



Contents

PART I Overview

1 Overview of Oracle JDeveloper 10g	3
JDeveloper: Past, Present, and Future	4
Past: Product History and Roots	4
Present: Where Is JDeveloper Now?	6
Future: The Vision	8
What Is New in JDeveloper 10g?	9
New Integrated Development Environment (IDE)	9
Application Development Framework	10
Struts	10
Modeling in JDeveloper	11
Business Services	12
Desupported and Deprecated Features	12
Creating Application Code in JDeveloper	13
Application Workspaces	13
Projects	14
JDeveloper Directory Structure	14
Development Steps	14
Hands-on Practice: Build a Client/Server Application Using the JDeveloper Wizards	15
I. Create the Application Workspace and Database Connection	16
II. Create the Model Project	20
III. Create the View Project	25
Hands-on Practice: Create a Simple JSP Page	29
I. Create the JSP Project	29
II. Create the JSP File	30
2 The Integrated Development Environment Overview	33
The JDeveloper 10g IDE	35
IDE Window	35
About Monitor Resolution	37

Editor Window	38
Dockable Windows	40
Customizing the IDE	43
Preferences	44
Main Toolbar	44
Main Menu	45
About Javadoc	50
The Help System	54
Finding the Help Topic	56
Displaying the Help Topic	56
Locating Help Centrally	56
Software Configuration Management	57
CVS Concepts	57
Hands-on Practice: Set Up CVS Locally	58
I. Install and Set Up the CVS Server	59
II. Set Up a CVS Server Connection	60
III. Import and Check Out a Module	61
3 IDE Tools	63
Navigators	65
Application Navigator and System Navigator	65
Connection Navigator	68
Run Manager	69
Structure Window	70
Structure Window Icons	70
Other Structure Window Operations	71
Structure Window Views	71
Component Palette	72
Code Snippets	72
Property Inspector	73
Properties	73
Events	75
Log Window	76
Code Editor	76
End Tag Completion	77
Code Templates	77
Code Insight and Tag Insight	78
Other Text Editing Features	80
Editing PL/SQL	81
Code Editor Style Sheet Support	82
Debugger	84
Debugger Windows	85
Debugging Java Code	85
Debugging PL/SQL	86
Other Tools for Improving Code	86
Visual Editors	86
HTML and UIX Previewers	87

New Gallery	89
Wizards	90
New File Dialogs	91
Data Control Palette	91
Other Editors	92
XML Editor	92
Class Editor	92
EJB Module Editor	94
Viewer Windows	95
Image Viewer	95
Archive Viewer	95
Tables Viewer	96
Sequence Viewer	96
SQL Worksheet	97
Modelers and Diagrammers	98
Types of Diagrams	99
Creating a UML Diagram	99
Some Diagramming Operations	100
XML Schema Editor	101
Struts Page Flow Diagrammer	102
4 Introduction to the Oracle Application Development Framework	103
What Is a Framework?	104
Why Use a Framework?	105
What Is a Development Framework?	106
Why Use a Development Framework?	106
Support for the Framework	107
What Is Oracle Application Development Framework?	107
Who Will Use ADF?	108
ADF User Community Support	109
Development Frameworks in Previous JDeveloper Releases	109
ADF Architecture Model	110
MVC	110
ADF Layers and Components	110
ADF Code Libraries	118
Some ADF Libraries	118
ADF Development Method and IDE Support	120
The Steps	120
5 Java Language Concepts for JDeveloper Work	131
Why Java?	132
Benefits	133
Drawbacks	134
Transitioning to Java	135
Object Orientation Concepts	136
Handling and Storing Data	137
Inheritance	139
Other Primary Object-Oriented Concepts	140

Java Language Review	140
Annotated Java Code Example	140
Annotated Use of the Box Example Class	146
Other Java Language Concepts	147
Hands-on Practice: Create Java Class Files	160
I. Make an Application Workspace and Project	160
II. Create and Test Java Class Files	161
6 Naming Conventions	167
The Importance of Using Naming Conventions	169
The Consistency Issue	170
General Naming Convention Considerations and Guidelines	170
Use Prefixes and Suffixes	171
Consider the Capabilities of the Language	171
Be Aware of Case Sensitivity	172
Use a Dictionary of Allowable Words	172
Use Abbreviations Carefully	172
Recognized Naming Conventions in Java	173
Constants (Final Variables)	173
Classes	173
Class Instances, Exceptions, Methods, and Variables	174
Packages	174
JDeveloper-Specific Naming Conventions	175
Types of Objects to Name	175
Workspaces	176
Projects	176
Connections	176
Class Source Files	177
Libraries	177
Client Data Models	177
UML Diagram Elements	177
ADF Business Components	177
Java Client Application Files	179
Java Client UI Components	179
Web Client Components	179
7 J2EE Architectures and Deployment Alternatives	181
An Overview of J2EE	183
J2ME	183
J2SE	183
J2EE	184
Deploying a J2EE Application	186
J2EE Archive Files	187
Deployment Descriptor Files	187
Deploying Archive Files	188
JDeveloper and J2EE	188
Deploying J2EE Applications in JDeveloper	189

Java Client Architectures	189
Java Applications	189
Applets	194
Hands-on Practice: Deploy a Java Application	196
I. Prepare a Simple ADF BC Java Application	197
II. Create a Java Application JAR File	198
III. Create a Batch File and Run the JAR File	202
JavaServer Pages Architecture	204
Java Servlet Overview	204
JSP Technology Overview	205
Where Does OC4J Fit?	209
Hands-on Practice: Deploy a JSP Application	209
I. Create a JSP Application	210
II. Set Up OC4J	212
III. Deploy the JSP Application	216

PART II

Business Services

8 Introducing ADF Business Components	223
Why Use ADF BC?	225
ADF Business Components, XML, and Java	226
ADF Business Component Groups	226
Business Domain Components	227
Data Model Components	230
Hands-on Practice: Examine a Default ADF BC Layer	234
I. Create an Application Workspace with a Default ADF BC Layer	235
II. Explore the Business Domain Components	238
III. Explore the Data Model Components	242
IV. Test the Default Business Components	248
9 Creating Business Domain Components	253
Creating Default Business Domain Components	254
Entity Attributes	254
Adding and Deleting Attributes	255
Changing Datatypes	255
Representing Column Constraints	255
Synchronizing Entity Object Definitions with the Database	257
Representing Relationships Between Tables	257
Association Cardinality	258
Association Directionality and Accessor Attributes	259
Compositions	261
Representing Oracle Object Types	263
Business Components and Database Object Generation	264
Creating Entity Object Definitions for Table Generation	265
Creating Table Constraints for Generation	266
Creating Domains for Oracle Object Type Generation	268
Generating Database Objects	268

Hands-on Practice: Represent the HR Schema	269
I. Create an Application Workspace and Default Business Domain Components	270
II. Change Entity Attributes	272
III. Clean Up Accessor Attribute Names	274
IV. Create Business Domain Components for Table Generation	276
V. Generate the New Table and Constraints	278
VI. Create a Many-to-Many Association	279
VII. Test the Business Domain Components	281
10 More Complex Business Rules	287
Overview of the Entity Classes	288
Entity Object Classes	289
Entity Definition Classes	290
Entity Collection Classes	291
Manipulating Attribute Values	291
Attribute-Level Validation	293
Validation Rules	293
Validation Domains	295
Setter Method Validation	296
Choosing a Validation Style	297
Entity-Level Validation	298
Entity-Level Validation Rules	299
The validateEntity() Method	300
Choosing a Validation Style	301
Entity-Level Validation and Compositions	301
Hands-on Practice: Add Validation to the HR Business Domain Components	302
I. Remove Unneeded Entity Object Classes	303
II. Apply a Built-in Validation Rule	304
III. Create and Use a Custom Validation Rule	305
IV. Create and Use a Validation Domain	310
V. Provide Entity-Level Validation	312
Adding Default Values to Entity Attributes	314
Static Default Values	314
Dynamically Calculated Default Values	315
The SequenceImpl Class and the DBSequence Domain	315
Calculated Transient Attributes	316
Using Associations in Business Rules	317
Getting a Unique Associated Entity	317
Getting Many Associated Entities	318
Integrating with Business Logic in the Database	318
Integrating with Database Triggers	318
Using Stored Procedures to Perform DML Operations	319
Hands-on Practice: Add More Business Rules to the HR Business	
Domain Components	321
I. Add Defaulting Logic	322
II. Calculate an Attribute	323
III. Add Validation Logic That Uses an Association	326

11 Creating Data Model Components	329
View Attributes, Entity Attributes, and Caching	330
How Entity-Derived View Attributes Are Populated	331
How SQL-Only Attributes Are Populated	333
Entity-Derived vs. SQL-Only View Attributes	334
Entity Object Usages and Table Aliases	337
Keys	338
Refining a View Object's Query	339
Setting the WHERE and ORDER BY Clauses	339
Expert Mode	340
Hands-on Practice: Create View Object Definitions	342
I. Create an Application Workspace for the Data Model Components	343
II. Create a Simple View Object Definition	345
III. Create a SQL-Only View Object Definition	347
IV. Create a View Object Definition with Multiple Entity Object Usages	349
V. Create a View Object Definition with an Entity Object Usage and an Expert-Mode Query	352
Representing Relationships Between Query Result Sets	357
View Link SQL	357
View Link Definition Cardinality	359
View Link Definition Directionality and Accessor Attributes	360
Aggregating Data for Applications	361
Hands-on Practice: Create View Link and Application Module Definitions	362
I. Create View Link Definitions	362
II. Create and Test an Application Module Definition	364
12 Exposing ADF BC to Applications	369
ADF Model Layer Architecture	370
Data Controls	370
Binding Containers	371
Creating Data Controls and Bindings	372
The Data Control Palette	372
The UI Model Tab	374
Exposing a View Object Instance to an ADF Application	375
Ranges	375
Using Multiple Iterator Bindings	376
Accessing a Range of Data	376
Navigating Through Collections	378
Moving the Current Row Pointer	378
Scrolling the Range	379
Key Objects	380
Creating and Deleting Rows	381
Accessing Individual Values	381
Hands-on Practice: Create a Master-Detail JSP Application	381
I. Create a Read-Only Form with Navigation	382
II. Create a UI Table to Display Details	387

Selecting from a List	393
Navigation Mode	393
Enumeration Mode	394
LOV Mode	395
Managing Transactions	396
Hands-on Practice: Refine the JSP Application	397
I. Refine the Master-Detail Form	397
II. Create an Edit Form	400
13 Creating Custom Service Methods	407
Overview of Data Model Component Classes	409
Overview of Application Module Classes	409
Overview of View Classes	411
Custom Service Method Basics	412
Exposing and Accessing Service Methods	412
Finding View Object Instances in the Data Model	413
Finding View Object Instances in a Nested Application Module Instance	413
Retrieving View Rows	414
Stepping Through a View Cache	415
Row Keys	415
Manipulating Data	417
Reading and Changing Attribute Values	417
Creating and Deleting Rows	417
Restricting a View Object Instance's Cache	418
Using setWhereClause()	418
Parameterized WHERE Clauses	419
View Criteria Objects	420
Re-Executing the Query	421
Using View Link Definitions in Service Methods	421
Hands-on Practice: Create and Invoke Service Methods	422
I. Turn Off Batch Mode	423
II. Choose View Classes	423
III. Create a View Object Method Stub and a Service Method	425
IV. Access the Service Method from a JSP Application	427
V. Add Navigation Code to the View Object Method	431
VI. Add Code to Filter Data	432
VII. Add Code to Change Data	434
Dynamically Creating Master-Detail Relationships	436
Hands-on Practice: Create and Invoke Service Methods to Maintain a Dynamic Master-Detail Relationship	437
I. Remove a View Link Instance from an Application Module Definition	438
II. Create Service Methods to Maintain a Dynamic Master-Detail Relationship	438
III. Use the Dynamic Master-Detail Relationship	440
IV. Test the Service Methods	443

14 Business Service Technology Alternatives	445
Enterprise JavaBeans Technology	446
EJB Technology and the Business Domain	448
EJB Technology and the Data Model	450
EJB Technology and the ADF Model Layer	454
Hands-on Practice: Build a Simple EJB Application	457
I. Create an Application Workspace and EJB Diagram	458
II. Create the Business Domain	459
III. Create the Data Model	460
IV. Create Data Control Definition Files	467
V. Create a Master-Detail Form	468
TopLink Technology	470
TopLink POJO and the Business Domain	470
TopLink POJO and the Data Model	473
TopLink Technology and the ADF Model Layer	474
Hands-on Practice: Build a Simple TopLink Application	475
I. Create an Application Workspace	476
II. Create Java Classes	478
III. Create Descriptors and Direct Mappings	480
IV. Create a Relationship Mapping	484
V. Create a TopLink Query	485
VI. Create Data Control Definition Files	486
VII. Create a Master-Detail Form	486
Web Services	488
WSDL, SOAP, and Web Service Stubs	488
Web Services and the ADF Model Layer	489
Hands-on Practice: Create an Application That Uses a Web Service	489
I. Register to Use the Google Web APIs	490
II. Create an Application Workspace	490
III. Create a Web Service Stub	491
IV. Create a Wrapper Class for the Stub	492
V. Create Data Control Definition Files	493
VI. Create a Search Application	494

PART III

Java Client and Web Applications

15 Creating Java Client Applications	501
The JDeveloper IDE for Java Client Development	502
Building Java Client Applications	503
Directory Organization for a Java Client Application	504
Java Client Architecture Decisions	504
How Many Independent Programs Will You Use?	505
How Many Directories, Workspaces, and Projects Will You Create?	505
Where Will the Model and Data Validation Take Place?	505

What Type of Container Layout Should Be Used?	506
How Many Packages Will You Create and How Will You Name Them?	506
Other Issues	506
ADF JClient	507
Swing UI Architecture	507
Developing a Client Data Model	508
Binding Swing Components to ADF BC	508
Binding Panels	509
Working with Swing Components in JDeveloper	510
The Data Control Palette	510
The Component Palette	511
Using Swing Components	511
Adding Swing Components to a Program	512
Categories of Swing Components	512
Container Objects	513
Modifying Swing Components	517
Defining Events	517
Getting the Right Information	518
Hands-on Practice: Create a Tabbed User Interface Application	520
I. Create the ADF BC Project	520
II. Create the Java Application Project	521
III. Create a Three-Tab User Interface	524
Hands-on Practice: Customize the Component Palette and Create a JavaBean	532
I. Create and Deploy a JavaBean	533
II. Create a Library for the JavaBean	535
III. Add a Component Palette Page and Add the Custom JavaBean	536
IV. Test the Custom Component	539
16 Layout Managers	543
How Does This Work in a J2EE Web Application?	545
Layout Manager and Container Concepts	545
Laying Out a User Interface	546
The Default Layout Manager	548
Setting Layout Manager Properties	548
Java Visual Editor Tools	549
Layout Managers in JDeveloper	550
Overview of the Layout Managers	551
A Word About the “null” Layout	551
BorderLayout	551
BoxLayout2	554
CardLayout	556
FlowLayout	557
GridBagLayout	559
GridLayout	566
OverlayLayout2	568
PaneLayout	569

VerticalFlowLayout	571
XYLayout	572
Layout Manager Usage	573
Multiple Layouts	574
Hands-on Practice: Work with Layouts	577
I. Set up an Application Workspace and Two Projects	577
II. Use the BorderLayout Manager	579
III. Use the GridBagLayout Manager	583
17 Working with Struts	591
Struts Architecture	592
Struts Elements	593
ApplicationResources.properties Files	594
The struts-config.xml File	595
View-Level Struts Tag Libraries	597
JDeveloper's Implementation of Struts	598
Working in the Page Flow Diagram	599
Struts Elements in JDeveloper	600
Data Controls	604
Introduction to the Hands-on Practices	606
Hands-on Practice: Create a Simple Struts Application	607
I. Create the Application Workspace and ADF BC Project	607
II. Create Browse and Edit JSP Data Pages	608
III. Add the Ability to Save Edits	613
IV. Add the Ability to Create Records	615
V. Add Delete Functionality to the Application	617
VI. Add a Logic Tag to Highlight the Current Record	619
VII. Shade Every Other Row	622
VIII. Test for a Value in the Table	623
Hands-on Practice: Create a Struts Application with Data Actions for Custom Logic	625
I. Create Another ViewController Project	625
II. Create a Data Action and a JSP Data Page	627
III. Add Edit and Update Functionality	631
18 Working with JSP Pages	637
JSP Development Requirements	638
Required Language Skills	639
Understanding JSP Compilation and Runtime	639
Additional Information Sources	640
JSP Application Development in JDeveloper	641
JSP/HTML Visual Editor	641
Code Editor	644
Structure Window	645
Page Flow Diagram	646
Property Inspector	647

Component Palette	648
Data Control Palette	650
Steps for Creating a Struts JSP Application	657
Some ADF BC JSP Coding Techniques	657
Find Mode	657
ADF Business Component Properties	658
Using Control Hints for Labeling and Formatting	659
Hands-on Practice: Build JSP Query and Results Pages	663
I. Create the Application Workspace and Business Services	664
II. Create the JSP Pages and Controller Actions	665
III. Modify the Data Elements	671
IV. Modify the Visual Aspects	680
19 Working with ADF UIX Pages	685
ADF UIX Overview	686
UIX Features	687
UIX Page Design Structure	689
UIX Application Development in JDeveloper	693
Wizards and File Creation Dialogs	693
UIX Visual Editor	694
UIX Preview	694
XML Editor	694
Structure Window	694
Page Flow Diagram	695
Property Inspector	695
Component Palette	696
Data Control Palette	697
Steps for Creating a Struts UIX Application	697
Hands-on Practice: Build a UIX Application	699
I. Prepare the Projects	699
II. Create the Browse Page Using a Wizard	700
III. Add Data and Action Components to the Browse Page	703
IV. Create and Link the Edit Page	706
V. Create and Link the Search Page	708
VI. Modify the Look-And-Feel	711
 PART IV Appendixes 	
A Other Resources	715
Books	716
Websites	717
B Java Client User Interface Components	721
AWT	722
Code Snippets	724

JClient Controls	726
Swing	728
Swing Containers	730
C Overview of HTML, JavaScript, and Cascading Style Sheets	733
HTML	734
Editing HTML	734
HTML Tags	734
Sample HTML Code	737
JavaScript in HTML	739
Cascading Style Sheets	740
Building a Cascading Style Sheet	741
Using a Cascading Style Sheet	742
D Overview of JSP, JSTL, and EL Tags	745
Basic JSP Tags	746
Where to Put the Code?	747
Beginning and Ending Tags	747
Processing of Standard Tags	747
JSP Standard Tag Library	759
JSTL Example	760
Tag Libraries	761
Expression Language	763
An Annotated Example	765
Index	769