

R12.2.9 Oracle Implement Configurator

Oracle E-Business Suite

DURATION

5 Days

MODULES

30 Lectures

COURSE CODE

—

Course Overview

Experience the runtime Oracle Configurator operating within the Oracle Order Management host application. Learn how to set up Items and Item Catalog Categories in Oracle Inventory, construct BOM Models in Oracle Bills of Material, and then import the data into Oracle Configurator Developer (OCD). Work in Oracle Configurator Developer (OCD) to build configuration model structure and rules.

What You Will Learn

About This Course

- Objectives
- Course Goals
- Audience
- Assumptions
- Prerequisites
- Available Course Modules
- Key Course Topics
- Using This Course Most Effectively
- Supplement This Course with Self-Paced Online Learning
- Available Online Classes
- Terminology
- Quiz
- Summary
- For More Information

Implementation Overview

- Objectives
- High-level Implementation Flow from Design to Maintenance
- Optimizing Configuration Model Design
- Implementation Flow for Creating a Configuration Model

- Implementation Flow: Prepare and Import Steps
- Implementation Flow: Add Non-orderable Data
- Implementation Flow: Develop and Test Rules
- Implementation Flow: Develop and Test User Interfaces
- Implementation Flow: Publish and Deploy
- Deploying Oracle Configurator
- Manage Configurations and Configuration Models
- Maintaining Oracle Configurator
- Implementation Tools
- Programmatic Tools
- Quiz
- Summary

Configuration Models Overview

- Objectives
- Definition of a Configuration Model
- Key Parts of a Configuration Model
- Model Structure
- Rules
- User Interface Definitions
- Extensions
- Relationship of Model Structure to BOM Models
- Configuration Model At Runtime
- Quiz
- Summary

Configurations Overview

- Objectives
- Definition of a Configuration
- Creating Configurations
- Configuration Identity
- Saved Configurations
- Saved Configurations that are Restored, Deleted, or Copied
- Relationship of Configurations to Configuration Models and BOM Models
- Effect of Configuration Model Changes on Restored Configurations
- Managing Configurations as Order Lines
- Relationship of Configurations to Host Applications
- Configuring an Item in Oracle Order Management
- Booking the Order
- Quiz
- Summary

Getting Started with Oracle Configurator Developer, Part 1

- Objectives

- Oracle Configurator Developer
- Setting Preferences
- Searching for an Object
- Setting the Focus on a Specific Object
- Locator Links
- Creating a Folder
- Summary

Getting Started with Oracle Configurator Developer, Part 2

- Objectives
- Properties
- Moving and Copying Objects
- Locking
- Force Unlock
- Automatic Locking
- Viewing Locked Status
- Opening a Model for Editing
- Quiz
- Summary

About Configurator Data, Part 1

- Objectives
- Configurator Data and the CZ Schema
- Accessing Data in the CZ Schema
- Populating the CZ Schema
- CZ Schema Supported Processes
- Relationship of Development and Production Data
- Development and Production Processes
- Model Migration
- Remote and Local Servers
- Quiz
- Summary

About Configurator Data, Part 2

- Objectives
- The CZ Schema's Item Master
- Item Master Data
- Imported Item Master Data
- Item Master Area of the Repository
- Orderable Items
- Quiz
- Summary

Preparing Inventory Data

- Implementation Flow: Prepare Data
- Objectives
- Why Prepare Data for Import?
- Types of Imported Data
- Oracle Inventory Items
- Inventory Organizations
- Item Attributes
- Inventory and Order Management Attribute Settings for Configurable
- Inventory Items
- Bills of Material Attribute Settings for Configurable Inventory Items
- Models that Support Reconfiguration of Installed Configurations
- Task: Creating an Item in Oracle Inventory
- Oracle Inventory Item Catalogs
- Item Catalog Descriptive Elements
- Imported Oracle Inventory Item Data Mapped in Configurator Developer
- Quiz
- Summary

Preparing BOM Model and Rule Data

- Objectives
- Oracle Bills of Material (BOM) Models
- BOM Model Structure
- Configurable BOM Model Components
- Bills of Material in Inventory Organizations
- BOM Item Types
- BOM Models That Can Be Configured
- Option Classes
- Implicit Rules
- Option Class Selection Rules
- Bill of Material Example
- Preparing a BOM Model for Import
- Imported Rules
- Preparing Rules for Import
- Populating CZ_IMP_RULES
- Populating CZ_IMP_LOCALIZED_TEXTS
- Review of Implementation Flow: Preparing Data
- Quiz
- Summary

Importing Data

- Implementation Flow: Import
- Objectives
- Types of Data to Import

- Importing BOM Model Data Into the CZ Schema
- Standard Import Process
- Standard Import From 10.7, 11.0, 11i or 12 to CZ Schema
- Define and Enable the Import Source as a Remote Server
- Define and Enable Remote Server Concurrent Programs
- CZ_DB_SETTING Parameters
- Controlling the Import
- Using the Populate Configuration Models Concurrent Program
- Import Configuration Rules: Concurrent Program
- Using the Import Configuration Rules: Concurrent Program
- Validation of Rule Import Data
- Validation of Rule's CDL Syntax
- Importing Legacy Data
- Legacy Data
- Custom Import to CZ Schema
- Review Implementation Flow: Importing Data Steps on Import Target
- Quiz
- Summary

Verifying Data

- Implementation Flow: Prepare, Import, and Verify Data
- Objectives
- Verifying Imported Data
- Locating Imported BOM Models
- Locating Referenced BOM Models in Configurator Developer
- Refresh a Single Configuration Model Concurrent Program
- Refreshing Copies of BOM Models
- Refresh All Imported Configuration Models Concurrent Program
- Disable/Enable Refresh of a Configuration Model Concurrent Program
- Verifying a Refreshed BOM Model
- Verifying BOM Model Changes in a Refreshed Configuration Model
- Verifying Imported Rules
- Reimporting Rules
- Verifying Reimported Rules
- Quiz
- Summary

About Configuration Models

- Objectives
- Model Structure Imported from Oracle Applications
- Imported BOM Nodes
- Characteristics of Imported BOM Models
- Models Created in Configurator Developer (non-BOM Models)
- Working with non-BOM Models in Oracle Configurator Developer
- Model Details

- Model Structure and Instantiability
- What is Multiple Instantiation?
- Models Structure and Instantiation at Runtime
- Nodes That Are Instantiable
- Nodes That Are Not Instantiable
- Supporting Instantiability: Host Applications
- Quiz
- Summary

About Properties

- Objectives
- Properties
- Property Types
- Managing User Properties: Main area of the Repository
- Managing User Properties: Node Details Page
- Managing User Properties: Item Master area of the Workbench
- Modifying Properties: Summary
- Inherited Properties
- Properties Created from User Defined Attributes
- Quiz
- Summary

Creating Model Structure, Part 1

- Objectives
- Why Create Nodes in Configurator Developer?
- Creating Model Structure
- Model Structure and Parent-child Relationships
- Model Structure Node Types
- Model Structure Node Icons
- Components
- Features
- Option Features
- Options
- Numeric Features
- Numeric Features: Definition
- Boolean Features
- Text Features
- Summary

Creating Model Structure, Part 2

- Objectives
- Totals and Resources
- Totals
- Resources

- Model References
- Modifying a Non-BOM Model Reference
- Connectors
- Using Item Master Data to Create Model Structure
- Building a Model Using Populators in Configurator Developer
- Quiz
- Summary

Working with Model Structure, Part 1

- Objectives
- Reorganizing Model Structure
- Editing Model Node Details
- Reorder Children
- Quiz
- Summary

Working with Model Structure, Part 2

- Objectives
- Instantiability Settings
- Instantiability
- Changing Minimum and Maximum Instances at Runtime
- Instances and Quantity
- Example of Multiple Instantiation
- Effectivity
- Effectivity and Model Nodes
- Usages
- Effectivity Sets
- Applying Effectivity in Configurator Developer
- Effectivity Date Filter: Settings
- Effectivity Date Filter: Effects
- Effectivity Date Filter: Limitations
- Creating a Model Report
- Quiz
- Summary

Testing Model Structure

- Implementation Flow for Creating a Configuration Model
- Objectives
- Why Unit Test Model Structure?
- Model Debugger Access
- Quiz
- Summary

About Configuration Rules

- Implementation Flow for Creating a Configuration Model
- Objectives
- Overview
- Configuration Rule Types
- Oracle Configurator Logic
- Structure and Rules at Runtime
- Model Logic
- Logic Generation Warnings
- Logic Generation Errors
- Implicit BOM Rules
- Model Structure and Rules
- Model References and Rules
- Quiz
- Summary

Rule Creation

- Objectives
- Using Properties when Defining a Rule
- Rule Details: Disable Setting
- Rule Details: Violation Message
- Rule Details: Input Required Message
- Rule Sequences
- Quiz
- Summary

Logic Rules

- Objectives
- Logic Rules
- Requires Relation
- Implies Relation
- Negates Relation
- Excludes Relation
- Defaults Relation
- Logic Rule Definitions
- AllTrue and AnyTrue
- Using Model Node Properties in a Logic Rule
- Quiz
- Summary

Testing Configuration Rules

- Implementation Flow for Creating a Configuration Model
- Objectives

- Testing Rules
- Testing and Debugging Rules: Generating Logic
- Unit Testing
- Suggestions for Testing and Debugging Rules
- Quiz
- Summary

Numeric Rules

- Objectives
- Numeric Rules
- Numeric Rule Types
- Changing Minimum and Maximum Instances at Runtime
- Using Properties in a Numeric Rule: Additional Example
- Quantity Multiplier
- Integer Conversion Method
- Quiz
- Summary

Compatibility Rules

- Objectives
- Compatibility Rules
- Compatibility Rule Participants and Maximum Selections
- Explicit Compatibility Rules
- Defining an Explicit Compatibility Rule: Overview
- Explicit Compatibility Rule: Example
- Property-based Compatibility Rules
- Property-based Compatibility Rules: Example
- Property-based Compatibility Rules: Comparison Operators
- Design Charts
- Design Charts: Example
- Quiz
- Summary

Comparison Rules

- Objectives
- Comparison Rules
- Comparison Rule Operators
- Quiz
- Summary

Statement Rules

- Objectives
- Upgrading Rules
- Statement Rules

- Advantages of Statement Rules
- Statement Rule Definition
- CDL Elements
- Statement Rule Elements
- Converting Existing Rules into Statement Rules
- Statement Rule Validation
- Statement Rule: Iterators
- Iterator Statement: Using FOR ALL...IN Clause
- Iterator Statement: Using WHERE Clause
- Statement Rules: CONSUMES...FROM Equivalence
- Quiz
- Summary

Connectivity and Network Models

- Implementation Flow for Creating a Configuration Model
- Objectives
- Overview of Connectivity
- Models that Require Connectivity
- Computer Network Example
- Connectors
- Required vs. Optional Connections
- Connector Targets and Model Structure
- Connectors and Configuration Rules
- Connectors and Rules in Configurator Developer
- Connectors and Rules at Runtime
- Creating Connections at Runtime
- Creating Instances with Connectors at Runtime
- Quiz
- Summary

User Interfaces

- Objectives
- Kinds of Configurator User Interfaces
- User Interfaces Created in Configurator Developer
- Definition of Common UI Terms
- User Interface Templates
- User Interface Structure
- UI Structure and Elements in Configurator Developer
- UI Elements at Runtime
- User Interface Templates
- UI Master Template Settings
- Template-Based User Interface Creation
- UI Content Templates
- Creating a UI Template
- Implementation Flow for Creating a Configuration Model

- Generated vs. Empty User Interface
- Navigation Styles
- Step-by-Step Navigation Style at Runtime
- Dynamic Model Tree Navigation Style at Runtime
- Side Menu Navigation Style at Runtime
- Subtab Navigation Style at Runtime
- Default Icons
- Implementation Flow for Creating a Configuration Model
- Before Unit Testing a User Interface
- Testing a User Interface
- Editing a User Interface
- The UI Definition
- The Menus Folder
- Menus
- Page Links
- The Page Flows Folder
- UI Elements: Introduction
- Example of a Generated UI Element
- Layout Regions
- Understanding Layout Regions
- Example of a Stack Layout Region
- Example of a Row Layout Region
- Creating UI Elements
- Basic UI Elements
- Other UI Elements
- Types of “Button” UI Elements
- Copying, Moving, and Deleting UI Elements
- UI Captions
- Runtime Actions
- Sorting Options at Runtime
- Refreshing a User Interface
- The Refresh Enabled Setting
- Task: Refreshing a User Interface
- Quiz
- Summary

Publishing Configuration Models

- Objectives
- Implementation Flow for Creating a Configuration Model
- Publishing Configuration Models
- Publishing Flow
- Publication
- Applicability Parameters
- Applications Parameter
- Languages Parameter
- Usages and Date Range Parameters

- Publishing Concurrent Programs
- Maintaining Publications
- Publication Status
- Republishing a Configuration Model
- Quiz
- Summary
- A Configurator Extensions
- Implementation Flow for Creating a Configuration Model A-2
- Objectives A-3
- Overview of Configurator Extensions A-4
- Key Elements of a Configurator Extension Implementation A-5
- Overview of the CIO A-6
- Configurator Extensions and the CIO A-7
- Configurator Extension Invocation A-8
- Configurator Extensions and Model Structure A-9
- Multiple Instances at Runtime A-10
- CIO Interaction With the User Interface A-11
- Custom Applications and the CIO A-12
- Using the CIO Interfaces and Classes A-13
- Example of Using CIO in Java Code A-14
- Program Flow in CX Java Class Files A-15
- Recommended Practices A-16
- For More Information A-17
- Implementation Flow: Develop Configurator Extensions A-18
- Development Prerequisites A-19
- Overview of Configurator Extension Development A-20
- Task: Building Java Classes A-21
- Java Source File Example A-22
- Task: Building Java Classes A-23
- Java Source File Example Using CIO A-24
- Task: Building Java Classes A-25
- Remote Debugging with JDeveloper A-27
- Configurator Extension Requirements and Restrictions A-28
- Configurator Extension Archives – Definition A-29
- Configurator Extension Archive Path – Definition A-30
- Defining Configurator Extension Rules A-31
- Elements of a Configurator Extension Rule A-32
- Configuration Events A-33
- Types of Configuration Events A-34
- Predefined Configuration Events A-35
- Event Binding – Definition A-36
- Event Binding Behavior A-37
- Elements of an Event Binding A-38
- Model Entities for Argument Bindings A-39
- Relationship of Java Method Parameters to Event Binding Arguments A-40
- Creating an Event Binding A-41

- Relationship of Event Binding Arguments to Java Method Parameters A-42

- Command Event – Definition A-43

- Updating Logic and User Interface A-44

- Command Event Buttons A-45

- Changing Default Button Captions A-46

- Testing Command Event Buttons A-47

- Quiz A-48

- Summary A-51

- B Configurator Extension Command Events

- Objectives B-2

- Command Event – Definition B-3

- Updating Logic and User Interface B-4

- Command Event Buttons B-5

- Changing Default Button Captions B-6

- Testing Command Event Buttons B-7

- Quiz B-8

- Summary B-11

- C Course Summary

- Objectives C-2

- Configuration Models C-3

- Configurations C-4

- Implementation Tasks C-5

- CZ Schema Data C-6

- Model Structure C-7

- Configuration Rules and Implicit BOM Rules C-8

- Configuration Rules C-9

- User Interfaces C-10

- Types of User Interfaces C-11

- UI Templates C-12

- UI Elements and Settings C-13

- Editing User Interfaces C-14

- Configurator Extensions C-15

- Unit Testing C-16

- Publishing C-17

- Using Configurator Developer Effectively C-18

- Runtime Access of Configuration Models C-19

- BOM Model-based Configuration Model At Runtime C-20

- For More Information C-21