



When Execution Matters

White Paper

## Unleash The Power Of Oracle Configurator

Sunilkumar.s  
October 2005



## Table of Contents

1	Executive Summary -----	3
2	Introduction -----	3
3	What is Configurator?-----	3
4	An Evolving Technology -----	3
5	CASE STUDY-----	4
	5.1 Key Business Challenges -----	5
	5.2 Configurator Solution -----	5
	5.2.1 The Power of Usages: -----	5
	5.2.2 The Power of Extensions: -----	6
	5.3. Capturing Non-BOM Attributes/Multiple Instantiable Model -----	9
	5.4 Maintenance of BOMs: -----	10
	5.5 Calendar Feature: -----	11
	5.6 Key Findings :------	11
6	Summary:-----	11

# 1 Executive Summary

Oracle's E-Business Suite has revolutionized the way, complex configuration rules are written to fulfill the customer needs through Oracle Configurator. For many years, Oracle Applications has provided a VB-based configurator developer for creating configuration rules and java based rules through functional companions but this functionality is limited. With the latest release, 11.5.10, more complex rules can be written thru Constraint Definition Language (CDL) and more options are available for running the Configurator Extensions, popularly referred to as CX (which replaces functional companions).

# 2 Introduction

This white paper will provide a brief overview of Configurator and how it has evolved and then describe its power to handle common and uncommon challenges. It is aimed at people who have had some experience in implementing Configurator and are familiar with maintaining items, Bills of Material (BOMs) and non-BOMs. It also discusses some of the best-practices guidelines and dos and don'ts

# 3 What is Configurator?

As the name suggests, Configurator takes a series of customer requirements or selections and, using product structures and rules, creates a unique and valid configurable solution for the customer. Customers can produce quotes for complex products and service offerings based on product rules, prices, ATP information, customer profiles and agreements. However, leveraging the existing Oracle applications data is key to successfully implement Oracle Configurator. Oracle Configurator offers tight integration to BOM and OM, ease of rule development, the ability to maintain those rules in one place across the enterprise and the ability to handle numerous and complex attributes. Configurator is a powerful addition to ERP and CRM.

# 4 An Evolving Technology

Configurator has evolved over period of years; many new functionalities have been included in releases, up to 11.5.10, the most current version. The following is a review of new-feature development.

- 11.5.5 New Features:
  - Referencing
  - Effectivity
  - Publishing
  - Dynamic UI (Hidden Features & Options)

- 11.5.6 New Features:
  - Multiple Language Support
  - UI Look & Feel Enhancements
  - Name/Desc/Name&Desc Captions
  - Multi-Segment Item Names
- 11.5.7 New Features:
  - Option Sorting
  - Text Links
  - Wizard-Style Navigation
  - BOM Synchronization
  - Advanced Math Expressions
- 11.5.8 New Features:
  - Solution-Based Modeling (PT Multi-Instantiation)
  - Connectors
  - Configurator Attributes
- 11.5.9 New Features:
  - Telco Service Upgrades
  - Dynamic UI (Hidden & Read-only Controls)
  - Solution-Based Modeling (AT Multi-Instantiation)
- 11.5.10 New Features:
  - Web-Based Developer
  - Model Debugger / Generic UI
  - Statement Rules
  - UI Templates & Editing Enhancements
  - Decimal Quantities
  - Contracts Integration
  - Configurator Extensions

## 5 CASE STUDY

This case study focuses on an engagement with a leading provider of network neutral colocation, interconnection and managed services. Through its 17 Internet Business Exchange centers (IBX) centers, in the United States and Asia, customers can directly interconnect with each other for critical traffic exchange requirements.

## 5.1 Key Business Challenges

- Evolve different business processes for different business users
- Optimize product sales through Configurator
- Reduce turnaround time by creating Identical Configured instances with the non-BOM installation details
- The installation details must be available across multiple module OM, Quote and IB
- Scalability for new product offerings
- Maintenance of BOMs
- Create a cross-browser functional calendar with date and time validations

## 5.2 Configurator Solution

### 5.2.1 The Power of Usages:

The client has two types of user:

i) Sales Reps

ii) Sales Engineers

Sales Reps create quotes and get them approved. When a quote is created, the model is selected and configured.

The model can have BOM and non-BOM Items. Since most of the non-BOM details are required items, it makes sense to exclude them when Sales Reps create quotes because they are more concerned with the BOM structure (as BOM items are only priced) than the other, non-BOM details which aid in installation. This can be achieved by usages by setting the CZ:Publication Usage property = "QOT" for the given user (Sales Rep)

Profile Option Name		Application	Responsibility	User
CZ: Publication Lookup Mode	Production		EQX - Quoting - User	
CZ: Publication Usage	Any Usage		QOT	...

and also in the parent model structure because the reference non-BOM model usage needs to be excluded.

Sales Engineers:

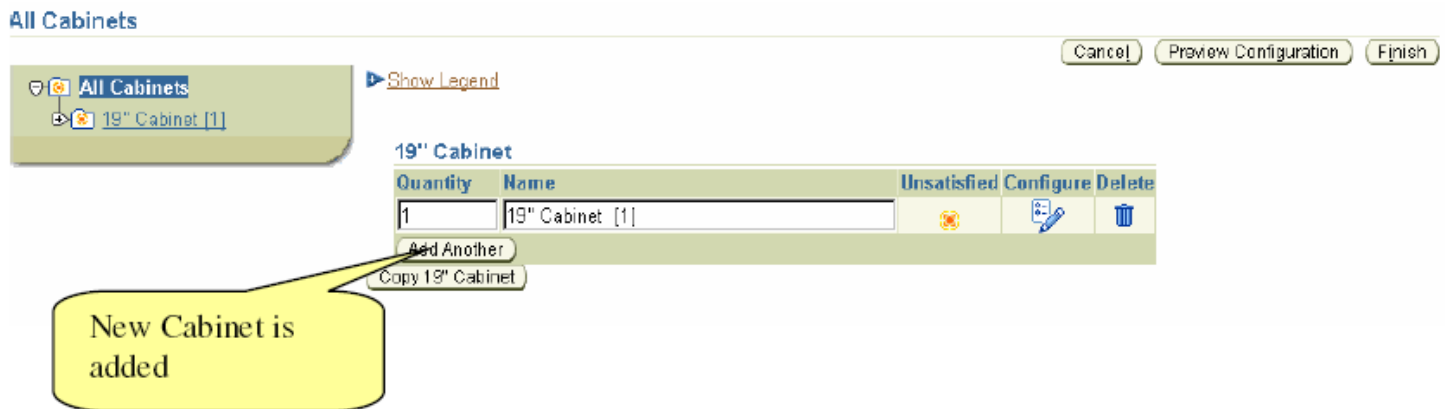
Once a quote is created and approved internally, the BOM needs to be configured along with all the required installation details, which can be sent to customer for approval.

## 5.2.2 The Power of Extensions:

### 1. Copy Configuration Instances

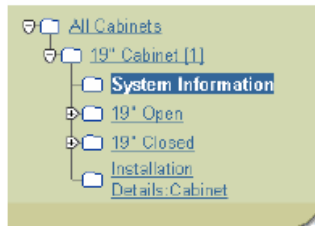
The client required that we create identical configuration instances on the fly by excluding some of the fields during copy. We needed to configure 100 cabinets and each configuration was identical except for the cabinet number, which is unique, but Configurator has a limitation; we can create a new empty instance but cannot copy from instance to another. With Configurator extensions, we can copy configurations on the fly within the same session by excluding certain items to copy, based on properties. This CX would create a new empty instance, copy all selections and values from the first configured instance and exclude items from copy which had a property called "COPY\_TEXT = No". This method significantly reduces the time required to configure large models and also provides the flexibility to change the configuration once it is copied. In this case, we will select a cabinet number from Lov (another extension which gets data from Database), once it is copied. We can see how Configurator limitations can lead to custom extensions.

#### STEP I) Add a New Instance



## STEP II) Configuring the Instance

### System Information



Show Legend

Cancel Preview Configuration Finish

#### System Information

System Name CH1:00400:UUNET

Cabinet 0202

Cabinet# Clear

Sub Cabinet #

Customer Cab Ref 1234

#### Comments

Comments

#### SR Details

SR #

SR Status

SR Type

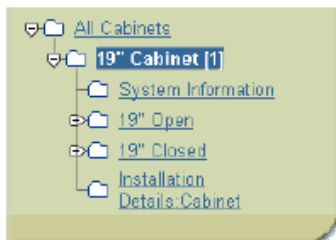
SR Owner

Cabinet value is user-entered

Customer Cab Ref is user-entered

Cancel Preview Configuration Finish

### 19" Cabinet [1]



Show Legend

Cancel Preview Configuration Finish

Status Selected Quantity 1

Select	Status	Quantity	Description	Unsatisfied	Configure
<input type="checkbox"/>		0	19" Open		
<input type="checkbox"/>		0	19" Closed		
<input type="checkbox"/>		0	Installation for 19" Cabinet equivalent		
<input checked="" type="checkbox"/>		1	Installation for Half-Cabinet		

Configure System Information

Configure Installation Details:Cabinet

User selections

Cancel Preview Configuration Finish

STEP III) Copy Configured Instances

All Cabinets

All Cabinets

19" Cabinet [1]

Show Legend

Cancel

Preview Configuration

Finish

19" Cabinet

Quantity	Name	Unsatisfied	Configure	Delete
1	19" Cabinet [1]			

Add Another

Copy 19" Cabinet

Click COPY 19" Cabinet Button

All Cabinets

All Cabinets

19" Cabinet [1]

19" Cabinet [2]

Show Legend

Cancel

Preview Configuration

Finish

19" Cabinet

Quantity	Name	Unsatisfied	Configure	Delete
1	19" Cabinet [1]			
1	19" Cabinet [2]			

Add Another

Copy 19" Cabinet

Unsatisfied mark; some required item values are not copied because COPY\_TEXT property is set to NO

19" Cabinet [2]

All Cabinets

19" Cabinet [1]

19" Cabinet [2]

Show Legend

Cancel

Preview Configuration

Finish

Status Selected Quantity 1

Select	Status	Quantity	Description	Unsatisfied	Configure
<input type="checkbox"/>		0	19" Open		
<input type="checkbox"/>		0	19" Closed		
<input type="checkbox"/>		0	Installation for 19" Cabinet equivalent		
<input checked="" type="checkbox"/>		1	Installation for Half-Cabinet		

Configure System Information

Configure Installation Details:Cabinet

Logical exclusion

User selection copied as well

Cancel

Preview Configuration

Finish



## Cabinet Number is not copied as COPY\_TEXT = NO Property is set

The screenshot displays a software window with two tabs: "System Information" and "SR Details". The "System Information" tab is active, showing fields for System Name (CH1:00400:UUNET), Cabinet # (with a "Cabinet # Clear" button), Sub Cabinet #, and Customer Cab Ref (1234). A yellow callout points to the Cabinet # field, stating "Cabinet Number user-entered value Not copied". Another yellow callout points to the Customer Cab Ref field, stating "Customer Cab Ref user-entered value". The "SR Details" tab shows fields for SR #, SR Status, SR Type, and SR Owner. A purple callout points to the Comment field, stating "Formatted: Font: 10 pt". Buttons for "Cancel", "Preview Configuration", and "Finish" are visible at the top and bottom of the window.

## 5.3. Capturing Non-BOM Attributes/Multiple Instantiable Model

### Attributes:

Traditionally, Attributes are captured only from BOM nodes. Oracle provides "WriteAttributes" FC which can capture attributes for BOM nodes, but there can be cases when it is necessary to capture the installation details of multiple instantiated non-BOM models.

The client had a requirement to render different services to their customers and they wanted to create only one order for all the maintenance services they render. Additionally, those services needed to be classified by a ".Item" in the model structure. Services could include:

- i) Customer visiting client site
- ii) Giving access to registered users
- iii) Removing access to registered users

and many more . At any point for a given SERVICE.MOD line in a sales order, only one service can be rendered. Details of service to be rendered are entered in the corresponding Non-BOMs; in this case it is required to capture details of Non-BOMs (e.g. "Work Visits"), along with its multiple instantiated childs (Visitor).

Sample Model Structure:

SERVICE.MOD

WORK\_VISITS.ITEM (BOM STD ITEM)

HANDSCANADD.ITEM (BOM STD ITEM)

HANDSCANREMOVE.ITEM (BOM STD ITEM)

Installation Details: Work Visits

|\_\_ Vistors (1/\*)

Installation Details: Handscan Add

|\_\_ Vistors (1/\*)

Installation Details: Handscan Remove

|\_\_ Vistors (1/\*)

i) This is achieved by CX , which checks when any .ITEM is selected; the corresponding NonBomModel “Install Details “ instance is created. When the configuration is saved, another CX,

WriteNonBomAttributes is invoked. This captures the attributes of the selected non-BOM and its child reference which is multiple instantiable, i.e. the “Visitors” model.

ii) The multiple-instantiated “Visitors” models are distinguished in the CZ\_Config\_Attributes table by the attribute category. The Instance Name is stored as the attribute category.

Here, we see how CX is used to drive complex business processes

## 5.4 Maintenance of BOMs:

i) The client has 17 different orgs and each org has a different BOM structure for each service they render. There are 20 different BOMs in total.

ii) If an individual BOM structure is created for each org, 340 (20 multiplied by 17) BOMs need to be created and maintained, which is a cumbersome process.

iii) To avoid creating replicating BOMS across orgs, we created one BOM for each service; this BOM would also contain all org-specific items and, depending upon the org it is being invoked from, all the other orgspecific elements are not shown to the users; this is done thru CX

iv) On Load of Model, two CX are invoked

a. The first CX will load all orgs in the selected box

b. The second CX will check the inventory organization from which the model is invoked. Depending on which model is invoked, it sets the selected value in the option box. This triggers the On Change of select box, which invokes exclusion rules, excluding items from the BOM that are not relevant to the org..

## 5.5 Calendar Feature:

- i) A cross-browser compatible calendar with the same look and feel as an OA calendar
- ii) Ability to do different date and time validations
- iii) Can be attached as a CX to any text node.

## 5.6 Key Findings :

BOM-structure planning is significant

- Items Versus Alternatives to Items (attributes)
- Architecture challenges with modeling
- Developing different business process for different users based on usages
- Reduce Configuration time significantly due to copy functionality, which is generic, to be attached to any models

## 6. Summary:

With the power of extensions and usages, we can drive complex business process and make them look simple. However, because Oracle Configurator is seamlessly integrated with BOM and OM Quote, leveraging the existing Oracle applications data is key to a successful implementation of Oracle Configurator, from Prospect to Order booking.

## About Trianz

Trianz is a management consulting, technology, engineering, and outsourcing services firm that helps senior leaders execute business and technology initiatives. A unique mix of business consulting and technology capabilities has helped Trianz rapidly grow since its inception in 2000. Trianz provides services to a diversified global client base. Clients are result-focused leaders employed in businesses ranging from Fortune 1000 corporations to emerging rapid-growth companies. Service offerings focus on the following areas:

Operations Consulting Services

Enterprise Applications Services

Software Product Engineering Services

Transformational Outsourcing Services

Disclaimer for white papers

©2006 Trianz. All rights reserved

*Copyright in whole and in part of this document belongs to Trianz, Inc. This work has been provided for informational purposes only, and may be copied for personal use only. This work may not be used, sold, transferred, adapted, abridged, copied, or reproduced in whole or in part, in any media, by enterprises, without the prior written permission of Trianz, AND an acknowledgement of "Trianz" as the source of the content. All trademarks and copyrights mentioned in this white paper are the property of their respective owners. Neither the author nor Trianz bears any responsibility for damage resulting from the use of the information contained herein.*

*For more information about Trianz and its capabilities, visit '[www.trianz.com](http://www.trianz.com)'*