(22B) Oracle CX Sales and Oracle B2B Service: Composers & Scripting

Oracle CX Sales and B2B Service

DURATION	MODULES	COURSE CODE
3.5 Days	**21 Lectures**	**—**

Course Overview

This training applies toOracle CX Sales and Oracle B2B Service.

What You Will Learn

Course Introduction

- Learning Objectives
- Course Audience
- Course Prerequisites
- Course Goal
- Course Objectives
- Course Methodology
- Course Materials
- Course Agenda
- Resources: Documentation
- Resources: My Oracle Support
- Resources: Other Resources

Extensibility Framework

- Learning Objectives
- Oracle Advertising and Customer Experience (CX)
- Oracle CX Sales and B2B Service
- Oracle Cloud Technology
- Full Lifecycle Support
- Oracle Cloud Predefined Functionality
- User Access to Oracle Cloud Applications: User Interface (UI)
- Extending Sales and Service

- Oracle Cloud Applications
- MVC Implementation
- Terms Used in Modifying Sales and Service
- Extending Oracle Cloud Applications
- Oracle Metadata Services (MDS)
- MDS Modifications
- Modification Levels
- Example: Two Users, Two Experiences
- Composer Tools
- Development Best Practices
- Resources
- Lesson Highlights

Sandboxes

- Learning Objectives
- Technical Challenge and Solution: Configurator Conflict
- Sandboxes
- When Should You Use a Sandbox?
- Create a Sandbox
- Enter a Sandbox
- Work in a Sandbox
- Object Locking
- Publish a Sandbox
- Refresh a Sandbox
- Publishing Sandboxes
- Delete a Sandbox
- Recommended Practices and Additional Information
- Use the Metadata Manager Tool
- Object-Specific Metadata
- Lesson Highlights
- Practice

Extend Standard Objects

- Learning Objectives
- Objects
- Standard Objects
- Custom Objects
- Field
- Field Types
- System Fields
- Standard Fields
- Custom Fields
- Pages
- Application Pages
- Modifying an Existing Object

- Examples: Use Cases For Modifying an Existing Object
- Modify an Existing Object
- 1. Select and Activate a Sandbox
- 2. Navigate to the Object
- 3. Add Fields
- 4. Modify the Page Layouts
- 5. Verify the Appearance of the Object
- 6. Publish the Sandbox
- Lesson Highlights
- Practice

Create Custom Objects

- Learning Objectives
- Additional Entity Requirements
- Custom Objects
- Custom Objects: Standard Fields
- Custom Objects: Pages
- Custom Objects: Security
- Custom Objects: Available Operations
- Examples: Use Cases For Creating Custom Objects
- Create a Custom Object
- 1. Create or Activate a Sandbox
- 2. Create a Custom Object
- 3. Create Fields for the Custom Object
- 4. Create Pages for the Custom Object
- 5. Manage Custom Object Security
- 6. Review and Publish the Object
- Custom Objects and Relationships
- Lesson Highlights
- Practice

Fixed Choice Lists

- Learning Objectives
- Fixed Choice Lists
- Lookup Types
- Lookup Types: Definitions
- Lookup Types: Other Important Attributes
- Examples: Use Cases For Creating New Lookups
- Configure a Fixed Choice List Field
- 1. Create or Edit a Fixed Choice List Field
- 2. Select or Create the Lookup Type
- 3. Edit the Lookup Values
- 4. Make the Field Visible on a Page
- 5. Save, Close, and Test the Choice List
- Dependent Fixed Choice List Fields

- Value Maps
- Create a Dependent Fixed Choice List Field
- 1. Create the Parent Fixed Choice List Field
- 2. Create the Child Fixed Choice List Field, Specifying the Parent
- 3. Create a Value Map
- 4. Make the Field Visible and Test the Choice List
- Lesson Highlights
- Practice

Dynamic Choice Lists

- Learning Objectives
- Dynamic Choice List Field
- Dynamic Choice List Field: Example
- Search and Select Dialog Layouts
- Search and Select Dialog Layouts: Default Search
- Search and Select Dialog Pages: Default Search
- Examples: Use Cases For Creating Dynamic Choice Lists
- Configure a Dynamic Choice List Field
- 1. Create or Edit a Dynamic Choice List Field
- 2. Select the Related Object and Field to Display
- 3. Select Additional Options: Data Filter
- 3. Select Additional Options: Advanced Data Filter
- 4. Make the Field Visible on a Page
- 5. Save, Close, and Test the Choice List
- Objects Without Search and Select Dialog Layouts
- Lesson Highlights
- Practice

Child Objects, Related Objects, and Relationships

- Learning Objectives
- Associated Entities
- Object Relationships
- Types of Objects in Relationships
- Top-Level objects
- Child Objects
- Relationship Cardinality
- Relationship Type
- Reference Relationship
- Many-to-Many Relationship
- Parent-Child Relationship
- Choice List Relationship
- Configure a Relationship
- Examples: Use Cases For Creating Object Relationships
- Configure a Parent-Child Relationship
- 1. Navigate to the Parent Object

- 2. Create the Child Object
- 3. Add Fields to the Child Object
- 4. (Optional) Add Child Object Pages
- 5. Edit the Child Object Security
- 6. Display the Object
- 7. Verify the Results
- Joins
- Join Fields
- Display Join Fields
- Data Modeling: Considerations
- Data Modeling: Top-level versus Child Object
- Data Modeling: One-to-Many Relationship
- Data Modeling: Many-to-Many Relationship
- Lesson Highlights
- Practice

Subtabs

- Learning Objectives
- Additional Content
- Notes
- Change History
- Context Link Data
- Web Application (Mashup) Content
- Analytics Subtabs
- Display Notes or Change History
- 1. Select the Layout to Modify and Add a Subtab
- 2. Add a Notes or Change History Subtab
- 3. For Change History, Enable Auditing
- Display Context Link Data
- 1. Select the Layout to Modify and Add a Subtab
- 2. Specify the Object, Filters, and Fields
- Example: Show All Other Demonstration Assets Associated with the Same
- Owner
- Display Web Content
- 1. Create a Mashup
- 2. Select the Layout to Modify and Add a Subtab
- 2. (Alternative) Add Mashup Content to the Summary Page
- 3. Modify the URL
- 4. Test the Results
- Display Analytics
- 1. Expose an Analytics Subtab
- 2. Add Content to the Subtab
- Additional Subtab Features
- Lesson Highlights
- Practice

Dynamic Layouts

- Learning Objectives
- Dynamic Layouts
- Dynamic Layout Examples
- Dynamic Layout Types
- Summary of Features: Page Layouts
- Example: Enable Mass Update of Fields
- Example: Group User Information Fields
- Summary of Features: Subtab Layouts
- Object-Specific Dynamic Layout Types
- Example: Service Request-Specific Dynamic Layout Areas
- Record Type Fields
- Record Type Field Properties
- Configure a Dynamic Layout for a Page
- 1. Modify or Duplicate an Existing Layout
- 2. Configure the Conditions: Record Types
- 2. Configure the Conditions: Roles
- 2. Configure the Conditions: Expressions
- 3. Configure the Layout
- 4. (Optional) Reorder the Layouts
- 5. Test the Results
- Recommended Practices
- Lesson Highlights
- Practice

Scripting

- Learning Objectives
- Scripts
- Groovy
- Groovy Syntax
- Groovy Comments
- Add Groovy Variables
- Use Object Fields in Groovy
- Set Object Field Values in Groovy
- Dealing with Null Values
- Expression Builder
- Palette
- Functions Tab
- Add a Function
- Fields Tab
- Keywords Tab
- Web Services Tab
- Validate a Script
- Scripting Uses

- Scripting Recommended Practices
- Lesson Highlights

Field Level Scripting

- Learning Objectives
- Scripting Uses
- Formula Field
- Formula Field: Example
- Create a Formula Field
- 1. Create the Field
- 2. Specify the Field Dependency
- 3. Write the Script
- 4. Test the Field
- Field-Level Scripts
- Field-Level Scripts: Examples
- Example: Require Package
- Example: Make Package Read-Only
- Example: Calculate Delivery Date
- Scripting Recommended Practices for Field-Level Scripts
- Lesson Highlights
- Practice

Validation and Triggers

- Learning Objectives
- Object Level Scripting
- Validation Rule
- Trigger
- Create a Validation Rule
- 1. Select the Object
- 2. Select the Level of the Rule
- 3. Create the Error Message
- 4. Create the Rule
- 5. Test the Rule
- Create a Trigger
- 1. Select the Object
- 2. Select the Level of the Trigger
- 3. Specify the Triggering Event
- 4. Write the Trigger Script
- 5. Create the Error Message
- 6. Test the Trigger
- Recommended Practices for Optimal Performance
- Lesson Highlights
- Practice

Object and Global Scripts

- Learning Objectives
- Object Level Scripting
- Object Function
- Global Function
- Invoke a Function
- Actions and Links
- Adding Actions and Links to Pages
- Smart Actions
- Notifications
- Use Cases for Object Functions
- Create a Function
- 1. Create an object level or Global Function
- 2. Write the Script
- 3. Invoke the Function from Other Functions
- 4. Add an Action to Invoke the Function
- 5. Test the Function
- Lesson Highlights
- Practice

Accessing Other Objects in Groovy

- Learning Objectives
- Review: Use an Object's Fields in Groovy
- Access Fields of Related Objects in Groovy
- Review: Access Parent Object Fields
- Access Child Object Fields
- Example: Count the Number of Contacts Named Smith Associated with the Current Opportunity
- Access Dynamic Choice List Fields
- Access 1:M Relationships
- Many-to-Many Relationships
- Access M:M Relationships
- Code Example: Determine Whether An Account Is Assigned Any Demonstration Assets
- View Objects
- View Criteria
- Additional View Object Methods
- Code Example: Create a New Demonstration Asset Record
- Bind Variables
- Code Example: Use a Bind Variable
- Lesson Highlights
- Practice

Debugging Scripts

- Learning Objectives
- Debugging

- Run Time Messages
- Enable Runtime Messages
- Generate Runtime Messages
- Review Runtime Messages
- Exceptions
- Generate an Exception
- Generate a Warning
- Time Script Execution
- Use the Debugger
- 1. Start the Debugger
- 2. Select an Object and Script
- 3. Use the Debugger
- Lesson Highlights
- Practice

Invoking Web Services with Groovy

- Learning Objectives
- Review: Web Service
- Types of Web Services
- Differences Between RESTful and SOAP Web Services
- Consequences: Using REST
- Outbound Web Services in Oracle Application Composer
- Web Service Security: REST
- Web Service Security: SOAP
- Web Service Arguments
- Web Service Response
- Invoke a RESTful Web Service Using Groovy
- 1. Register the Web Service
- 2. Create the Request
- 3. Invoke the Web Service
- 4. Parse the Response
- 5. Test the Results
- Scripting Recommended Practices for Optimal Performance
- Lesson Highlights

Object Workflows

- Learning Objectives
- Object Workflows
- Conditions
- The Object Workflow's Object
- The Object Workflow's Event Point
- The Object Workflow's Event Condition
- Event Actions
- Common Action Settings
- Update Fields

- Create a Task
- Create a Task: Example
- Send an Outbound Message
- Email Notification
- Email Templates
- Email Templates: Example
- Invoke a Business Process Flow
- Invoke a Groovy Script
- Create an Object Workflow
- 1. Exit the Sandbox to Create and Necessary Objects, then Re-enter the Sandbox
- 2. Create the Workflow and Set the Event Point
- 3. Specify the Event Condition
- 4. Create One or More Actions for the Workflow
- 5. Test the Workflow
- Recommended Practices for Workflow Performance
- Lesson Highlights
- Practices

Business Process Composer

- Learning Objectives
- Oracle Business Process Composer (Process Composer)
- Process Composer: Features
- Deploying Business Processes
- An Invoking Workflow
- Approval Parameters
- Approval Page
- Create a Business Process
- 1. Exit the Sandbox
- 2. Create the Business Process Flow
- 3. Edit the Business Process Flow
- 4. Deploy the Business Process Flow
- 5. Create a Workflow
- 6. Test the Results
- Lesson Highlights

Tailoring the User Interface

- Learning Objectives
- Tailoring the User Interface
- Page Composer
- Page Composer: Support Context
- The Add Content Tab
- The Select Tab
- Use Page Composer
- Create Legacy Work Area Saved Searches
- Modify Dashboards

- Modify Components of a Page
- Should I use Application Composer or Page Composer?
- Page Composer and Application Composer
- The Structure Page
- Edit a Functional Area
- Determine Available Infolet Pages
- Personalization
- Personalize the Springboard
- Themes
- Modify a Theme
- Summary: Administrator Tools (Require an Active Sandbox)
- Summary: User Tools (Personalization, no Sandbox Required)
- Lesson Highlights
- Practices

Additional Topics

- Learning Objectives
- Technical Challenge 1: Displaying Third-Party Web Content
- Page Integration
- Security
- Category
- Structure
- Test the Results
- Additional Features
- Technical Challenge 2: Modifying User Interface Text
- User Interface Text
- Replacement Options
- Preview Changes
- Apply Changes
- Lesson Highlights
- Practices