

Siebel Open UI Adv JavaScript API

Student Guide

D84417GC10

Edition 1.0

April 2014

D85883

ORACLE®

Authors

Customer Experience (CX)
Curriculum Team

Technical Contributors and Reviewers

Customer Experience
(CX) Team

Publishers

Michael Sebastian Almeida
Syed Ali

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Contents

1 Course Introduction

- Lesson Agenda 1-2
- Instructor and Class Participants 1-3
- Training Site Information 1-4
- Course Audience 1-5
- Course Prerequisites 1-6
- Course Goal 1-7
- Course Objectives 1-8
- Course Methodology 1-9
- Course Materials 1-10
- Information Sources 1-11
- Course Agenda 1-12

2 Overview of Open UI Customization

- Objectives 2-2
- Siebel Open UI 2-3
- The Open UI Framework 2-4
- Siebel Open UI Client 2-5
- Parts of the Open UI Framework 2-6
- Customizing the Open UI Framework 2-10
- The JavaScript Application Programming Interface (JSAPI) for Open UI 2-11
- Lesson Highlights 2-12
- Practice 2-13

3 Administer the Manifest

- Objectives 3-2
- The Manifest 3-3
- Administering the Manifest: Manifest Files 3-4
- Selecting or Adding a File 3-5
- Administering the Manifest: Manifest Expressions 3-6
- Expression 3-7
- Creating a New Conditional Expression 3-8
- Administering the Manifest: Manifest Administration 3-9
- UI Objects 3-10
- Extending Existing Objects 3-11

Default Behavior 3-12
Object Expressions 3-13
Multiple Conditions 3-14
Complex Conditions 3-15
Setting up a Complex Condition 3-16
Files 3-18
Administer the Manifest 3-19
1. Identify the Name of the Object to Customize 3-20
2. Ensure the File to Apply is Registered 3-21
3. Select the Object to Customize 3-22
4. Associate the Object with a Condition 3-24
5. Associate the Object/Condition with a File 3-25
6. Apply the Modified Manifest 3-26
7. Test the Customization 3-27
Removing a Customization 3-28
Lesson Highlights 3-29
Practice 3-30

4 Using jQuery

Objectives 4-2
jQuery 4-3
jQuery in JavaScript 4-4
Selecting HTML Elements in jQuery 4-5
Selecting Child HTML Elements in jQuery 4-7
Performing Actions on the Selected Set 4-8
Function Actions 4-9
CSS Function Example 4-10
Setting a Style Defined in a CSS Style Sheet 4-11
html Function Example 4-12
after/before Function Example 4-14
Attaching Event Handler Function to Selected Set 4-15
Examples of a Click Event Handler with Various Selectors 4-16
Storing Selected Elements in a Variable 4-19
Chaining Actions 4-20
jQuery in a Physical Renderer 4-21
Choosing a jQuery Selector 4-22
Disabling the Cache 4-25
Caching versus Re-Reading the Manifest 4-26
Use jQuery 4-27
1. Open Developer Tools 4-28
2. Identify Control(s) / Field(s) to be Read or Manipulated 4-29

- 3. Add jQuery Code to an Appropriate Method 4-30
- 4. Apply the PR File to the Applet 4-31
- 5. Test Your New Customization 4-32
- Lesson Highlights 4-33
- Practice 4-34

5 Customizing a Presentation Model for a Form Applet

- Objectives 5-2
- The Presentation Model 5-3
- Customizing a PM with a New Method 5-4
- The Collection of Controls 5-5
- Finding Control Key and FieldName Values 5-6
- Methods Used in a PM to Work with Controls in an Applet 5-7
- A Basic PM for a Form Applet 5-8
- Using AddMethod() to Hook into a Framework Method 5-15
- Settings Arguments to AddMethod() 5-16
- A Custom Method Bound to FieldChange 5-17
- The ShowSelection Method 5-18
- Accessing Fields in a Method Bound to ShowSelection 5-19
- Accessing an Individual Control 5-20
- Accessing the Value in a Control 5-21
- Another Way to Access an Individual Control 5-22
- Customize a PM for a Form Applet 5-23
 - 1. Create the Basic PM file 5-24
 - 2. Use AddMethod() to bind a custom method to the appropriate internal framework method 5-25
 - 3. Write a new method to implement the business requirement 5-26
 - 4. Save the file with its new file name 5-28
 - 5. Apply the new PM to the applet 5-29
 - 6. Test the Solution 5-30
- Lesson Highlights 5-31
- Practice 5-32

6 Customizing a Physical Renderer for a Form Applet

- Objectives 6-2
- A Physical Renderer (PR) 6-3
- Custom Physical Renderers 6-4
- Overriding a Method from the Parent Class 6-5
- Some of the JavaScript API Methods Used in a PR 6-6
- A Basic PR for a Form Applet 6-7
- The BindData Method 6-14

- Adding New Functionality to BindData 6-15
- Adding a New Method 6-16
- Adding a New Prototype Method 6-17
- Adding a Non-Prototype Method 6-18
- Accessing a Specific Control/Field in the PR 6-19
- The GetInputName() Method 6-20
- Customize a PR for a Form Applet 6-21
- 1. Create a Basic PR File for a Form Applet 6-22
- 2. Modify the File to Achieve the Business Goal 6-23
- 3. Apply the New PR to the Applet 6-25
- 4. Test the Solution 6-26
- Lesson Highlights 6-27
- Practice 6-28

7 Presentation Model / Physical Renderer Interaction

- Objectives 7-2
- Separation of Concerns in the Open UI 7-3
- Advantages of SoC in Siebel Open UI 7-4
- PM / PR Interoperation 7-5
- Example of MVC Interoperation Using a Property in the PM Bound in the PR 7-6
- Methods Used to Create Interactions Between PR and PM 7-7
- Creating a Custom Property in the PM 7-8
- Changing the Custom Property in the PM 7-9
- Binding a Callback Method in the PR to a Property Change in the PM 7-10
- Accessing the Custom PM Property Value in the Callback Method in the PR 7-11
- Implementing PM / PR Interaction by Binding a Property in the PM to the PR 7-12
- Tips on Locating Code 7-16
- Tips on Locating Code: Details 7-17
- Create PM/PR Interaction 7-18
- 1. Add a Property to the PM 7-19
- 2. In the PM, Add Code to Set the Property As Needed 7-20
- 3. In the PR, Bind a Callback to the Property 7-21
- 4. Code the Callback Method 7-22
- 5. Refresh the View 7-23
- Lesson Highlights 7-24
- Practice 7-25

8 Debugging

- Objectives 8-2
- Sources of Problems in Scripts 8-3
- Verify that the Script was Loaded at Runtime 8-4

- Errors that Can Prevent Script Loading 8-5
- Check for Syntax Errors in the JavaScript that Prevent Execution from Starting 8-6
- Check for Runtime Exceptions that Halt Execution 8-7
- Search the Code for Logical Errors 8-8
- Text Logging and Interactive Alerts 8-9
- Debugging Tools 8-10
- Viewing Variable Values 8-12
- Inspecting the Call Stack 8-14
- Stepping Through Code 8-15
- Ensure that the Cache Is Disabled or Cleared 8-16
- Debug a Script 8-17
 1. Verify that the Cache is not Active 8-18
 2. Verify that Script has been Loaded 8-19
 3. Verify that There are No Syntax Errors in the JavaScript 8-20
 4. Verify that There are No Runtime Errors in the JavaScript 8-21
 5. Use Debugging Tools to Search for Logical Errors 8-22
- Lesson Highlights 8-24
- Practice 8-25

9 Customizing a List Applet

- Objectives 9-2
- Structure of a List Applet 9-3
- Placeholder 9-4
- Record set 9-5
- An Individual Record 9-6
- Accessing an Individual Row by ID 9-7
- Accessing a Field within a Row 9-8
- A Basic PR for a List Applet 9-9
- To Search for a Row in a List Applet 9-10
- Alternate Looping Technique 9-11
- BindData Method in a List Applet 9-12
- ShowUI Method 9-13
- Coloring Every Other Line of a List Display 9-14
- To Search for a Cell Value in a List Applet 9-16
- A PM for List Applets 9-17
- Detecting Changed Values in a List PM Code 9-18
- Detecting Changed Values in a List PR Code 9-20
- Customize a List Applet 9-21
 - 1: Create the Basic PR File 9-22
 - 2: Add Code to BindData() 9-23
 - 3: Apply the New PR to the Applet 9-26

4: Test Your Changes 9-27
Lesson Highlights 9-28
Practice 9-29

10 Customization of Non-Applet Objects

Objectives 10-2
Applet and Non-Applet UI Elements 10-3
postload 10-4
Adding a Listener for postload 10-5
Application-Level Customizations 10-6
Example Global Customizations Appropriate for Use in Postload 10-8
Making Applets Collapsible 10-9
Sorting in jQuery 10-10
A Custom Sorting Function 10-11
Customize a Non-Applet Element using Postload 10-12
1. Create a Custom File with a Listener for the postload Event 10-13
2. Write Custom Code in the Event Handler 10-14
3. Administer the File 10-16
4. Test your Changes 10-17
Lesson Highlights 10-18
Practice 10-19

11 Using External Libraries and Web Sites

Objectives 11-2
Using External Code 11-3
External jQuery Libraries 11-4
Using External jQuery Libraries 11-5
Copy the External Package to Your Site 11-6
Include the External jQuery Package as a Dependency in Your Code 11-7
Follow the Documentation for the External Package 11-8
JavaScript Code Accessed via the Web 11-9
Accessing Code from a URL Asynchronously 11-10
Steps to Access Code Online via a URL 11-11
Asynchronously Load the API 11-12
Load Module/Packages with a Callback 11-15
Check that Code Is Available 11-16
Accessing a Web-Based Application 11-17
Calling an External Web Application 11-18
Steps to Open an External Web Application in a New Window Based on a User
Action in the UI 11-19
Create a String with an Appropriate URL 11-20

Populating the URL from a Value in the Applet 11-21
Write JavaScript to Open a Window to the URL 11-22
Call JavaScript When the User Action is Detected in the UI 11-23
Creating a New Link Dynamically 11-24
Access an External Web Site with Data from Siebel 11-25
1. Create a Presentation Model (PM) 11-26
2. Add a Property and Methods to the PM 11-27
3. Create a Physical Renderer (PR) 11-28
4. Add HTML that will be Used as a Link 11-29
5. Add a Binding 11-30
6. Apply the New PR / PM 11-32
7. Test the New Functionality 11-33
Lesson Highlights 11-34
Practice 11-35

12 Siebel Business Services and Profile Attributes

Objectives 12-2
Siebel Business Service 12-3
Accessing a Siebel Business Service 12-4
Make the Service Available to the Application 12-5
Write Code to Access the Service 12-6
Profile Attributes 12-8
Recommended Practice 12-9
Create a Manifest Expression 12-10
Use the Expression to Apply a Customization Conditionally 12-11
Conditional Expressions and Customization 12-12
Solution Steps 12-13
1. Identify the Service, its Methods, and the Parameters for each Method 12-14
2. Configure the Application to Use the Business Service 12-16
3. Write Code for the Business Requirement 12-19
3. Write Code in a Custom Postload Handler 12-20
4. Apply the New Code to the Application 12-28
5. Test Your Change 12-29
Lesson Highlights 12-30
Practice 12-31

13 Using Siebel Tools with Siebel Open UI

Objectives 13-2
Add Presentation Model Properties to a View, Applet, or Control 13-3
Presentation Model Properties as User Properties 13-4
Read Presentation Model Properties in the Presentation Model Code 13-6

- 1. Load the Siebel Constants 13-7
- 2. Get the Presentation Model Properties 13-8
- 3. Parse the Property Set 13-9
- Complete Code Example: Applet 13-10
- Result 13-11
- Special Property: EnableDragAndDropInList 13-12
- Enable Drag-and-Drop Imports 13-13
- 1. Add the User Property to the Applet 13-14
- 2. Prepare the Spreadsheet 13-15
- 3. Test the Results 13-16
- Lesson Highlights 13-18
- Practice 13-19

14 User Preferences

- Objectives 14-2
- User Visualization 14-3
- User Visualization in Siebel Open UI 14-4
- Selecting Default User Visualization 14-5
- Configuring Visualizable Applets 14-6
- Visualizable Web Template 14-7
- Physical Renderer and Presentation Model 14-8
- Create a Visualization Model 14-10
- 1. Add the Web Templates to the Applet 14-11
- 2. Edit the Web Layouts 14-12
- 3. Administer the Web Templates 14-13
- 4. Administer the Manifest 14-14
- 5. Test the Results 14-15
- Additional Visualizations 14-16
- Other User Preferences 14-17
- User Preferences 14-18
- Read a User Preference 14-19
- Set a User Preference 14-20
- Example: A Button that Hides and Shows Applets 14-22
- 1. Add the Button to the User Interface 14-23
- Verify the Button 14-24
- 2. Add the Script that Reacts to the Button 14-25
- 3. Test the Results 14-28
- Lesson Highlights 14-29
- Practice 14-30

15 Configuring Siebel Mobile Views

- Objectives 15-2
- Siebel Mobile Applications 15-3
- As-Delivered Siebel Mobile Applications 15-4
- JavaScript Files for Siebel Mobile Applications 15-5
- CSS Files for Siebel Mobile Applications 15-6
- Architecture of Siebel Mobile Applications 15-7
- Mobile Applets 15-8
- Mobile Applet Web Templates 15-9
- Mobile Views 15-10
- Mobile View Web Templates 15-11
- Mobile Screens 15-12
- Customizing the User Interface for Siebel Mobile Applications 15-13
- Customizing a Siebel Mobile Application 15-14
- Add a View to a Siebel Mobile Application 15-15
 1. Select the View You Want to Add 15-16
 2. Copy the View and Assign It a Mobile Web Template 15-17
 3. Replace the Parent Applet with Its Associated Mobile Applet 15-18
 4. Add the View to the Screen 15-19
 5. Register the New View 15-20
 6. Test the Results 15-21
- Lesson Highlights 15-22
- Practice 15-23

16 Deploying to a Production Environment

- Objectives 16-2
- Technical Challenge 16-3
- Technical Solution: Migrate Customizations to the Server 16-4
 1. Use Standard Techniques to Migrate Siebel Repository Changes 16-5
 2. Copy the Files 16-6
- File Considerations 16-7
 3. Administer the Manifest for the Enterprise 16-8
- Manifest Considerations 16-9
 4. Test the Results 16-10
- Lesson Highlights 16-11
- Practice 16-12