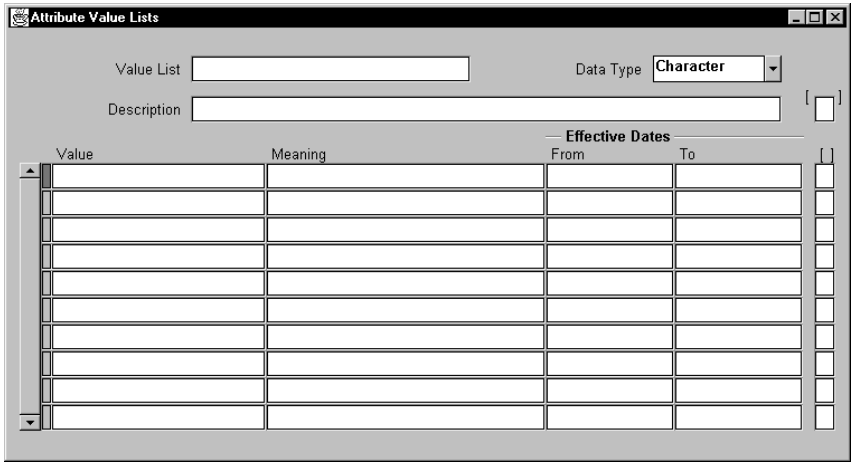
Assigning Attribute Values: page 2 – 10

Defining Attribute Value Lists

Use this window to define attribute value lists for attributes that need more than one value.

▶ **To define attribute value lists:**

1. Navigate to the Attribute Value Lists window.



2. Enter a value list name.
3. Select a numeric or character datatype for the values list.
4. Enter a description for the value list.
5. For each value you are defining:
   - Enter a meaning of the value.
   - Enter an effective From date and, optionally, a To date when the value is disabled.

See Also

Defining Attributes: page 2 – 6

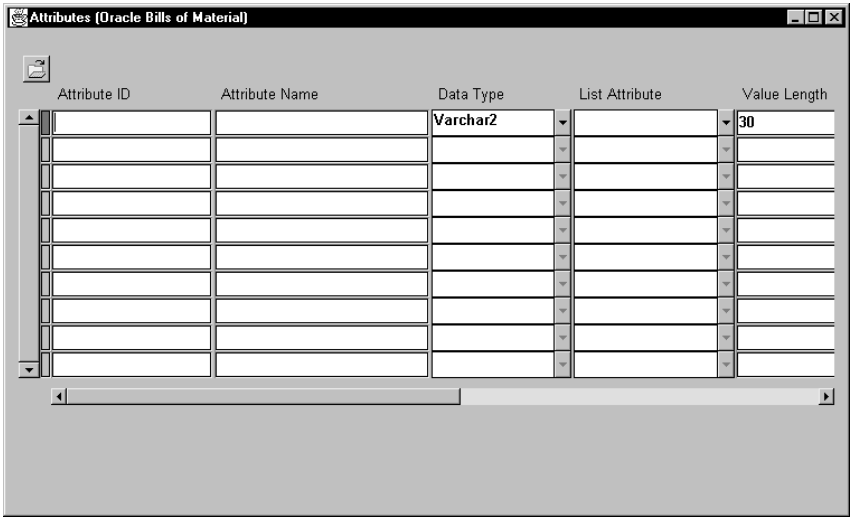
Defining Attributes

The primary use of attributes is to describe properties of items. These attributes and their values are referenced by constraints to ensure valid product configurations.

Another use of attributes is as a container for a list of attribute values. When defining an attribute for this purpose, you must define the list attribute and an attribute value list.

▶ **To define attributes:**

1. Navigate to the Attributes folder window.



2. Enter an attribute ID.
3. Enter a user–friendly attribute name. This is used when defining constraints.
4. Select a datatype, either VarChar2 (long alphanumeric) or numeric.

   If you are defining a list attribute, the datatype must be VarChar2.
5. If you are defining a list attribute, indicate whether attribute values reference character datatype or a numeric datatype.

   Note: Providing a value here means the attribute you are defining will employ a value list.
6. Optionally, modify the length of value or accept the default.

7. Optionally, click the Uppercase button to activate it. This feature is not used by Product Configurator.
8. Optionally, modify the long label (for character attributes only). The default is the attribute name. This feature is not used by Product Configurator.
9. Optionally, enter a label length or accept the default. (This feature is not used by Product Configurator).
10. Optionally, enter a default character value (for character attributes only).
11. Optionally, enter a default number value (for numeric attributes only).
12. Save your work.
13. Save to add the new attribute(s):
    - as object attributes to the database view, MTL_SYSTEM_ITEMS_KFV
    - as region items in the Configurator Attributes Region, CZ_REGION (the primary alternative region) in the Item Attribute Values window.

See Also

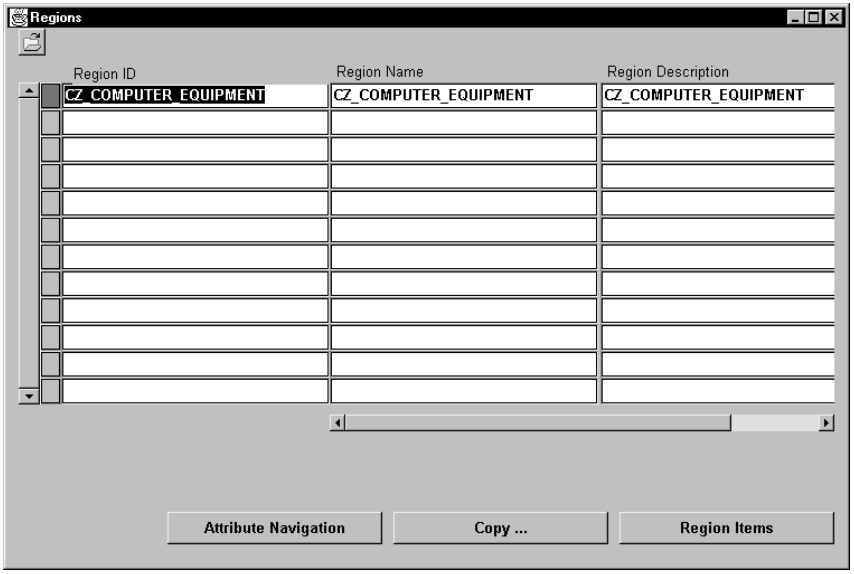
Defining Attribute Value Lists: page 2 – 5

Defining Regions

Regions, or “alternative regions,” are used to group lists of attributes on the Item Attribute Values window. To easily manage attribute values, you can create regions to group similar configuration attributes.

▶ **To define a region:**

1. Navigate to the Attribute Regions window.



2. Enter a region ID.
3. Enter a region name.
4. Optionally, enter a region description. This is a reference field that describes the group of attributes this region contains.

See Also

Associating Attributes with Regions: page 2 – 9

Defining Region Items

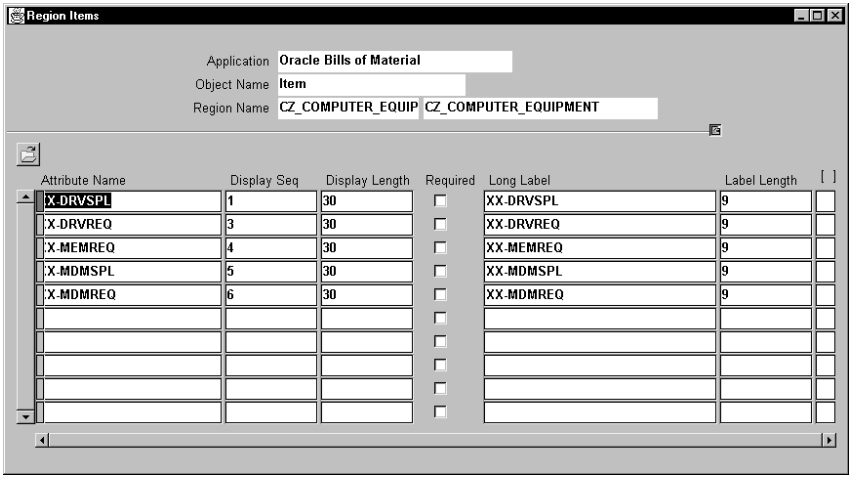
Region items represent an association between a configurator attribute and a region. When attributes are defined they are automatically added to the default region.

Prerequisites

- ❑ Define attributes to associate with regions.
- ❑ Define regions.

▶ **To associate attributes with regions:**

1. Navigate to the Regions window and choose the Region Items button. The Region Items folder window appears.



2. Select an attribute name to associate to a region.
3. Specify a display sequence for the region item. This field is not used by Product Configurator.
4. Optionally, modify the required flag, display length, long label, and label length or accept the defaults. These fields are not used by Product Configurator.

   Note: You can assign an attribute to any number of regions. However, an item attribute value is unique. When you modify an item attribute value in any region, the modified item attribute value appears in all region item attribute value records.

See Also

Defining Attributes: page 2 – 6

Defining Regions: page 2 – 8

Assigning Attribute Values

Item attributes can have one or more values, but only one value per item name. Use the Item Attribute Values window to define and update attribute values. You can query regions in the Item Attribute Values window. During validation, attribute values are taken from the OE: Validation Organization.

▶ **To define attribute values in the Item Attribute Values window:**

1. Navigate to the Item Attribute Values window.
2. Enter or select a region in the Find Attributes window.

   Note: When attributes are created they are automatically assigned to the default region, Configurator Attributes Region (CZ_REGION).
3. Enter or select an item.
4. Select the attribute to assign values to.

   Note: Only attributes assigned to the region in which you reside are displayed in the LOV.
5. Enter values to assign to the attributes.

See Also

Defining Items, *Oracle Inventory User's Guide*

Defining Attributes: page 2 – 6

Overview of Configuration Constraints

Product constraints enforce business rules and guide the configuration process. You can define constraints that operate on and evaluate the product structure and configuration attributes for product options.

You can report on current model constraints and the effects of changes to those constraints.

You can define any number of constraints for a product. Each constraint can have any number of constraint clauses.

See Also

Constraint Clauses: page 2 – 11

Assigning Constraints: page 2 – 23

Constraint Scope: page 2 – 15

Constraint Types: page 2 – 15

Constraint Sequencing: page 2 – 17

Overview of Configuration Attributes: page 2 – 4

Defining Constraints: page 2 – 20

Constraint Clauses

Constraints are made up of one or more clauses. Constraint clauses reference and evaluate attributes for configuration options.

Constraints are of the general syntax:

```
Function(Object,Attribute) Operator Function(Object,Attribute)
```

You can define the precedence of evaluation using up to five levels of open and closed parentheses around constraint clauses.

Functions

The following functions are used in constraint clauses to evaluate attributes of the configuration:

COUNT Counts the number of times the attribute occurs within the configuration.

MIN	Returns the minimum value for items with the attribute.
MAX	Returns the maximum value for items with the attribute.
QTYSUM	Sums the quantity per parent model of items with the attribute.
SUM	Sums the total order quantity of items with the attribute.
VALUE	Returns a specific value for the item with the attribute.
ATTRIBUTE VALUE LIST	Returns a list of values for a single selected object attribute.
LITERAL LIST	Returns the list of alphanumeric values specified.
CHARACTER	Returns the specified character string.
NUMBER	Returns the specified number.
USER VALIDATION	Returns the user validation. See: PL/SQL Script 1: page 3 – 14 or PL/SQL Script 2: page 3 – 25

Null Attribute Action

The Null attribute action specifies how to process functions that cannot be evaluated.

- True: Sets the clause to True.
- False: Sets the clause to False.
- Stop: Stops the evaluation and skips to the next constraint.

Note: If the object and attribute referenced in the function does not exist on the order, the Configurator evaluates the functions MIN, MAX, VALUE, and LIST based on the Null Attribute Action. The functions COUNT, SUM, and QTYSUM evaluate to 0.

Objects

Objects include:

- ITEMS
- SALES_ORDER_HEADER
- SALES_ORDER_LINES

- USER–ID (Web Configurator)

Attributes

Attributes are assigned to objects within your product structure. When you define constraints, you reference these attributes.

You can define an unlimited number of configuration attributes. See: Overview of Configuration Attributes: page 2 – 4

You define attribute values for each object attribute. Then, when you configure products, attributes and their values are evaluated by your product constraints.

You can optionally create regions to group configuration attributes to ease maintenance of attribute values.

Operators

You can compare two functions using the following comparison operators:

>	Greater than.
<	Less than.
>=	Greater than or equal to.
<=	Less than or equal to.
=	Equal to.
!=	Not equal to.
IN	Evaluates whether a specified value exists in the specified list.
NOT IN	Evaluates whether a specified value does *not* exist in the specified list.
NA	No comparison made. Used for autoselection and autoexclusion constraints that use a number of constraint values in the optimization expression, but do not need comparisons. Clauses built with the NA operator do not require a right side.

Complex Constraints Using And / Or

Individual constraint clauses can be combined, using the Boolean operators And and Or, to form complex constraints.

Example 1 Assume you need a constraint to ensure that the memory ordered for a laptop PC is sufficient for the most demanding software ordered. Also, add a constraint to recommend additional memory to increase performance. A valid constraint clause for these two rules may look like the following:

```
((amount of memory ordered) = (amount of memory required)
AND
(recommended memory) > (amount of memory required))
```

See Also

Defining Constraints: page 2 – 20

Clause Evaluation

During product configuration validation, the left and right side of the constraint clause are compared. This evaluation returns a clause value of either True or False. Each function of a constraint clause is evaluated for the object attribute. This results in either a number or a character, as follows:

Function	Object/Attribute Type	Return Value Type
COUNT	All item attributes except List and Item Name	Number
SUM	Numeric attributes of items	Number
QTYSUM	Numeric attributes of items	Number
MIN	All item attributes except List and Item Name	Number or Character
MAX	All item attributes except List and Item Name	Number or Character
VALUE	All object / attributes except List	Number or Character
USER VALIDATIONS	All object / attributes except List	Number or Character
LIST	Attribute value lists and literal lists	Set of values
NUMBER	Literal numbers	Number
CHARACTER	Literal characters	Character

Table 2 – 1

Constraint Scope

Option Classes and Models

Constraints can be assigned to items whose BOM item type is set to Model or Option Class. Constraints automatically evaluate all selected items beneath the model or option class item to which the constraint is assigned.

You can assign constraints hierarchically to option classes and models under other option classes and model structures.

You can also assign the same constraint to many option classes or model structures. You can assign any number of constraints to a particular option or model.

Configurator Systems

You can assign constraints to a configurator system type, and can associate that system type with a system in Oracle Service. The Product Configurator automatically evaluates all order lines and configurations in the system.

See Also

Defining Constraints: page 2 – 20

Constraint Types

You can define messages and actions to take effect when constraint action values are met. Enter the constraint action value True or False in the Evaluate If field in the Assign Constraints window. You can define the type and content of the messages, define actions to prevent further order processing, and define items to add to the order.

Suggestion You can select this constraint type for any constraints assigned to option classes, models, and configurator system types.

Once you validate the order, and the constraint action is met, a message window displays "Suggestion" and your message.

You can continue processing the order and are not required to make any further corrections.

Warning

You can select this constraint type for any constraints assigned to option classes, models, and configurator system types. Once you validate the order, and the constraint is violated, a message window displays “Warning” and your message.

You can continue processing the order and are not required to make any further corrections.

Overridable Error

You can select this constraint type for any constraints assigned to option classes, models, and configurator system types. Once you validate the order, and the constraint is violated, a message window displays “Error” and your message.

You can override the error, continue processing the order, and are not required to make any further corrections. If you do not override the error, booked orders are automatically placed on a line level configuration hold to prevent further processing.

Error

You can select this constraint type for any constraints assigned to option classes, models, and configurator system types. Once you validate the order, and the constraint is violated, a message window displays “Error” and your message. Booked orders are automatically placed on a line level configuration hold to prevent further processing. You must correct the error to continue processing the order.

AutoSelection

You can select this constraint type for any constraints assigned to option classes and models. You can define any number of items to add to the configuration based on your product constraints.

Once you validate the order, a message window displays your message. The Product Configurator automatically selects defined autoselection options when the security rules permit adding options, and places the order on a line level configuration hold when the order entry security rules prevent adding options.

AutoExclusion You can select this constraint type for any constraints assigned to models. You can define any number of items to exclude from view in the configurator window.

AutoExclusion constraint types do not display messages.

See Also

Defining Constraints: page 2 – 20

Constraint Sequencing

There are four broad classifications of constraint sequencing referred to as constraint groups: at initialization, and before, during, and after autoselections. These correspond to preliminary validation checking, item autoselection, and overall compatibility validation of selected options, respectively. This level of constraint sequencing should not be confused with the sequence for constraint evaluation *within* each constraint group.

Initialization group constraints are evaluated upon navigation to the Configurator window.

Constraints evaluated before autoselections check basic business rules that preclude further evaluation. That evaluation could generate messages, suggestions, or violations. Violations at this stage could potentially cause numerous errors at later stages if, for example, more power cords are ordered for a PC than available power supplies.

Constraints evaluated after autoselection are for final configuration validation only. Minimum configuration requirements can be validated as well as compatibility requirements between selected and autoselected items.

Table 2 – 2 summarizes the constraint types and actions associated with each constraint group:

Group	Constraint Types	Actions
Initialization	Autoselection Autoexclusion	Initialization constraints are evaluated before navigation into the Configurator window.
Before Autoselections	Suggestion, Warning, Error, Overridable Error	The violation of Error and Error Override constraints (unless overridden) prevent evaluation of AutoSelection and After AutoSelection constraint groups. If the order is booked, a configuration hold is applied.
During autoselections	Autoselection	Options are automatically selected, or a configuration hold is applied and After AutoSelection constraints are not evaluated.
After autoselections	Suggestion, Warning, Error, Overridable Error	The violation of Error and Error Override constraints (unless overridden) prevent evaluation of autoselection and After autoselection constraint groups. If the order is booked, a configuration hold is applied.

Table 2 – 2

See Also

Defining Constraints: page 2 – 20

Validation

Automatic Validation

Orders are automatically validated when you:

- book orders
- book orders through order import
- enter change orders to booked configurations using the Option or Configurator windows
- cancel items on configuration orders

Validation on Request

You can validate orders an unlimited number of times at any point during the ordering procedure.

▶ **To validate an order on request:**

- Select the Validate button in the Configurator window or select Validate Configuration from the Special menu.

See Also

Entering Orders, *Oracle Order Entry/Shipping User's Guide*

Viewing Configuration Messages: page 3 – 12

Defining Constraints

You can define complex constraints by combining one or more constraint clauses together using the And and Or operators. You can also use parentheses around constraint clauses to control the precedence of evaluation.

You can define a new constraint by manually constructing constraint clauses. Or you can copy existing constraints and then edit the copied constraint clauses.

▶ **To define constraints:**

1. Navigate to the Constraints window.
2. In the Find Constraints window, choose the New button to define a new constraint.

   To update existing constraints, select Item to find constraints assigned to models or option classes, System to find constraints assigned to configurator system types, or both. Enter other search criteria and choose the Find button.

3. Enter a name and description for the constraint.

   Note: You can copy the type, message, and all constraint clauses from an existing constraint by choosing Copy Constraint from the Special menu.

4. Select the constraint type.
5. Enter the message text.
6. For each constraint you are defining, construct one or more constraint clauses.

▶ **To copy a constraint:**

1. Navigate to the Constraints window.
2. In the Find Constraints window, choose the New button to define new constraints.
3. Enter a name and description for the constraint.
4. Choose Copy Constraint from the Special menu.
5. Select an existing constraint to copy the constraint type, message text, and all constraint clauses into the constraint you are now defining.

   You cannot copy a constraint once you have selected a type or message text.

▶ **To construct constraint clauses:**

1. Unless this is the first constraint clause for the definition, select either the And or Or operator.
2. Optionally, enable an open parenthesis "(".
3. Select the first function. See: Constraint Clauses: page 2 – 11.
4. Select an object:

   Item: User–defined attributes of items or item name.

   User: Additional security for Web Configurator.

   Sales Order Header: Customer Name or Country Code.

   Sales Order Line: Sales order line descriptive flex segments.
5. Enter an attribute for the first function to operate on.
6. Select an operator. See: Constraint Clauses: page 2 – 11.
7. Select the second function.
8. Select a second object. If the second function is Number or Character, the second object is blank.
9. Enter an attribute or a value for the second function to operate on.

10. If you entered an open parenthesis, enter a closed parenthesis ")".
11. Repeat the above steps for each constraint clause you are constructing. Each clause, except for the first, must be separated by either an And or an Or operator.
12. Select a null attribute action to take if there is a null attribute condition.

    A null attribute condition exists when a constraint clause evaluates for an attribute and there are no items selected that have that attribute. For these situations, select an action to take:

    Stop: evaluation is stopped for current constraint and skips to the next constraint

    True: a true condition is assumed for the clause

    False: a false condition is assumed for the clause

See Also

Overview of Configuration Constraints: page 2 – 11

Constraint Clauses: page 2 – 11

Constraint Types: page 2 – 15

Viewing Constraint Attribute Values: page 2 – 22

Viewing Constraint Attribute Values

View constraint attributes and their values for any constraint assignment.

▶ **To view constraint attribute values:**

1. Navigate to the Assign Constraints window.
2. Query for the constraints to view.
3. Drill down on the constraint you want to view.

See Also

Overview of Configuration Constraints: page 2 – 11

Constraint Clauses: page 2 – 11

Assigning Constraints

You can assign any number of complex constraints to option classes, models, or configurator system types. However, you cannot have more than one assignment where the constraint, item, organization, and autoselect group are the same.

During validation, constraint assignments are taken on configurator system types from the OE: Validation Organization profile option.

For each constraint assignment to an option class, model, or configurator system type, you can define a message.

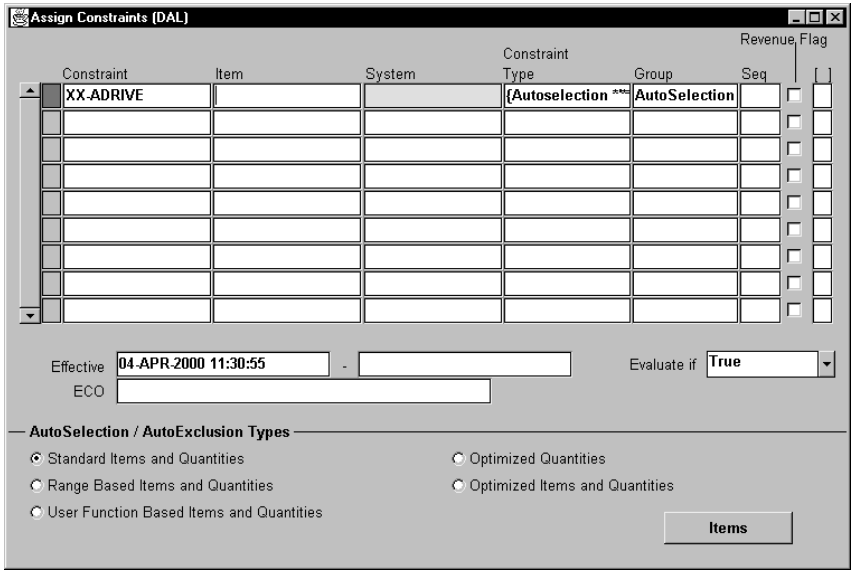
For each constraint assignment of the autoselection / autoexclusion type, you can define any number of autoselection / autoexclusion items.

Prerequisites

- ❑ Define items to assign constraints.
- ❑ Define constraints to assign to model or option class items, or to configurator system types.

▶ **To assign a constraint:**

1. Navigate to the Constraint Assignments window.



2. In the Find Assignments window, choose the New button to create a new constraint assignment. To update existing constraint assignments, enter selection criteria and choose the Find button.
3. Select a constraint.
4. Enter either an item or a configurator system type to assign the constraint to. Autoselection constraints cannot be assigned to a configurator system type.
5. Enter a constraint group: Initialization, Before AutoSelect, AutoSelect, After AutoSelect (default).
6. Enter an execution sequence.
7. Optionally, enter an effective date range.
8. Select an Evaluate If value, True (default) or False.

   If set to True and the constraint evaluates to True, the constraint message text appears in the Messages window after validation. If an autoselection is defined, the Product Configurator evaluates it.

   If set to False and the constraint evaluates to False, the constraint message text appears in the Messages window after validation. If an autoselection is defined, the Product Configurator evaluates it.

   If set to True and the constraint evaluates to False, or vice versa, the message text does not display and the autoselect evaluation is not performed.
9. Optionally, enter an Engineering Change Order (ECO) to associate with the constraint assignment. This field is intended for use only as a note. The text you enter here is not validated or processed.
10. If the constraint group is autoselect, select a type:

    Standard Items and Quantities: Specified items and quantities are autoselected for the configuration.

    Range Based Items and Quantities: Uses an optimization expression (numeric only) to return a value. Items and quantities are selected based on ranges of the returned value.

    Optimized Quantities: Uses an optimization expression to return a value. Specified items are selected based on ranges of the returned value and attributes. Quantities are calculated based on the numeric expression, range, and attribute.

    Optimized Items and Quantities: Uses an optimization expression to return a value.

- For numeric expressions: Items are selected by the configurator to match the value returned from the optimization expression, the range, and the attribute. Quantities are calculated based on the numeric expression, range, and attribute.
- For character expressions: Items are selected by the configurator to match the value returned from the optimization expression, the range, and the attribute. Quantity is derived from the BOM default quantity.

User–Returned Items and Quantities: Items and quantities are returned from a user–defined database procedure. The procedure is executed. Items and quantities returned by the user procedure are validated against the orderable bill of materials. See: PL/SQL Script: Sample Package Specifications: page 3 – 14 or PL/SQL Script: Sample Package Body: page 3 – 25

11. If the constraint group is Autoexclude, select a type:

    Standard Items: Specified items and quantities are autoselected for the configuration.

    Range Based Items: Uses an optimization expression to return a value. Items are selected based on ranges of the returned value.

    Optimized Items: Uses an optimization expression to return a value. Items are selected by the configurator to match the value returned from the optimization expression, the range, and the attribute.

    User–Returned Items: Items and quantities are returned from a user–defined database procedure. The procedure is executed. Items and quantities returned by the user procedure are validated against the orderable bill of materials. See: PL/SQL Script: Sample Package Specifications: page 3 – 14 or PL/SQL Script: Sample Package Body: page 3 – 25

12. Choose the Items button to assign autoselection and autoexclusion items. See: Assigning AutoSelection and AutoExclusion Items: page 2 – 26.

See Also

Constraint Clauses: page 2 – 11

Constraint Scope: page 2 – 15

Constraint Types: page 2 – 15

Constraint Sequencing: page 2 – 17

Defining Constraints: page 2 – 20

Assigning AutoSelection Items: page 2 – 26

Assigning Autoselection and Autoexclusion Items

For constraints assigned to the autoselection / autoexclusion constraint types, you must assign autoselection and autoexclusion items.

Autoselection constraints can be assigned to any number of option class or model items, and any number of autoselection constraints can be assigned to an option class or a model item.

Autoexclusion constraints can be assigned to any number of model items, and any number of autoexclusion constraints can be assigned to one model item.

You can define any number of autoselection / autoexclusion items for each constraint assignment.

Prerequisites

- ❑ You must define an autoselection constraint. See: Defining Constraints: page 2 – 20 and Constraint Types: page 2 – 15.

▶ **To assign autoselection / autoexclusion items:**

1. Navigate to the AutoSelection / AutoExclusion Items window. Do this by choosing the Items button from the Constraint Assignments window.

   This is only possible if you chose to assign the constraint to the initialization or autoselection constraint groups.

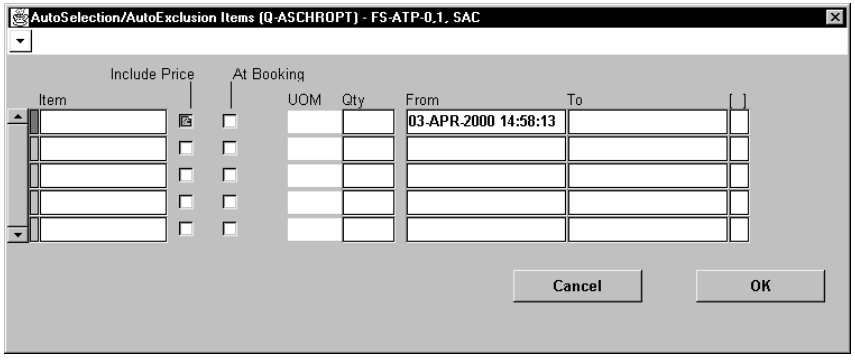
2. Depending upon which autoselection / autoexclusion option you selected in the Constraint Assignments window, do one of the following:

   - Assign autoselection items for Standard Items and Quantities
   - Assign autoexclusion items for Standard Items
   - Assign autoselection items for Range Based Items and Quantities
   - Assign autoexclusion items for Range Based Items
   - Assign autoselection items for Optimized Quantities
   - Assign autoselection items for Optimized Items and Quantities

- Assign autoexclusion items for Optimized Items
- Assign autoselection items for User–Returned Items and Quantities
- Assign autoexclusion items for User–Returned Items

▶ **To assign autoselection / autoexclusion items for standard items and quantities:**

1. Select an autoselection / autoexclusion item from the bill of material list of values, which displays the following data:
   - item name
   - default extended bills of material quantity
   - mutually exclusive
   - optional
   - effective date range
   - minimum extended bills of material quantity
   - maximum extended bills of material quantity
   - item description



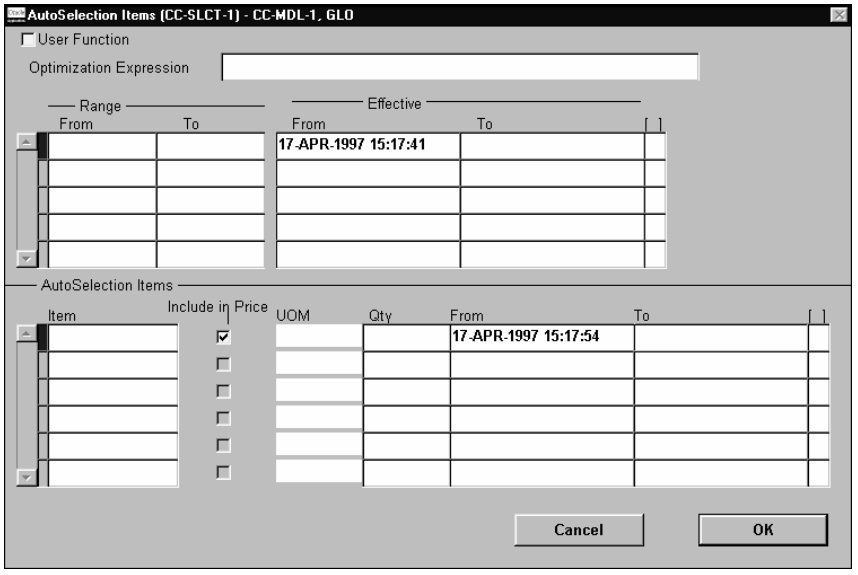
2. Indicate whether to include the item in the price of the product.
3. Indicate whether to autoselect the item at booking or when the order is validated (default).
4. For autoselection items only, enter the quantity of the item.
5. Enter an effective date range for the item.
6. When finished, choose OK.

For autoselection, items and quantities are autoselected for each range that contains your expression value. For autoexclusion, items that contain your expression value are excluded from display in the Configurator window.

▶ **To assign autoselection / autoexclusion items for range based items and quantities:**

1. Optionally, if you want the value of the of the optimization expression returned by a user–defined PL/SQL procedure, enable the User Function check box.

   Otherwise, leave User Function disabled and enter an optimization expression. See: Optimization Expressions: page 2 – 32.



2. Enter value ranges for the result of the optimization expression. You must specify a From value; a value for To is optional.

   Note: Autoselection / autoexclusion ranges can overlap.

3. Enter and effective date range.

4. Enter one or more autoselection / autoexclusion items for each range or select the item from the indented bills of material list of values.

5. For autoselection only, indicate whether to include the item in the price of the product.

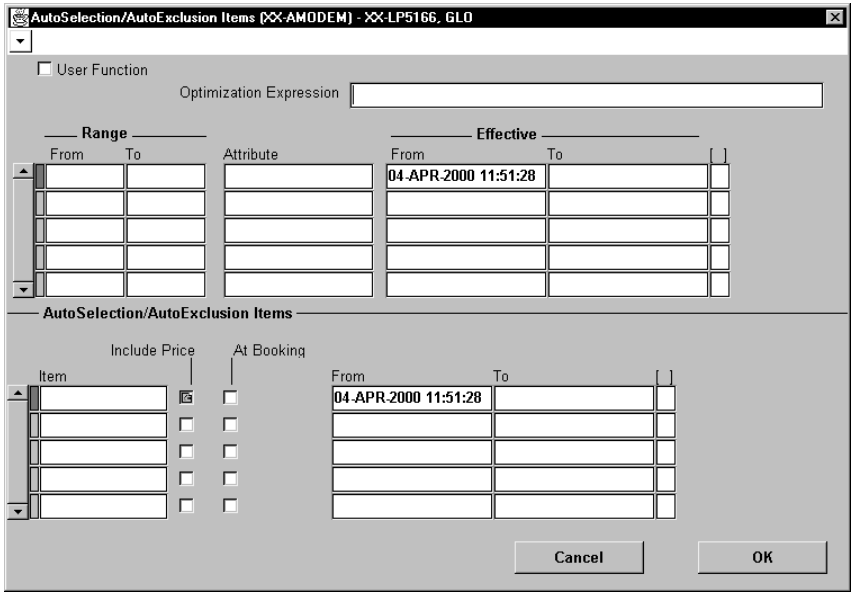
6. For autoselection only, indicate whether to autoselect the item at booking or when the order is validated (default).
7. For autoselection only, enter the quantity of the item
8. Enter an effective date range for the item.
9. When finished, choose OK.

   The optimization expression is evaluated. The value returned by the expression is compared to the range you specified. For autoselections, items and quantities are autoselected for each range that contains your expression value. For autoexclusions, items that contain your expression value are excluded from display in the Configurator window.

▶ **To assign autoselection items for optimized quantities:**

1. Optionally, if you want the value of the of the optimization expression returned by a user–defined PL/SQL procedure, enable the User Function check box.

   Otherwise, leave User Function disabled and enter an optimization expression. See: Optimization Expressions: page 2 – 32.



2. Enter a value range for the result of the optimization expression. You must specify a From value; a value for To is optional.

Note: Autoselection ranges can overlap.

3. Select an attribute.

   Note: Optimization expressions can only be numeric for the optimized quantities autoselection type. Consequently, only numeric attributes display in the list of values.

4. Enter an effective date range.
5. Enter one or more autoselection items.
6. Indicate whether to include the item in the price of the product.
7. Enter an effective date range for the item.

   The numeric optimization expression is evaluated. The value returned by the expression is compared to the ranges you specified. Items are selected and quantities are determined for each range that contains your expression value.

   Item quantities are evaluated as follows:

   ```
   expression value / attribute value
   ```

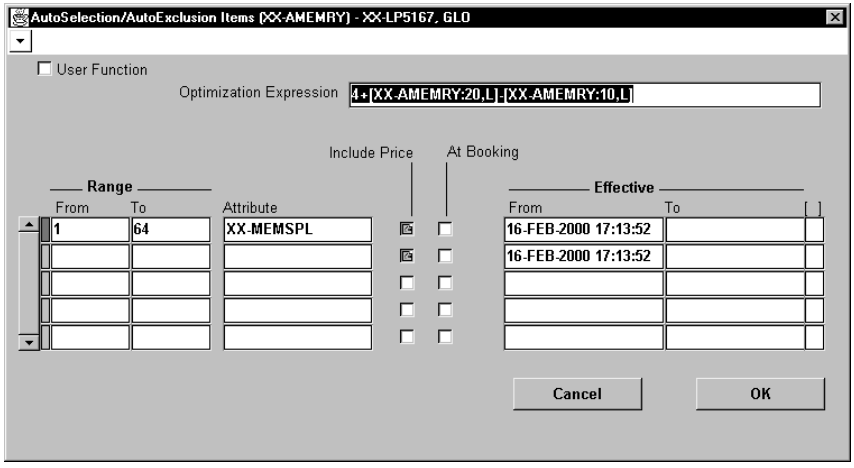
   where the attribute value is the value of the attribute for the autoselection item. The item with the lowest allowable integer quantity is selected.

8. When finished, choose OK.

► **To assign autoselection / autoexclusion items for optimized items and quantities:**

1. Optionally, if you want the value of the of the optimization expression returned by a user–defined PL/SQL procedure, enable the User Function check box.

   Otherwise, leave User Function disabled and enter a character or numeric optimization expression. See: Optimization Expressions: page 2 – 32.



2. Enter a value range for evaluation with the result of the optimization expression. You must specify a From value; a value for To is optional.

   Note: Autoselection / autoexclusion ranges can overlap.

3. Select an attribute.

4. Enter an effective date range for the item.

   For numeric expressions, the value returned by the expression is compared to the ranges you specified. For autoselections, items with the attribute you specify are evaluated and quantity is determined for each range that contains your expression value. For autoexclusion, the item is excluded from display in the Configurator window.

   A quantity for each item that has the attribute is evaluated. Quantities are evaluated as follows:

   `expression value / attribute value`

where the attribute value is the value of the attribute for the autoselection item. The item with the lowest allowable integer quantity is selected.

For character expressions, each item that has the attribute you specify with a value equal to the optimization expression value is identified. For autoselection, each item is selected with the default extended bill of material quantity. For autoexclusion, the item is excluded from the Configurator window.

5. When finished, choose OK.

▶ **To assign autoselection / autoexclusion items for user–returned items and quantities:**

1. Optionally, if you want the value of the of the optimization expression returned by a user–defined PL/SQL procedure, enable the User Function check box. See: Optimization Expressions: page 2 – 32.
2. Enter the name of the user–defined PL/SQL procedure in the function box.
3. When finished, choose OK.

See Also

Constraint Scope: page 2 – 15

Constraint Types: page 2 – 15

Constraint Sequencing: page 2 – 17

Defining Constraints: page 2 – 20

Assigning Constraints: page 2 – 23

Optimization Expressions: page 2 – 32

Optimization Expressions

Those constraints for which the autoselection / autoexclusion constraint type is selected must have an optimization expression (unless you selected the Standard Items and Quantities autoselection / autoexclusion type). The optimization expression returns a value which is used in the evaluation of the items and/or quantities of items.

These items and quantities are then automatically added to the configuration or excluded from the Configurator window.

Optimization expressions can be numeric or character expressions and return a single value. They can also indicate a user–returned value based on a user–defined database stored procedure. The expression value for this autoselection / autoexclusion type is validated against the stored database procedure and checked for a single return value. See: PL/SQL Script: Sample Package Specifications: page 3 – 14 or PL/SQL Script: Sample Package Body: page 3 – 25

Numeric Expressions

Numeric expressions cannot contain alphanumeric values. Standard multiplication, division, addition, subtraction, and the truncation function are supported. Standard precedence of operation is enforced; multiplication and division are performed first, then addition and subtraction. Evaluation can be controlled by using parentheses to group expressions.

> **Note:** Numeric expressions that use division and reference clause names in the denominator could result in divide by 0 errors. These are treated as failed autoselections.

Numeric expressions can reference individual numeric functions within a constraint clause. Refer to the Syntax for Referencing Clause Functions section.

Character Expressions

Character expressions are used in range based and optimized items and quantities constraint types. Literal values must be enclosed in single quotes. Concatenation is supported.

Character expressions can use both numeric or character clause name references. All literal values and clause name references must be concatenated and are validated as follows:

- clause name references must exist in the autoselection / autoexclusion constraint
- each expression segment must be concatenated
- literal values must be enclosed in single quotes

Syntax for Referencing Clauses

A constraint clause has a first (”left”) function and a second (”right”) function. The left and right sides are separated by the operator.

The optimization expression can parse the left, the right, or both sides of the constraint clause. For example:

```
SUM(memory ordered) < SUM(memory required)
```

where *memory ordered* is the left function and *memory required* is the right function. The less than operator separates the left function from the right.

The syntax for referencing individual functions of the constraint clause in the optimization expression is:

```
[clause_name,x]
```

where *clause_name* is the name of the clause and *x* is an uppercase ’L’ or an uppercase ’R’ to indicate the left or right side of the clause. This entire reference must be enclosed by brackets.

The clause name (referenced in the Constraints window) defaults to:

```
constraint_name:clause_sequence
```

where *constraint_name* is the name of the constraint, and *clause_sequence* is a sequence number (10, 20, 30, and so on) that defaults when the clause is written. The first clause written is 10, the second is 20. The third clause written is 30, even if it is inserted between the first and second. The default clause name and clause sequence can be overridden by the user with a user defined clause name.

When the clause name is defaulted, a colon separates the clause name from the sequence number for the purposes of referencing in an optimization expression. For example, the default name for the second clause of the MEM constraint is:

```
MEM:20
```

In the optimization expression, a comma (,) separates the clause name from the left or right side designator. The following example references the left side of the second clause of the MEM constraint:

```
[MEM:20,L]
```

Note: The optimization expression cannot contain blank spaces.

Example A complete optimization expression using the MEM constraint and two constraint clauses, lines 10 and 20, might look like:

```
2+[MEM:20,L]*[MEM:10,R]
```

This optimization expression multiplies the left function of the second clause (MEM:20,L) by the right function of the first clause (MEM:10,R) and adds 2 to the result.

Product Configurator Profile Options

During implementation, the system administrator sets up and maintains profile options.

Profile Option Settings

You can set or view the following profile options in Oracle Product Configurator. The table also includes profile options from other applications that are used by Oracle Product Configurator.

Key	
✓	You can update the profile option.
–	You can view the profile option value but you cannot change it.

Profile Option	User	System Administrator				Requirements	
	User	User	Resp	App	Site	Required?	Default Value
CZ: Use Simple Configurator	–	✓	✓	✓	✓	Required	Yes
CZ: Automatically Validate on Exit	–	✓	✓	✓	✓	Required	Yes
BOM: Configurator Server Validation	–	✓				Optional	No
BOM: Save Configuration Details Flag	–	✓				Optional	No
BOM: Check for Duplicate Configuration	–				✓	Optional	No
BOM: Allow Multilevel ATO	–				✓	Optional	No
BOM: Configurator Installed	–				✓	Optional	No
OE Configurator Display Mode	–	✓				Optional	No

CZ: Use Simple Configurator

Determines whether the web–based Product Configurator uses simple HTML or JAVA coding. Yes (default) indicates HTML and No indicates JAVA.

CZ: Automatically Validate on Exit

Determines whether Product Configurator automatically validates an order when the user chooses the OK button.

BOM: Configurator Server Validation

Determines whether validation in the Configurator window is done on the server (Yes) or on the client (No).

BOM: Save Configuration Details Flag

Determines whether the Product Configurator saves attribute value details.

BOM: Configurator Installed

Determines whether the Product Configurator is installed.

OE: Configurator Display Mode

Sets the configurator form to display only the selected items.

CHAPTER

3

Configuring Products

This chapter describes how to configure products using Oracle Product Configurator. Task descriptions provide you with detailed information about the following:

Configuring a Product

The Configurator window displays indented model structures that you can expand or collapse.

When configuring products, you can use Order Entry functions for option lines in configurations:

- maintain and apply pricing discounts
- create and maintain schedule details
- create and maintain shipping information
- create and maintain service information
- take scheduling actions.

▶ **To configure products:**

1. Navigate to the Configurator folder window. You can do this in two ways:
   - from the Sales Order window in Oracle Order Entry, enter a model to configure on a sales order line and choose the Configure button.

     Note: If defining a folder, you must select never query.
   - from the Bills of Material window in Oracle Bills of Material, choose Configure Bill from the Special menu.
2. Choose the Configurator button to open the Configurator window.

   The indented bill structure appears, showing mandatory option classes in green and exclusive options in italics.
3. Configure the model:
   - by selecting the item's check box.
   - by choosing the GoTo Item button and selecting the item from the list of values.
   - by entering a new quantity for the option. The default quantity equals the component quantity multiplied by the quantity of the parent item.

     Note: When a child item is selected, it's parent item is also automatically selected.
   - disabling the item's check box.
   - disabling the last child item automatically disables the parent.

- by setting the quantity to 0 or blank.

Note: If you disabled the check box, the quantity is set to null in Bills of Material, and null or 0 in Order Entry (based on security rules). If you change the quantity to null in Bills of Material and null or 0 in Order Entry (as permitted by security rules), the check box is disabled.

4. Optionally modify the quantity.

   The following restrictions apply:

   - quantity must be a multiple of the parent quantity (1 if the parent item is the model line)
   - quantity cannot be less than that of the parent
   - quantities modified in a parent item automatically modify proportionally quantity on selected child items
   - quantities of option class items with a basis set to Option Class cannot be modified
   - only one item in an option class that is mutually exclusive can be selected

5. Choose the Validate button to validate the configuration. See: Validating Configurations: page 3 – 4.

6. Choose the Save button to save the configuration.

   To cancel the current configuration, choose the Cancel button. The Configurator window reverts to the last saved configuration.

See Also

Entering Orders, *Oracle Order Entry/Shipping User's Guide*

Using the Configurator from Order Entry: page 3 – 10

Validating Configurations

▶ **To validate a configuration:**

- After configuring your product, choose the Validate button.

  When validation is complete, the Configuration Messages window appears. If defined constraint action values are met, the message text appears.

  Note: You can click on the message text to view option items evaluated by the constraint. Also, you can click in the record line indicator (drill down box) of the message line to view the constraint syntax, items, and attribute values evaluated in the Constraint Detail window

See Also

Viewing Configuration Messages: page 3 – 12

Validation: page 2 – 18

Product Configurator for the Web

If you installed Oracle Web Customers, your customers can use a standard web browser to place valid orders for complex products with many features and options. The Web Configurator automatically validates each order to prevent invalid order from being introduced into your order processing system.

Configurator Systems

Oracle Product Configurator allows you to assign a system number to multiple order lines, either within a single order or across several orders. Consequently, you can validate a single model line or the entire system. You can assign any number of constraints to your configurator system types. If you are an Oracle Service customer, you can define a system from the Sales Orders form.

See Also

Defining Configurator System Types: page 3 – 6

Assigning Constraints to Configurator Systems: page 3 – 7

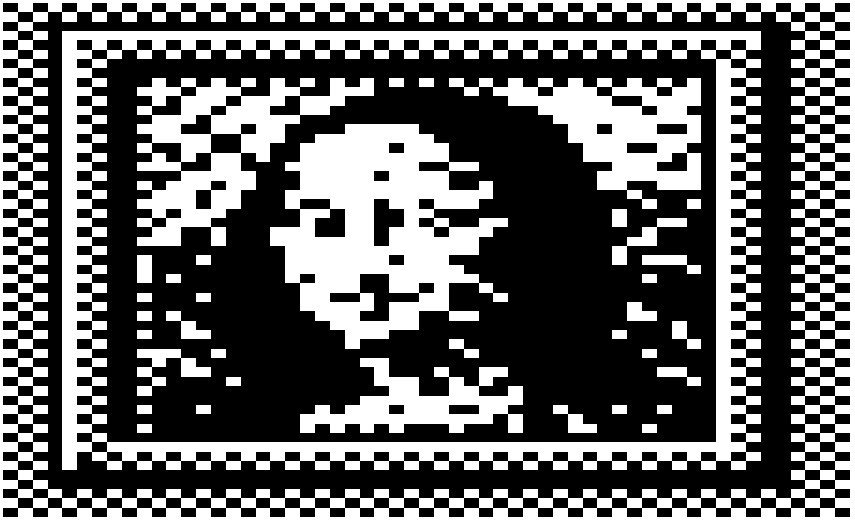
Validating a System: page 3 – 8

System Add–ons and Upgrades: page 3 – 9

Defining Configurator System Types

▶ **To define configurator system types:**

1. Navigate to the Configurator System Types window.



2. Enter the system information.

3. Enter effectivity dates.

See Also

Assigning Constraints to Configurator Systems: page 3 – 7

Validating a System: page 3 – 8

Assigning Constraints to Configurator Sytems

Prerequisites

- ❑ You must first define constraints in the Product Configurator. See: Defining Constraints: page 2 – 20

▶ **To assign constraints to configurator systems:**

1. Navigate to the Assign Constraints window. Do this by selecting Constraint Assignments from the navigator.
2. Enter a configurator system.
3. Query an existing constraint.

   Note: Autoselection constraint types cannot have a system type.
4. If you would like to control the order which constraints are validated, enter a sequence number.

See Also

Defining Configurator System Types: page 3 – 6

Validating a System: page 3 – 8

Validating a System

When an order is booked, Product Configurator will automatically validate all constraints based on the order quantity of each line associated with your systems and the constraints assigned to your configurator system type.

Prerequisites

- ❑ You must define the configurator system types that are included in the configurator system type you are trying to validate.
- ❑ You must assign constraints to the Configurator System Types.
- ❑ If you are validating against the installed base in Oracle Service, you must link the Service System Number with the Oracle Product Configurator System Type.

▶ **To validate a configurator system:**

1. Navigate to the Configurator window. Make sure your system is in the Primary region.

   Note: You can request validation of your system from the Sales Orders or Configurator windows. You can also select Validate Configuration from the Special menu.

   Attention: You can only validate a system from the Sales Order Special menu if your sales order line is a standard item.

2. Choose the Validate button.

   Note: If a sales order line is part of a configurator system and this system has constraints, then all of these constraints will be evaluated.

See Also

Defining Configurator System Types: page 3 – 6

Assigning Constraints to Configurator Systems: page 3 – 7

System Add–ons and Upgrades

Oracle Service customers are able to keep track of the installed systems of their customer base. As a result, you can take an order for an add–on option or replacement option for an existing installed system and validate the order while taking into account the options on the installed system.

Validating System Add–ons and Upgrades

Prerequisites

- ❑ You must have the Oracle Service application installed.
- ❑ To validate add–ons against installed systems the system must be in the installed base.

▶ **To validate system add–ons and upgrades on existing systems:**

1. See: Validating a System: page 3 – 8

See Also

Defining Configurator System Types: page 3 – 6

Assigning Constraints to Configurator Systems: page 3 – 7

Using the Configurator from Oracle Order Entry

If you have the Oracle Product Configurator installed, you can perform the following Configurator functions from Order Entry.

- Validate configuration orders from the Configurator window accessed from the Sales Orders window
- Book new orders from the Sales Orders window or the Order Import program. The Configurator performs the validation of the new orders.
- Make changes to existing booked configuration orders from either the Sales Orders or Cancel Orders windows. The Configurator validates the new configuration and places invalid configurations on hold to prevent further processing.
- Correct invalid orders and remove configuration holds.
- Manually remove holds to allow processing to resume without correcting the invalid configuration. If errors are not corrected, re–validating the configuration places the order on hold again.
- Check for existing matching configurations and optionally reserve against it at the time of order entry. See: Using Match and Reserve: page 3 – 11

Both the option selection and validation functions are available from the Special menu of Oracle Order Entry.

See Also

Configuring a Product: page 3 – 2

Entering Orders, *Oracle Order Entry/Shipping User's Guide*

Using Match and Reserve

While entering an order line you can search for matching active Assemble–To–Order (ATO) configurations and associated available–to–transact quantities based on the options selected for the line.

Prerequisites

- ❑ The order must be booked. See: Overview of Sales Orders, *Oracle Order Entry/Shipping User's Guide*
- ❑ The order line cannot have an associated configuration item assigned.
- ❑ The profile option *BOM: Check for Duplicate Configurations* must be set to *Match and Reserve.* See Product Configurator Profile Options: page 2 – 36

▶ **To match and reserve existing configurations:**

1. Navigate to the Sales Orders window.
2. Enter order information including configuration details.
3. Navigate to the model order line.
4. Select More from the Special menu.
5. From the resulting list of values, select Match ATO Configuration.

   Note: The Match ATO Configuration Inquiry locates and displays any existing configured items with exactly the same options as the configured model on your order line. The Match ATO Configurator Inquiry does not perform a reservation.
6. From the resulting list of values, select Match and Reserve ATO Configuration.

   Note: The Match and Reserve ATO Configuration function locates an existing configured item with exactly the same options as the configured model on your order line. Match and Reserve ATO Configurations checks for an existing configured item and reserves it for your order line.

Viewing Configuration Messages

You can override errors that are set to the overridable error type and determine which attribute and which value of a specific item is causing the error.

You can also use information in the messages summary to determine if the attribute and the value have been assigned correctly, and if the constraint was written correctly.

You can view configuration messages from:

Product Configurator > Configurator Messages

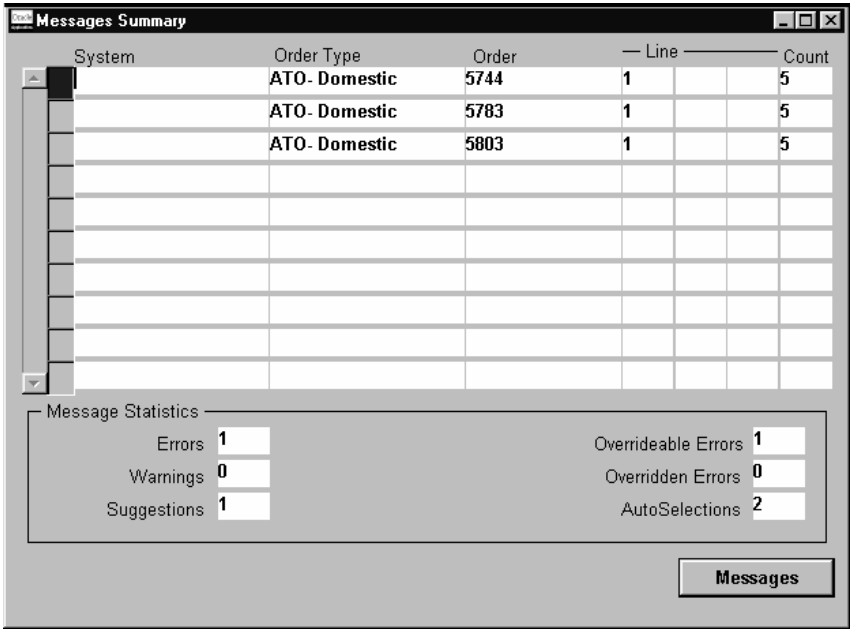
Order Entry > Sales Orders > Special menu > View Config Messages

Order Entry > Configurator window > Configurations Messages (after validation)

▶ **To view configuration messages:**

1. Navigate to the Messages Summary window.

   Note: If you are using Oracle Order Entry, configuring a product, and the Configuration Messages window appears, you can only view messages for the current order.



2. In the Find window, enter search criteria and choose the Find button.

   All orders and their associated order lines that have messages are listed in the Messages Summary window.

3. Select an order and choose the Messages button to open the Configurator Message window.

4. If the constraint type is overridable, you can override the error by choosing the Override button.

   Note: If the booked order has unresolved errors, a line–level configuration hold is automatically placed on the order line to prevent further processing.

   Choose the Summary button to return to the Messages Summary window and select another message.

   Choose the record line drill down box to open the Constraint Details window and view evaluated constraint clause(s), items, attributes, and attribute values.

See Also

Defining Attributes: page 2 – 6

Defining Constraints: page 2 – 20

Assigning Constraints: page 2 – 23

Assigning Autoselection Items: page 2 – 26

PL SQL Script: Sample Package Body

```
whenever sqlerror exit failure rollback;
create or replace package body cz_user_autoselection as
/*+=========================================================================+*/
/* $Header: CZUASLTB.pls 110.2 97/07/15 17:52:53 appldev ship $              */
/*+=========================================================================+*/
/* Copyright (c) 1997 Oracle Corporation Redwood Shores, California, USA     */
/*                       All rights reserved.                                */
/*                       Oracle Manufacturing                                */
/*+=========================================================================+*/
/*                                                                           */
/* Program:     CZUASLTB.pls                                                 */
/* Description: This file contains the prototypes for the following functions */
/*                                                                           */
/*             1. User_AutoSelection                                         */
/*            This serves as an extension to the built in capabilities       */
/*             of Oracle Product Configurator.                               */
/*                                                                           */
/*          The User AutoSelection function is enabled by selecting the      */
/*          User Function Based Items and Quantities AutoSelection           */
/*          Option from the Assign Constraints form. The Name of the         */
/*          PL/SQL Package is entered in the Function field of the           */
/*          AutoSelection Items window.  The Product Configurator engine     */
/*          calls the named package and validates the items and              */
/*          quantities the PL/SQL package returns against the orderable bill */
/*          of material.  Items must exist uniquely and the quantity must    */
/*          be permitted for that item in the orderable bill of material.    */
/*                                                                           */
/*         2. User_Optimize_Expression                                       */
/*         A packaged function that evaluates a complex optimization         */
/*         expression external to the Product Configurator,                  */
/*         and returns the value to the Product Configurator                 */
/*         to be used for the AutoSelection/AutoExclusion                    */
/*         Types: Range Based Items and Quantities, Optimized Quantities     */
/*         and Optimized Items and Quantities.                               */
/*                                                                           */
/*         The User_Optimize_Expression function is enabled by checking      */
/*         the User Function check box on the AutoSelection Items window     */
/*         and entering the Name of the PL/SQL Package in the                */
/*         Optimization Expression field of the same window.                 */
/*         The Product Configurator engine calls the named package which     */
```

```
/*         returns a character or numeric value to be used for              */
/*         evaluation with the defined ranges of the specified              */
/*        AutoSelection/AutoExclusion Type.                                 */
/*                                                                          */
/*        3. User_Clause_Function                                           */
/*         A packaged function that evaluates and returns a value of        */
/*         true/false/stop to the Product Configurator to be used by        */
/*        the evaluating class.                                             */
/*                                                                          */
/*         The User_Clause_Function is invoked by entering the name of the  */
/*         packagefunction in the User Validate Function column of the      */
/*         constraints form.  The function column in the Defined            */
/*         Constraint form should be 'User Validate' and the operator       */
/*        should be 'NA'.                                                   */
/*                                                                          */
/*           Context Information:                                           */
/*             The previous autoselections that have occured for the sales  */
/*             order line in this run of Configuration validation are       */
/*             viewable through the view CZ_COMPONENT_SELECTIONS_V and      */
/*             filtered based on the group_id which is passed in to the     */
/*            User_AutoSelection function. Item_svrid column contains the   */
/*             constraint assignment_id that has autoselected the item,     */
/*             while those with NULL value represent customer selections.   */
/*                                                                          */
/* Called By:   CZ_VALIDATE_CONFIGURATION (CZVCFGRS/B.pls)                  */
/*                                                                          */
/* Changed by:  Who               When    What                              */
/*              Tony Gaughan    9/17/97   Updated layout (standards)        */
/*                                                                          */
/****************************************************************************/
/****************************************************************************/
/* Function:       User_AutoSelection                                       */
/*                                                                          */
/*        Define user extensible rules to perform autoselections.           */
/*      AutoSelected items are inserted into BOM_CONFIG_EXPLOSIONS          */
/*                where the following rules must apply.                     */
/*                                                                          */
/*                Validation: (Done after exiting this procedure)           */
/*                                                                          */
/*                1. inventory_item_id, organization_id and sort_order      */
/*                    should be NOT NULL                                    */
/*                2. the record should represent a valid component on the   */
/*                   bill of the model ordered on the sales order line      */
/*                3. quantity should be NOT NULL and consistent with the    */
```

```
/*                    BOM component quantities                                */
/*                                                                            */
/* Parameters In:  pnLineID                                                   */
/*                 The sales order line that is being validated.  This        */
/*                 always points to the top model that has been ordered.      */
/*                                                                            */
/*                 pnAssignmentD                                              */
/*                 Unique identifier of constraint assignment that is being   */
/*                 evaluated                                                  */
/*                                                                            */
/*                 pnGroup ID                                                 */
/*                 Used to identify prior autoselections that have occured    */
/*                 in this run of validation                                  */
/*                                                                            */
/*            Out: xcErrorMessage                                             */
/*                 Message string which is displayed to user                  */
/*                                                                            */
/*                                                                            */
/* Returns:        Integer Value ( 0 – Failure, 1 – Success)                  */
/*                                                                            */
/******************************************************************************/
function User_AutoSelections(  pnLineID         in      number,
                               pnAssignmentID   in      number,
                               pnGroupID        in      number,
                               xcErrorMessage   out   varchar2 )
        return integer is
lnStatement  number := 0;                  -- define any locals
begin
  lnStatement := 10;
/******************************************************************************/
/*  Function Logic goes here                                                  */
/*  Execute custom code that will evaluate a complex optimization             */
/*  expression and return the value to be used                                */
/*  items to be added to the sales order line that is being validated         */
/******************************************************************************/


/******************************************************************************/
/*  Template for insert into Bom_Config_Explosions table                      */
/******************************************************************************/
/*

  insert into bom_config_explosions
  (
```

```
top_bill_sequence_id,
bill_sequence_id,
organization_id,
explosion_type,
component_sequence_id,
component_item_id,
item_svrid,
plan_level,
extended_quantity,
sort_order,
creation_date,
created_by,
last_update_date,
last_updated_by,
group_id,
session_id,
select_flag,
select_quantity,
top_item_id,
context,
attribute1,
attribute2,
attribute3,
attribute4,
attribute5,
attribute6,
attribute7,
attribute8,
attribute9,
attribute10,
attribute11,
attribute12,
attribute13,
attribute14,
attribute15,
component_quantity,
so_basis,
optional,
mutually_exclusive_options,
check_atp,
shipping_allowed,
required_to_ship,
required_for_revenue,
include_on_ship_docs,
```

```
  include_on_bill_docs,
  low_quantity,
  high_quantity,
  pick_components,
  primary_uom_code,
  primary_unit_of_measure,
  base_item_id,
  atp_components_flag,
  atp_flag,
  bom_item_type,
  pick_components_flag,
  replenish_to_order_flag,
  shippable_item_flag,
  customer_order_flag,
  internal_order_flag,
  customer_order_enabled_flag,
  internal_order_enabled_flag,
  so_transactions_flag,
  description,
  assembly_item_id,
  configurator_flag,
  rounding_factor,
  component_code,
  loop_flag,
  parent_bom_item_type,
  operation_seq_num,
  item_num,
  effectivity_date,
  disable_date,
  implementation_date,
  rexplode_flag,
  common_bill_sequence_id,
  comp_bill_seq_id,
  comp_common_bill_seq_id,
  num_col1,
  num_col2,
  num_col3,
  date_col1,
  date_col2,
  date_col3,
  char_col1,
  char_col2,
  char_col3
)
```

```
select top_bill_sequence_id,
       bill_sequence_id,
       be.organization_id,
       'OPTIONAL',
       be.component_sequence_id,
       component_item_id,
       x_assignment_id,
       plan_level,
       extended_quantity,
       sort_order,
       sysdate,
       1,
       sysdate,
       1,
       x_group_id,
       null,                                  -- session_id
       'Y',                                   -- select_flag
       1,                                     -- select_qty
       top_item_id,
       be.context,
       be.attribute1,
       be.attribute2,
       be.attribute3,
       be.attribute4,
       be.attribute5,
       be.attribute6,
       be.attribute7,
       be.attribute8,
       be.attribute9,
       be.attribute10,
       be.attribute11,
       be.attribute12,
       be.attribute13,
       be.attribute14,
       be.attribute15,
       component_quantity,
       so_basis,
       optional,
       mutually_exclusive_options,
       check_atp,
       shipping_allowed,
       required_to_ship,
       required_for_revenue,
       include_on_ship_docs,
```

```
        include_on_bill_docs,
        low_quantity,
        high_quantity,
        pick_components,
        primary_uom_code,
        primary_unit_of_measure,
        base_item_id,
        atp_components_flag,
        atp_flag,
        bom_item_type,
        pick_components_flag,
        replenish_to_order_flag,
        shippable_item_flag,
        customer_order_flag,
        internal_order_flag,
        customer_order_enabled_flag,
        internal_order_enabled_flag,
        so_transactions_flag,
        description,
        assembly_item_id,
        null,
        null,
        be.component_code,
        loop_flag,
        parent_bom_item_type,
        operation_seq_num,
        item_num,
        sysdate,
        disable_date,
        implementation_date,
        rexplode_flag,
        common_bill_sequence_id,
        comp_bill_seq_id,
        comp_common_bill_seq_id,
        null, -- sequence number
        num_col2,
        num_col3,
        date_col1,
        date_col2,
        date_col3,
        char_col1,
        char_col2,
        char_col3
from   so_lines_all    sol,
```

```
        bom_explosions  be
  where  sol.line_id             =  x_line_id
  and    be.top_bill_sequence_id =  sol.component_sequence_id
  and    be.component_code       =  sol.component_code
  and    be.organization_id      =  sol.warehouse_id
  and    be.explosion_type       =  'OPTIONAL'
  and    be.bom_item_type not    in ( 1, 2 )
  and    be.effectivity_date     <= sol.creation_date
  and    be.disable_date         >  sol.creation_date;
*/
 return (1);                                -- success
exception
  when others then
    xcErrorMessage := 'CZ_USER_SELECTION.User_AutoSelections(' ||
                           to_char( lcStatement) || '):' || SQLCODE || ':' ||
                           substr( SQLERRM, 1, 60 );
     return (0);                                -- failure
end User_AutoSelections;



/*****************************************************************************/
/* Function:     User_Optimization_Expression                                */
/*               User extensible function to allow complex optimization      */
/*              calculations                                                 */
/*                                                                           */
/* Parameters In:  pnLineID                                                  */
/*                 The sales order line that is being validated.  This       */
/*             always points to the top model that has been ordered.         */
/*                                                                           */
/*                 pnAssignmentD                                             */
/*                Unique identifier of constraint assignment that is being   */
/*                 evaluated                                                 */
/*                                                                           */
/*                 pnGroup ID                                                */
/*               Used to identify prior autoselections that have occured     */
/*                 in this run of validation                                 */
/*                                                                           */
/*            Out: xcExpressionValue                                         */
/*                  Optimization expression value to be returned             */
/*                                                                           */
/*               xcDataType                                                  */
/*             Data Type of returned expression (number, varchar2)           */
/*                                                                           */
```

```
/*                xcErrorMessage                                              */
/*                Message string which is displayed to user                   */
/*                                                                            */
/*                                                                            */
/* Returns:       Integer Value ( 0 – Failure, 1 – Success)                   */
/*                                                                            */
/******************************************************************************/
function User_Optimization_Expression(    pnLineID          in     NUMBER,
                                         pnAssignmentID     in     NUMBER,
                                         pnGroupID          in     NUMBER,
                                         xcExpressionValue   out   VARCHAR2,
                                         xcDataType          out   VARCHAR2,
                                         xcErrorMessage      out   VARCHAR2 )
        return integer is
lnStatement  number := 0;                    -- local variables
begin
  lnStatement := 10;
/******************************************************************************/
/*  Function Logic goes here                                                  */
/*  Execute custom code that will return value for the optimized expression   */
/******************************************************************************/

return (1);                                  -- success, no internal exception
exception
  when others then
   xcErrorMessage := 'CZ_USER_SELECTION.User_Optimize_Expression(' ||
                          to_char( lnStatement ) || '):' || SQLCODE || ':' ||
                         substr( SQLERRM, 1, 60 );
    return (0);                               -- failure, internal exception
end User_Optimization_Expression;



/******************************************************************************/
/* Function:     User_Clause_Function                                         */
/*                      User extensible function to allow complex clause      */
/*                      execution                                             */
/*                                                                            */
/* Parameters In:  pnLineID                                                   */
/*              The sales order line that is being validated.  This           */
/*              always points to the top model that has been ordered.         */
/*                                                                            */
/*                pnAssignmentD                                               */
/*               Unique identifier of constraint assignment that is being     */
```

```
/*                evaluated                                                  */
/*                                                                           */
/*                 pnGroup ID                                                */
/*              Used to identify prior autoselections that have occured      */
/*                in this run of validation                                  */
/*                                                                           */
/*        Out: xnReturnedValue                                               */
/*                Value of clause evaluation 0 clause evaluates to FALSE     */
/*                                           1 clause evaluates to TRUE      */
/*                                           2 clause evaluates to STOP      */
/*                                                                           */
/*                 xcErrorMessage                                            */
/*               Message string which is displayed to user                   */
/*                                                                           */
/* Returns:        Integer Value ( 0 – Failure, 1 – Success)                 */
/*                                                                           */
/*****************************************************************************/
function User_Clause_Function(       pnLineID          in      number,
                                     pnAssignmentID  in      number,
                                            pnGroupID         in      number,
                                            xnReturnedValue  out     number,
                                            xcErrorMessage    out    varchar2 )
         return integer is
lnStatement  Number := 0;                    -- local variables
begin
  lnStatement := 10;
/*****************************************************************************/
/*  Function Logic goes here                                                 */
/*  Execute custom code that will return TRUE, FALSE or STOP for the         */
/*  clause being evaluated                                                   */
/*****************************************************************************/

if (Success) then
    xnReturnedValue := 1;
elsif (Stop)                                 -- stop evaluation of constraint
    xnReturnedValue := 2;
else (Failure)                               -- default
    xnReturnedValue := 0;
end if;
*/  return (1);                     -- success,  no internal exception
exception
  when others then
    xcErrorMessage :=   'CZ_USER_SELECTION.User_Clause_Function('||
                              to_char( lnStatement ) || '):' || SQLCODE || ':' ||
```

```
                                substr( SQLERRM, 1, 60 );
    return (0);                                 -- failure, internal exception
end User_Clause_Function;
end CZ_User_Autoselection;
/
exit;
```

PL SQL Script: Sample Package Body

```
whenever sqlerror exit failure rollback;
create or replace package cz_user_autoselection as
/*============================================================================*/
/* $Header: CZUASLTB.pls 110.2 97/07/15 17:52:53 appldev ship $               */
/*============================================================================*/
/* Copyright (c) 1997 Oracle Corporation Redwood Shores, California,USA       */
/*  All rights reserved.                                                      */
/*  Oracle Manufacturing                                                      */
/*============================================================================*/
/*                                                                            */
/* Program:     CZUASLTB.pls                                                  */
/* Description: This file contains the prototypes for the following function  */
/*                                                                            */
/*   1. User_AutoSelection                                                    */
/*  This serves as an extension to the built in capabilities                  */
/*   of Oracle Product Configurator.                                          */
/*                                                                            */
/*  The User AutoSelection function is enabled by selecting the               */
/*   User Function Based Items and Quantities AutoSelection                   */
/*   Option from the Assign Constraints form. The Name of the                 */
/*   PL/SQL Package is entered in the Function field of the                   */
/*   AutoSelection Items window.  The Product Configurator engine             */
/*   calls the named package and validates your items and                     */
/*   quantities against the orderable bill of material.  Items                */
/*   must exist uniquely and the quantity must be permitted for               */
/*   that item in the orderable bill of material.                             */
/*                                                                            */
/*   2. User_Optimize_Expression                                              */
/*   A packaged function that evaluates a complex optimization                */
/*   expression that is external to the Product Configurator,                 */
/*   and returns the value to the Product Configurator                        */
/*   to be used for the range based autoselection/autoexclusion               */
/*   types Range Based Items and Quantities, Optimized Quantities             */
/*   and Optimized Items and Quantities.                                      */
/*                                                                            */
/*   The User_Optimize_Expression function is enabled by checking             */
/*   the User Function check box on the AutoSelection Items window            */
/*   and entering the Name of the PL/SQL Package in the                       */
/*   Optimization Expression field of the same window.                        */
/*   The Product Configurator engine calls the named package and              */
```

```
/*   returns a character or numeric value to be used for                    */
/*   evaluation with the defined ranges of the specified                    */
/*   AutoSelection/AutoExclusion Type.                                      */
/*                                                                          */
/*   3. User_Clause_Function                                                */
/*   A packaged function that evaluates and returns a value of              */
/*   true/false/stop to the Product Configurator to be used by              */
/*   the evaluating class.                                                  */
/*                                                                          */
/*   The User_Clause_Function is invoked by entering name of the            */
/*   package.function in the User Validate Function column of the           */
/*   constraints form.  The function column in the Defined                  */
/*   Constraint form should be 'User Validate' and the operator             */
/*   should be 'NA'.                                                        */
/*                                                                          */
/*   Context Information:                                                   */
/*   The previous autoselections that have occured for the sales            */
/*   order line in this run of Configuration validation, are                */
/*   viewable through the view CZ_COMPONENT_SELECTIONS_V,                   */
/*   filtered based on the group_id which is passed in to the               */
/*   User_AutoSelection function. Item_svrid column contains the            */
/*   constraint assignment_id that has autoselected the item,               */
/*   while those with NULL value represent customer selections              */
/*                                                                          */
/*   Called By:   CZ_VALIDATE_CONFIGURATION (CZVCFGRS/B.pls)                */
/*                                                                          */
/* Changed by:  Who                  When      What                         */
/*          Tony Gaughan    9/17/97   Updated layout (standards)            */
/*                                                                          */
/****************************************************************************/
function User_AutoSelections(   pnLineID           in     number,
                              pnAssignmentID      in     number,
                              pnGroupID           in     number,
                              xcErrorMessage      out    varchar2 )
        return integer;
function User_Optimization_Expression(   pnLineID            in     NUMBER,
                                       pnAssignmentID       in     NUMBER,
                                       pnGroupID            in     NUMBER,
                                     xcExpressionValue      out   VARCHAR2,
                                     xcDataType             out   VARCHAR2,
                                     xcErrorMessage         out   VARCHAR2 )
        return integer;
function User_Clause_Function(  pnLineID             in     number,
                              pnAssignmentID        in     number,
```

```
                                        pnGroupID           in      number,
                                        xnReturnedValue     out   number,
                                        xcErrorMessage      out   varchar2 )
        return integer;
end CZ_User_Autoselection;
/
exit;
```

CHAPTER

4 Reports

This chapter describes Product Configurator reports. Each section includes a description of submission parameters..

Configuration Messages Report

Use the Configuration Messages Report to list open orders with configuration messages that include Overidden Error messages. The Configuration Messages Report will list selected sales order header data, the sales order line model and options.

Report Submission

In the Submit Requests window, select Configuration Messages Report in the Name field.

Report Parameters

Order Type

To restrict the report to a certain type of items, enter the order type.

Order Number

To restrict the report to a specific order, enter the sales order number. If you enter a sales order number, the following fields are unavailable: Booked Only, Customer Name or ID and Sales Rep Name or ID).

Line Number

To restrict the report to a certain line, enter a line number.

Date Range From/To

To restrict the report to a certain date range, enter a beginning and ending date.

Booked Only

Select one of the following options:

No	Display orders that have yet to be booked.
Yes	Only display orders that have been booked.

On Configuration Hold

Select one of the following options:

No — Do not display orders that are on configuration hold.

Yes — Include orders that are on configuration hold in the report.

Note: This field must be yes if you entered No in the Booked Only field.

Customer Name or ID

To restrict the report to the orders of a certain customer, enter a customer name or ID.

Sales Rep Name or ID

To restrict the report to the orders taken by a certain sales rep, enter a sales rep name or ID.

Warehouse

To restrict the report to the orders located in a certain warehouse, enter a warehouse.

Model

To restrict the report to certain model type, enter a model.

See Also

Submitting a Request, *Oracle Applications User's Guide*

AutoSelection / AutoExclusion Item and Attribute Where Used Reports

Use the AutoSelection / AutoExclusion Item and Attribute Where Used reports to list constraint assignments where a specific AutoSelection / AutoExclusion Item or Attribute is used in the constraint assignment.

Auto Selction Items and/or Attibutes are used in the following constraint types:

- Standard items and quantities
- Range based items and quantities
- Optimized quantities
- Optimized Items and quantities
- User function based items and quantities

Auto Exclusion items and attributes are used in the following constraint types:

- Standard items
- Range based items
- Optimized items
- User function based items

Report Submission

In the Submit Requests window, select AutoSelection / AutoExclusion Item and Attribute Where Used Report in the Name field.

Report Parameters

AutoSelection / AutoExclusion Item

To restrict the report to a certain item, enter the item.

Attribute

To restrict the report to a specific attribute, enter the attribute.

Note: AutoSelection Item and Attribute are mutually exclusive and mandatory parameters. One of the two parameters must be entered.

Effective Date

To restrict the report to items occurring after a certain date, enter an effective date.

See Also

Submitting a Request, *Oracle Applications User's Guide*

Attribute Where Used Reports

Use the Attribute Where Used Reports to list constraint clauses where a specific attribute is used in the constraint definition.

Report Submission

In the Submit Requests window, select Attribute Where Used Report in the Name field.

Report Parameters

Attribute (Required) **ZOOM PICK HELP**

To restrict the report to a specific attribute, enter the attribute.

> **Note:** AutoSelection Item and Attribute are mutually exclusive and mandatory parameters. One of the two parameters must be entered.

Effective Date (Optional, Flexfield) **ZOOM HELP**

To restrict the report to items occurring after a certain date, enter an effective date.

See Also

Submitting a Request, *Oracle Applications User's Guide*

APPENDIX

A

Windows and Navigator Paths

This appendix shows you the default navigator path for each Product Configurator window. Refer to this appendix when you do not already know the navigator path for a window you want to use.

Product Configurator Windows and Navigator Paths

For windows described in other manuals:

See...	Refer to this manual for a complete window description.
User	*Oracle Applications User's Guide*

Text in brackets ([ ]) indicates a button.

Product Configurator Responsibility	
Window Name	**Navigation Path**
Assign Constraints: page 2 – 23	Constraints > Constraint Assignments
Attributes	Attributes > Attributes
Attribute Values	Attributes > Attribute Values
Attribute Value Lists: page 2 – 5	Attributes > Attribute Value Lists
AutoSelection / AutoExclusion Items: page 2 – 26	Constraints > Constraint Assignments > [Items]
Configurator: page 3 – 2	Switch to OE responsibility. Orders, Returns > Sales Orders > [Configurator]
Configurator Attribute Values	Attributes > Item Attribute Values
Configurator Message: page 3 – 12	Configurator > Configurator Messages > [Messages]
Configurator System Types (See **User**)	Configurator > Configurator System Types
Constraint Clause: page 3 – 12	Configurator > Configurator Messages > [Messages] > [Open]
Constraints: page 2 – 20	Constraints > Constraints
Messages Summary: page 3 – 12	Configurator > Configurator Messages
Regions	Attributes > Attribute Regions

Glossary

assemble–to–order (ATO) An environment where you open a final assembly order to assemble items that customers order. Assemble–to–order is also an item attribute that you can apply to standard, model, and option class items.

bill of material A list of component items associated with a parent item and information about how each item relates to the parent item. Oracle Manufacturing supports standard, model, option class, and planning bills. The item information on a bill depends on the item type and bill type. The most common type of bill is a standard bill of material. A standard bill of material lists the components associated with a product or subassembly. It specifies the required quantity for each component plus other information to control work in process, material planning, and other Oracle Manufacturing functions. Also known as **product structures.**

BOM item type An item classification that determines the items you can use as components in a bill of material. BOM Item types include standard, model, option class, and planning items.

component item An item associated with a parent item on a bill of material.

configuration A product a customer orders by choosing a base model and a list of options. It can be shipped as individual pieces as a set (kit) or as an assembly (configuration item).

configuration bill of material The bill of material for a configuration item.

configuration item The item that corresponds to a base model and a specific list of options. Bills of Material creates a configuration item for assemble–to–order models.

configurator A form that allows you to choose options available for a particular model, thus defining a particular configuration for the model.

configure–to–order An environment where you enter customer orders by choosing a base model and then selecting options from a list of choices.

engineering change order (ECO) A record of revisions to one or more items usually released by engineering.

model bill of material A bill of material for a model item. A model bill lists option classes and options available when you place an order for the model item.

model item An item whose bill of material lists options and option classes available when you place an order for the model item.

option An optional item component in an option class or model bill of material.

option class bill of material A bill of material for an option class item that contains a list of related options.

option class item An item whose bill of material contains a list of related options.

pick–to–order A configure–to–order environment where the options and included items in a model appear on pick slips and order pickers gather the options when they ship the order. Alternative to manufacturing the parent item on a work order and then shipping it. Pick–to–order is also an item attribute that you can apply to standard, model, and option class items.

standard bill of material A bill of material for a standard item, such as a manufactured product or assembly.

standard item Any item that can have a bill or be a component on a bill except planning items, option classes, or models. Standard items include purchased items, subassemblies, and finished products.

Index

R

S

V

W

Reader's Comment Form

Product Configurator User's Guide
A57334–01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information we use 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? What did you like least about it?

If you find any errors or have any other suggestions for improvement, please indicate the topic, chapter, and page number below:

_____

_____

_____

_____

_____

_____

_____

_____

Please send your comments to:

Oracle Applications Documentation Manager
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065 USA
Phone: (415) 506–7000 Fax: (415) 506–7200

If you would like a reply, please give your name, address, and telephone number below:

_____

_____

_____

Thank you for helping us improve our documentation.