Siebel Open UI Foundations

Oracle Siebel CRM

DURATION	MODULES	COURSE CODE
2 Days	**9 Lectures**	**—**

Course Overview

This Siebel Open UI Foundations training first introduces students to the Siebel Open UI user interface and architecture. The second part of the course introduces the concept of the Open UI "manifest" and how to administer it, then describes presentation models and physical renderers. The final lesson of the course describes the Siebel mobile interface in both connected and disconnected modes.

What You Will Learn

Course Introduction

- Lesson Agenda
- Instructor and Class Participants
- Training Site Information
- Course Audience
- Course Prerequisites
- Course Goal
- Course Objectives
- Course Methodology
- Course Materials
- Information Sources
- Course Agenda

Introducing Siebel Open UI

- Objectives
- Siebel Open UI
- Siebel Open UI Benefits
- Native Browser Support
- Mobile Device Support
- Updated Look
- Enhanced User Experience

- Responsive Web Design
- Client-Side Customizations
- Large Community of Developer Resources
- Customizing Siebel Open UI
- Change the Overall Look and Feel
- Change the Way Data is Presented
- Using JavaScript Controls
- Add Client-Side Logic
- Siebel Tools
- JavaScript
- Considerations
- Lesson Highlights
- Practice

Siebel Open UI Architecture

- Objectives
- Siebel Open UI Key Components
- Application Object Manager (AOM)
- Siebel Web Template (SWT) Files
- Decoupled Layout
- SWT Folder Structure
- Cascading Style Sheet (CSS) files
- CSS Folder Structure
- JavaScript (JS) files
- JS Folder Structure
- Browser Scripts
- Open UI Client
- Proxy
- Presentation Model (PM)
- Physical Renderer (PR)
- Plug-In Wrappers (PWs)
- Manifests
- Manifest Administration
- Example: Navigating to the My Contacts View
- The Physical Renderer or Presentation Model Passes the Request to
- the Proxy
- 2. The Proxy Sends a Request to the Siebel Server
- 3. The Siebel Server Returns the Data
- 4. The Presentation Model Applies Business Logic
- 5. The Physical Renderer Generates HTML
- 6. The Result is Presented to the User
- Siebel Open UI Architecture Summary
- Summary of Differences (In Yellow)
- Recommended Practices
- Lesson Highlights
- Practice

Administering the Manifest

- Objectives
- The Siebel Open UI Manifest
- The Manifest Files View
- The Manifest Expressions View
- The Manifest Administration View
- Use the Manifest Administration View to Specify What Files Should be
- Downloaded When
- 1. The UI Objects Applet: Object Type
- 1. The UI Objects Applet: Usage Type
- 1. The UI Objects Applet: Name
- 2. The Object Expression Applet: Groups
- 2. The Object Expression Applet: Expressions
- 3. The Files Applet
- Caveat: Presentation Model and Physical Renderer Files
- Example: Customize Siebel Open UI
- 1. Create or Customize JavaScript Files
- Example: A Section of a Custom JavaScript File
- 2. Add Custom File(s) to the Manifest Files View
- 3. If Necessary, Add Expressions to the Manifest Expressions View
- 4. Administer the Manifest to Include the Custom File(s)
- 5. Test your Customizations
- Lesson Highlights
- Practice

Presentation Model

- Objectives
- Review: Proxy Framework
- Review: The Presentation Model
- A Presentation Model Object
- Customizing the Presentation Model Examples
- How to Customize a Presentation Model
- Example
- Terminology
- Implement a Presentation Model Customization
- 1. Verify the Object Class does not yet Exist
- 2. Add the Class to the Namespace
- 3. Define the File and Parent File(s) for the Class
- 4. Implement a Constructor Function for the Class

4.1 Declare the Class as a Function 5-15

4.2 Create the class Constructor 5-16

4.3 Specify the Class as an Extension of its Parent Class 5-17

4.4 Add Implementation Code to the Class 5-18

4.4 Add an Init Function 5-20

4.5 Add the Custom Methods for the Class 5-21

- Complete Example Page 1 of 3
- Complete Example Page 2 of 3
- Complete Example: Page 3 of 3
- Review: Presentation Model Structure
- Lesson Highlights
- Practice

Physical Renderer

- Objectives
- The Physical Renderer
- A Physical Renderer Object
- Customizing the Physical Renderer Examples
- How to Customize a Physical Renderer
- Example
- Implement a Physical Renderer Customization
- 1. Verify the Object Class Does Not Yet Exist
- 2. Add the Class to the Namespace
- 3. Define the File and Parent File(s) for the Class
- 4. Implement a Function for the Class

4.1 Declare the Class as a Function 6-13

4.2 Create the Class Constructor 6-14

4.3 Specify the Class as an Extension of its Parent Class 6-15

4.4 Add Implementation Code to the Class 6-16

4.5 Add the Custom Methods for the Class 6-17

- Review: Physical Renderer Structure
- Lesson Highlights
- Practice

Debugging

- Objectives

- Suggestions for Debugging a Script
- Browser Developer Tools
- Invoke Developer Tools
- Verify the Script was Downloaded
- Reasons for Failure to Download
- Use SiebelJS.Log()
- Output of SiebelJS.Log()
- The debugger Statement
- Debugging Using Browser Developer Tools
- Recommendations
- Lesson Highlights
- Practice

Cascading Style Sheets and Open UI Themes

- Objectives
- Review: Cascading Style Sheets (CSS)
- CSS Rules
- More Advanced CSS Rules
- Conflicting CSS Styles
- Location of CSS Files
- Examining CSS Rules
- Theme
- Themes for Siebel Open UI
- Selecting a Different Theme
- Defining a Theme
- Modify an Existing Theme
- 1. Locate the Styling Rules
- 2. Edit the Rules
- 3. Create the CSS File: Copy the Modified Rules to a New File
- 3. Create the CSS File: Save the File
- 4. Administer the Manifest
- 5. Test the Modified Theme
- 5. Test the Modified Theme: Verify the Rules
- Create a Custom Theme
- 1. Create the Style Sheets
- 2. Add the Theme to the List of Values
- 3. Administer the Manifest
- 4. Test the Custom Theme
- Lesson Highlights
- Practice

Siebel Mobile Applications

- Objectives
- Siebel Mobile Applications
- Mobile Application Modes

- As-Delivered Siebel Mobile Applications
- User Experience
- Features
- JavaScript Files for Siebel Mobile Applications
- CSS Files for Siebel Mobile Applications
- Enabling an Application Object Manager for Siebel Mobile Applications
- Architecture of Siebel Mobile Applications
- Mobile Applets
- Mobile Applet Web Templates
- Mobile Views
- Mobile View Web Templates
- Mobile Screens
- Customizing the User Interface for Siebel Mobile Applications
- Siebel Disconnected Mobile Applications
- Siebel Disconnected Architecture
- Lesson Highlights
- Practice