

Oracle Fusion Middleware 11g: Build Applications with ADF II

Student Guide

D56163GC20

Edition 2.0

July 2010

D68156

ORACLE

Authors

Jeff Gallus
Pam Gamer

Technical Contributors and Reviewers

Lynn Munsinger
Duncan Mills
Frank Nimphius

Editors

Vijayalakshmi Narasimhan
Malavika Jinka

Publishers

Syed Imtiaz Ali
Sumesh Koshy

Copyright © 2010, 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

Preface

I Oracle Fusion Middleware 11g: Build Applications with ADF II

11g: Java Developer Curriculum I-2

Objectives I-3

Lesson 1: Exploring the Project Structure and Core File Architecture I-4

Lesson 2: Advanced ADF Business Components I-5

Lesson 3: Using ADF Business Components for Validation, Calculations,
and LOV I-6

Lesson 4: Using Inheritance in the Business Domain Layer I-7

Lesson 5: ADF as Part of a Service-Oriented Architecture I-8

Lesson 6: Implementing Data Binding Concepts I-9

Lesson 7: Advanced ADF Data Binding and Life Cycle I-10

Lesson 8: Working with Managed Beans and JavaServer Faces I-11

Lesson 9: ADF Faces Advanced Features I-12

Lesson 10: Integrate with WebCenter I-13

Lesson 11: Customize the Application Look and Feel with Skins I-14

1 Exploring the Project Structure and Core File Architecture

Objectives 1-2

Agenda 1-3

Launching JDeveloper 1-4

Using the Component Palette 1-5

Using the Structure Window 1-6

Agenda 1-7

Metadata: Overview 1-8

Examining Metadata File Relationships 1-9

Examining Metadata File Hierarchy 1-11

Data Control Files 1-12

Configuring Application Modules: `bc4j.xcfg` 1-13

Data Binding Files 1-14

Defining a Binding Container: Page Definition Files 1-15

Registering Data Bindings: `Databindings.cpx` 1-17

Finding Data Controls: `adfm.xml` 1-19

| | |
|---|------|
| Examining the Binding Context | 1-20 |
| ADF Faces and Web Configuration Files | 1-21 |
| Examining ADF Faces and Web Configuration Files | 1-22 |
| Configuring the Application's Unbounded Task Flow: <code>adfc-config.xml</code> | 1-23 |
| Configuring the Application: <code>faces-config.xml</code> | 1-24 |
| Configuring Skins: <code>trinidad-config.xml</code> and <code>trinidad-skins.xml</code> | 1-25 |
| Setting Run-Time Configuration Parameters: <code>web.xml</code> | 1-26 |
| Examining Additional Metadata Files | 1-27 |
| Agenda | 1-28 |
| File Layout: Application | 1-29 |
| File Layout: Model Project | 1-30 |
| File Layout: User Interface Project | 1-31 |
| File Layout: User Interface Project | 1-32 |
| Summary | 1-33 |
| Practice 1 | 1-34 |

2 Advanced ADF Business Components

| | |
|--|------|
| Objectives | 2-2 |
| Agenda | 2-3 |
| Business Components: Preferences | 2-4 |
| Business Components: Naming Conventions | 2-5 |
| Business Components: View Objects (VO) | 2-6 |
| Business Components: Read-Only VO | 2-7 |
| Business Components: VO Tuning | 2-8 |
| Testing Business Components | 2-10 |
| Testing Java Code with JUnit | 2-11 |
| Agenda | 2-12 |
| Globally Extending ADF Business Components Functionality | 2-13 |
| ADF Business Components Framework Extension Classes | 2-14 |
| Creating a Framework Extension Class | 2-15 |
| Basing an ADF Component on a Framework Extension Class | 2-16 |
| XML-Only Component on a Framework Extension Class | 2-17 |
| Custom Java Component on a Framework Extension Class | 2-18 |
| Considerations | 2-19 |
| Creating a Layer of Framework Extensions | 2-20 |
| Agenda | 2-21 |
| ADF Business Rules in PL/SQL | 2-22 |
| Typical Environment | 2-23 |
| Creating a Database View-Based Entity Object | 2-24 |
| What Is Created by Default? | 2-25 |

- Centralizing Details in the Base Class 2-26
- Implementing the Stored Procedure for DML Operations 2-27
- Leveraging the Helper Method to Invoke Insert, Update, and Delete Procedures 2-28
- Agenda 2-29
- Controlling Entity Posting Order to Avoid Constraint Violations 2-30
- DB Constraints, Compositions, and Associations 2-31
- Transactions with Compositions 2-32
- Transactions with Associations 2-33
- Violating a Database Constraint 2-34
- Forcing the Master to Post Before the Detail 2-35
- Override `postChanges()` in `ProductsImpl.java` 2-36
- `postChanges()` Example 2-37
- Trigger-Assigned Primary Key from the Database Sequence 2-38
- Refreshing Reference to DB Sequence–Assigned Foreign Keys 2-39
- Summary 2-41
- Practice 2 Overview: Advanced ADF Business Components 2-42

3 Using ADF Business Components for Validation, Calculations, and LOV

- Objectives 3-2
- Agenda 3-3
- Cascading Lists 3-4
- Creating the Basics 3-5
- Dependent List Example 3-6
- Creating the View Criteria 3-7
- Defining View Accessors to the List Data Sources 3-8
- Attaching the List VO to an Attribute 3-9
- Setting the List UI Hints 3-10
- Setting the List for the Dependent List 3-11
- Agenda 3-12
- Transient Attributes 3-13
- Creating Transient Attributes 3-14
- Entering a Calculation 3-15
- Testing the Calculation 3-16
- Agenda 3-17
- Entity Object Validation 3-18
- Using Groovy for Validation 3-19
- Constraining Dependent Values 3-20
- Defining the Conditional Expression 3-21
- Agenda 3-22
- Defining the Failure Message 3-23

- Storing Error Messages as Resources 3-24
- Validating Foreign Keys 3-25
- Adding a Validation Rule 3-26
- Managing Failures 3-27
- Testing the Validation 3-28
- Script Expression Validation 3-29
- Groovy Rule Definition 3-30
- Failure Handling 3-31
- Summary 3-32
- Practice 3 Overview: Advanced ADF Business Components 3-33

4 Using Inheritance in the Business Domain Layer

- Objectives 4-2
- Agenda 4-3
- Using Inheritance in the Business Domain Layer 4-4
- Creating a Business Component That Extends Another 4-6
- Inheritance Example 4-7
- Creating the Base Entity Object in an Inheritance Hierarchy 4-8
- Creating a Subtype Entity Object 4-9
- Using Polymorphic View Objects 4-11
- Using View Objects with Polymorphic Entity Usage 4-12
- Creating a View Object with Polymorphic Entity Usage 4-13
- Using View Objects with Polymorphic View Rows 4-14
- Creating View Objects with Polymorphic View Rows: Defining the VO Hierarchy 4-15
- Creating View Objects with Polymorphic View Rows: Defining Subtypes in the Data Model 4-17
- Summary 4-18
- Practice 4 Overview: Using Inheritance 4-19

5 ADF as Part of a Service-Oriented Architecture

- Objectives 5-2
- Agenda 5-3
- Fusion Architecture 5-4
- Fusion Development 5-5
- Oracle SOA Platform 5-6
- Agenda 5-8
- Business Process Development 5-9
- BPEL Process Manager 5-10
- Orchestrating Services 5-11
- Oracle BPEL Process Manager 5-12

- Human Workflow 5-13
- BPEL and Human Workflow 5-15
- Visual Development in JDeveloper 5-17
- Service Component Architecture 5-18
- Access to Business Data 5-19
- Service Data Object Components 5-20
- Agenda 5-22
- SDO in JDeveloper 5-23
- SDO Creation in JDeveloper 5-24
- What Gets Created 5-25
- Deploying and Testing 5-26
- Consuming SDO in JDeveloper 5-27
- What Gets Created 5-28
- Implementing Human Task Flows 5-29
- Summary 5-30
- Practice 5 Overview: ADF as Part of a Service-Oriented Architecture 5-31

6 Implementing Data Binding Concepts

- Objectives 6-2
- Agenda 6-3
- What Is Data Binding? 6-4
- The Heart of ADF: ADFm 6-5
- ADF Model Data Control Concepts 6-6
- Application Module in the Data Control Palette 6-7
- Nested Application Modules 6-8
- Changing the Application Module Data Control Name 6-9
- ADF Data Bindings Concepts 6-10
- ADFm: Data Controls and Data Bindings 6-11
- ADF Model: Overview 6-12
- ADF Model Bindings 6-13
- Agenda 6-14
- DataBindings.cpx File 6-15
- DataBindings.cpx File: Details 6-16
- DataBindings.cpx Properties: Synchronization Mode 6-17
- Agenda 6-18
- PageDef.xml File 6-19
- Examining a Page Definition File 6-20
- Bindings Sections 6-21
- Identifying Types of Bindings 6-22
- Types of Bindings: Attribute 6-23

- Types of Bindings: Table 6-24
- Types of Bindings: List 6-25
- Types of Bindings: Action 6-26
- Using View-Level Operations in Action Bindings: Setting the Current Row 6-27
- Using View-Level Operations in Action Bindings: Inserting a New Row 6-28
- Create and CreateInsert 6-29
- Binding Objects in the Executables Section 6-30
- Controlling the Execution of Executables 6-31
- Leveraging Iterators 6-33
- Binding Objects in the Parameters Section 6-34
- Agenda 6-35
- Data Controls and Standard Metadata Interfaces 6-36
- Working with Named Bind Variables 6-37
- Transaction Control Operations 6-38
- Schematic Representation of a Query Process 6-39
- Summary 6-40
- Practice 5 Overview: Implementing Data Binding Concepts 6-41

7 Advanced ADF Data Binding and Life Cycle

- Objectives 7-2
- Agenda 7-3
- Defining the JSF Life Cycle 7-4
- JSF Life Cycle 7-5
- JSF Request Life Cycle: Restore View 7-6
- JSF Request Life Cycle: Apply Request Values 7-7
- JSF Request Life Cycle: Process Validations 7-8
- JSF Request Life Cycle: Update Model Values 7-9
- JSF Request Life Cycle: Invoke Application 7-10
- JSF Request Life Cycle: Render Response 7-11
- Using the immediate Attribute 7-12
- Agenda 7-13
- ADF Faces Life Cycle Extension 7-14
- ADF Faces Life Cycle Optimization 7-15
- Agenda 7-16
- ADF Life Cycle 7-17
- The Big Picture 7-18
- ADF Life Cycle in Action: Step 1 7-19
- ADF Life Cycle in Action: Step 2 7-20
- ADF Life Cycle in Action: Step 3 7-21

- ADF Life Cycle in Action: Step 4 7-22
- ADF Life Cycle in Action: Step 5 7-23
- Task Flows 7-24
- Task Flow Components 7-25
- More Task Flow Components 7-26
- Object Scope Life Cycle 7-27
- More on Object Scope Life Cycles 7-28
- Available Object Scopes 7-29
- About Regions 7-30
- Task Flow and ADF Life Cycle 7-31
- Declarative Control over Individual Page Life Cycle 7-32
- Summary 7-33
- Practice 7 Overview: Data Binding in Action 7-34

8 Working with Managed Beans and JavaServer Faces

- Objectives 8-2
- Agenda 8-3
- JSF Review 8-4
- Key Terms 8-5
- JSF Architecture 8-6
- Standard Component Model for Web Development 8-7
- JSF Expression Language 8-8
- Summary of Key Elements 8-9
- JSF Suggestions 8-10
- Agenda 8-11
- EL Used in Managed Beans 8-12
- Expression Language: Examples 8-13
- More Expression Language 8-14
- Immediate EL Syntax: `{ }` 8-15
- Deferred EL Syntax: `# { }` 8-16
- Agenda 8-17
- Managed Beans 8-18
- Defining Managed Beans 8-19
- Using Managed Beans 8-20
- Accessing One Managed Bean from Another Managed Bean 8-21
- Hard-Coded Access 8-22
- Using Properties to Access Values 8-23
- Managed Properties 8-24
- Managed Properties: Examples 8-25
- JSF Value Binding 8-26
- Writing Code That Interacts with Bindings 8-27

- Accessing the Bindings 8-28
- Writing Code That Interacts with Bindings 8-29
- Configuring ADF Binding Access for Managed Beans 8-30
- Getting and Setting Bound Data 8-31
- Executing Methods in ADF 8-32
- Executing Bound Methods on the JSF Page Load 8-33
- Context Objects 8-34
- JavaServer Faces Backing Beans 8-35
- Summary 8-36
- Practice 8 Overview: Working with JSF and Managed Beans 8-37

9 ADF Faces Advanced Features

- Objectives 9-2
- Agenda 9-3
- What Does ADF Faces Bring to JSF? 9-4
- ADF Faces Components 9-5
- Using ADF Data Visualization Components 9-6
- Visualizing Data 9-7
- Agenda 9-9
- Partial Page Rendering 9-10
- PPR: `partialSubmit` 9-11
- PPR: `autoSubmit` 9-12
- PPR: `partialTriggers` 9-13
- Agenda 9-14
- Exploring Task Flow Strategies 9-15
- Using a Single `.jspx` with Regions 9-16
- Using `.jspx` in Bounded Task Flows 9-17
- Putting all Pages in a Single Task Flow 9-18
- Agenda 9-19
- ADF Faces: Printable Pages 9-20
- Agenda 9-21
- Editable Check Box in the ADF Faces Table 9-22
- Model Requirements: Business Components 9-23
- Model Requirements: Custom Code 9-24
- User Interface Modifications 9-25
- Run-Time Behavior 9-26
- Agenda 9-27
- Creating Data-Bound Graphs 9-28
- Agenda 9-29
- Displaying a Pop-Up Window 9-30

Summary 9-31
Practice 9 Overview: Using ADF Faces 9-32

10 Integrate with WebCenter

Objectives 10-2
Agenda 10-3
What Is WebCenter? 10-4
Key Benefits of Oracle WebCenter 10-5
Oracle WebCenter Components 10-6
Oracle WebCenter Components Overview 10-8
WebCenter Framework 10-9
WebCenter Services 10-10
WebCenter Services List of Services 10-11
Oracle Composer 10-12
WebCenter Applications 10-13
WebCenter Spaces 10-14
Agenda 10-16
What Is a Portlet? 10-17
Anatomy of a Portlet 10-18
Portlet Modes 10-19
Customization and Personalization 10-20
Consuming a Portlet: Overview 10-21
Registering a Portlet Producer 10-22
Consuming a Portlet 10-23
Agenda 10-24
Oracle Composer 10-25
Run-Time Editing and Customization 10-26
Metadata Services (MDS) 10-27
Layered Customization 10-28
Agenda 10-29
Oracle Composer 10-30
Using Customizable Components at Design Time 10-31
Adding Customizable Components at Run Time 10-32
Adding Components 10-33
Custom Resource Catalog: Overview 10-34
Custom Resource Catalog ADF Faces Components 10-35
Custom Resource Catalog Custom Task Flows 10-36
Custom Resource Catalog Portlet Producers and Portlets 10-37
Installing WebCenter Framework 10-38
What Gets Installed 10-39
WebCenter Framework in JDeveloper 10-40

Summary 10-41
 Practice 10 Overview: Integrate with WebCenter 10-42

11 Customize the Application Look and Feel with Skins

Objectives 11-2
 Agenda 11-3
 How JavaServer Faces Renders the UI 11-4
 ADF Faces Rich Client Components 11-5
 About Skins 11-6
 More on Skins 11-7
 Skin Examples 11-8
 Agenda 11-9
 Cascading Style Sheets (CSS) 11-10
 Document Object Model 11-11
 CSS Rules 11-12
 More CSS Rules 11-13
 Agenda 11-14
 How Skinning Works 11-15
 Skinning Versus CSS 11-16
 Artifacts 11-18
 Agenda 11-19
 Building Custom Skins 11-20
 Configuration in `trinidad-config.xml` 11-21
 Using Skin Configuration Elements in `trinidad-skins.xml` 11-22
`trinidad-skins.xml` Example 11-24
 Skinning Keys 11-25
 Connecting the Dots 11-27
 Agenda 11-28
 Skin Development Support 11-29
 Agenda 11-31
`panelBox` Example: Skin Selectors 11-32
`panelBox` Component Skin Example 11-33
`panelBox` Component Instance Skin Example 11-34
 Contextual Skins 11-35
 Contextual Skins Example 11-36
 Changing the Splash Screen 11-37
 Agenda 11-38
 Using Multiple Skins in an Application 11-39
 Changing Skins at Run Time: Defining an LOV 11-40
 Discovering Skins at Run Time: Configuring the Skin Family 11-41

Discovering Skins at Run Time: Defining Skin Choices for the LOV 11-42

Summary 11-43

Practice 11 Overview: Customize the Application Look-and-Feel with

Skins 11-44

