core
JavaServer™ Faces
Second Edition

David Geary
Cay Horstmann
Contents

Preface xv
Acknowledgments xix

1 GETTING STARTED 2
   Why JavaServer Faces? 3
   Software Installation 4
   A Simple Example 6
      Ingredients 8
      Directory Structure 9
      Build Instructions 10
   Sample Application Analysis 12
      Beans 12
      JSF Pages 13
      Navigation 16
      Servlet Configuration 18
      The Welcome File 20
   Development Environments for JSF 21
      Integrated Development Environments 21
Contents

Visual Builder Tools 22
Automation of the Build Process with Ant 24
JSF Framework Services 28
Behind the Scenes 30
   Rendering Pages 30
   Decoding Requests 32
   The Life Cycle 33

2 MANAGED BEANS 36
Definition of a Bean 37
   Bean Properties 39
   Value Expressions 41
Message Bundles 42
   Messages with Variable Parts 44
   Setting the Application Locale 45
A Sample Application 46
Backing Beans 53
Bean Scopes 54
   Session Scope 54
   Application Scope 56
   Request Scope 56
   Life Cycle Annotations 57
Configuring Beans 57
   Setting Property Values 58
   Initializing Lists and Maps 59
   Chaining Bean Definitions 61
   String Conversions 62
The Syntax of Value Expressions 64
   Using Brackets 64
   Map and List Expressions 65
   Resolving the Initial Term 66
   Composite Expressions 68
   Method Expressions 69
3 NAVIGATION 70
  Static Navigation 71
  Dynamic Navigation 73
  Advanced Navigation Issues 84
    Redirection 85
    Wildcards 86
    Using from-action 87
    The Navigation Algorithm 87

4 STANDARD JSF TAGS 90
  An Overview of the JSF Core Tags 92
  An Overview of the JSF HTML Tags 94
    Common Attributes 97
  Forms 103
    Form Elements and JavaScript 105
  Text Fields and Text Areas 108
    Hidden Fields 112
    Using Text Fields and Text Areas 112
    Displaying Text and Images 116
  Buttons and Links 119
    Using Command Buttons 121
    Using Command Links 125
  Selection Tags 130
    Checkboxes and Radio Buttons 133
    Menus and Listboxes 136
    Items 138
  Messages 157
  Panels 163

5 DATA TABLES 170
  The Data Table Tag—h:dataTable 171
  A Simple Table 173
    h:dataTable Attributes 176
Contents

h:column Attributes 178
Headers, Footers, and Captions 178
JSF Components 182
Editing Table Cells 186
Styles 189
  Styles by Column 190
  Styles by Row 190
Database Tables 191
  JSTL Result Versus Result Sets 197
Table Models 197
  Editing Table Models 198
  Sorting and Filtering 203
Scrolling Techniques 214
  Scrolling with a Scrollbar 215
  Scrolling with Pager Widgets 216

6 CONVERSION AND VALIDATION 218
Overview of the Conversion and Validation Process 219
Using Standard Converters 221
  Conversion of Numbers and Dates 221
  Conversion Errors 225
  A Complete Converter Example 230
Using Standard Validators 233
  Validating String Lengths and Numeric Ranges 234
  Checking for Required Values 235
  Displaying Validation Errors 236
  Bypassing Validation 237
  A Complete Validation Example 238
Programming with Custom Converters and Validators 240
  Implementing Custom Converter Classes 240
  Implementing Custom Validator Classes 254
  Registering Custom Validators 257
  Validating with Bean Methods 259
## Contents

Supplying Attributes to Converters 260  
Validating Relationships Between Multiple Components 260

### 7 EVENT HANDLING 266

- Life Cycle Events 268  
- Value Change Events 269  
- Action Events 275  
- Event Listener Tags 285  
  - The `f:actionListener` and `f:valueChangeListener` Tags 285  
- Immediate Components 287  
  - Using Immediate Input Components 288  
  - Using Immediate Command Components 290  
- Passing Data from the UI to the Server 291  
  - The `f:param` Tag 292  
  - The `f:attribute` Tag 292  
  - The `f:setPropertyActionListener` Tag 293  
- Phase Events 295  
- Putting It All Together 303

### 8 SUBVIEWS AND TILES 314

- Common Layouts 315  
- A Book Viewer and a Library 316  
- The Book Viewer 318  
  - Monolithic JSF Pages 320  
  - Common Content Inclusion 326  
  - Content Inclusion in JSP-Based Applications 326  
  - JSF-Specific Considerations 327  
- Content Inclusion in the Book Viewer 328  
  - Looking at Tiles 331  
  - Installing Tiles 332  
  - Using Tiles with the Book Viewer 333  
  - Parameterizing Tiles 334
Contents

Extending Tiles 335
The Library 339
Nested Tiles 339
Tile Controllers 341

CUSTOM COMPONENTS, CONVERTERS,
AND VALIDATORS 354

Classes for Implementing Custom Components 356
Tags and Components 359
The Custom Component Developer's Toolbox 360
Encoding: Generating Markup 362
Decoding: Processing Request Values 366
Using Converters 369
Implementing Custom Component Tags 372
The TLD File 372
The Tag Handler Class 376
The Spinner Application 379
Defining Tag Handlers in JSF 1.1 383
Revisiting the Spinner 387
Using an External Renderer 387
Calling Converters from External Renderers 393
Supporting Value Change Listeners 394
Supporting Method Expressions 396
The Sample Application 397
Encoding JavaScript to Avoid Server Roundtrips 404
Using Child Components and Facets 408
Processing SelectItem Children 411
Processing Facets 412
Encoding CSS Styles 413
Using Hidden Fields 415
Saving and Restoring State 415
Firing Action Events 418
Using the Tabbed Pane 425
Implementing Custom Converters and Validators 432
  Custom Converter Tags 432
  Custom Validator Tags 441

10 EXTERNAL SERVICES 450
  Database Access with JDBC 451
    Issuing SQL Statements 451
    Connection Management 453
    Plugging Connection Leaks 453
    Using Prepared Statements 455
  Configuring a Data Source 457
    Configuring a Database Resource in GlassFish 457
    Configuring a Database Resource in Tomcat 459
    Accessing a Container-Managed Resource 462
    A Complete Database Example 464
  An Introduction to LDAP 473
    LDAP Directories 473
    Configuring an LDAP Server 474
    Accessing LDAP Directory Information 479
  Managing Configuration Information 483
    Configuring a Bean 484
    Configuring the External Context 486
    Configuring a Container-Managed Resource 487
    Creating an LDAP Application 491
  Container-Managed Authentication and Authorization 505
  Using Web Services 516

11 AJAX 528
  Ajax Fundamentals 530
  JavaScript Libraries 533
    The Prototype Library 533
    The Fade Anything Technique Library 534
  Form Completion 534
Contents

Realtime Validation  537
Propagating Client-Side View State   542
Direct Web Remoting   543
Ajax Components   546
    Hybrid Components   546
    Keeping JavaScript Out of Renderers   551
    Transmitting JSP Tag Attributes to JavaScript Code   552
Ajax4jsf   554
    Implementing Form Completion with Ajax4jsf   555
    Implementing Realtime Validation with Ajax4jsf   558

12 OPEN SOURCE   570
Web Flow—Shale   572
    Dialog Configuration   576
    Entering a Dialog   576
    Dialog Navigation   577
    Dialog Scope   578
    Dialog Context Sensitivity   580
    Subdialogs   583
Alternate View Technologies—Facelets   585
    XHTML Views   585
    Replacing Markup with JSF Components:
        The jsfc Attribute   587
    Using JSF Tags   590
    Page Composition with Templates   592
    Facelets Custom Tags   594
EJB Integration—Seam   596
    An Address Book   596
    Configuration   601
    Entity Beans   601
    Stateful Session Beans   603
    JSF DataModel Integration   606
    Conversation Scope   608
13 HOW DO I . . . 610
Web User Interface Design  611
   How do I find more components?  611
   How do I support file uploads?  614
   How do I show an image map?  623
   How do I include an applet in my page?  625
   How do I produce binary data in a JSF page?  627
   How do I show a large data set, one page at a time?  638
   How do I generate a pop-up window?  644
   How do I selectively show and hide components?  653
   How do I customize error pages?  654
Validation  658
   How do I write my own client-side validation tag?  658
   How do I use the Shale Validator for client-side validation?  666
   How do I validate relationships between components?  669
Programming  669
   How do I use JSF with Eclipse?  669
   How do I locate a configuration file?  673
   How can a JSF component access resources from a JAR file?  673
   How do I package a set of tags into a JAR file?  677
   How do I get the form ID for generating document.forms[id] in JavaScript?  678
   How do I make a JavaScript function appear only once per page?  679
   How do I carry out initialization or cleanup work?  679
   How do I store a managed bean longer than request scope but shorter than session scope?  680
   How do I extend the JSF expression language?  681
Debugging and Logging  684
   How do I decipher a stack trace?  684
   How do I avoid the “stack trace from hell”?  687
Contents

How do I “hot deploy” my application? 688
How do I comment out a part of a JSF page? 689
How do I find the logs? 689
How do I find out what parameters my page received? 691
How do I turn on logging of the JSF container? 692
How do I debug a stuck page? 696
How do I find the library source? 697

Index 699
When we heard about JavaServer Faces (JSF) at the 2002 JavaOne conference, we were very excited. Both of us had extensive experience with client-side Java programming, and had lived to tell the tale—David in Graphic Java™, and Cay in Core Java™, both published by Sun Microsystems Press. When we first tried web programming with servlets and JavaServer Pages (JSP), we found it to be rather unintuitive and tedious. JavaServer Faces promised to put a friendly face in front of a web application, allowing programmers to think about text fields and menus instead of fretting over page flips and request parameters. Each of us proposed a book project to the publisher, who promptly suggested that we should jointly write the Sun Microsystems Press book on this technology.

It took the JSF Expert Group (of which David was a member) until 2004 to release the JSF 1.0 specification and reference implementation. A bug fix 1.1 release emerged shortly afterwards, and an incremental 1.2 release added a number of cleanups and convenience features in 2006.

JSF is now the preeminent server-side Java web framework, and it has fulfilled most of its promises. You really can design web user interfaces by putting components on a form and linking them to Java objects, without having to mix code and markup. A strong point of JSF is its extensible component model, and a large number of third-party components have become available. The flexible design of the framework has allowed it to grow well and accommodate new technologies such as Ajax. The framework was designed for tool support, and
usable drag-and-drop GUI builders have finally emerged. And finally, unlike competing technologies that let you tumble down a deep cliff once you step beyond the glitz, JSF supports the hard stuff—separation of presentation and business logic, navigation, connections with external services, and configuration management.

We are still excited about JSF, and we hope you will share this excitement when you learn how this technology makes you a more effective web application developer.

About This Book

This book is suitable for web developers whose main focus is user interface design, as well as for programmers who implement reusable components for web applications. This is in stark contrast to the official JSF specification, a dense and pompously worded document whose principal audience is framework implementors, as well as long-suffering book authors.

The first half of the book, extending through Chapter 6, focuses on the JSF tags. These tags are similar to HTML form tags. They are the basic building blocks for JSF user interfaces. No programming is required for use of the tags. We assume only basic HTML skills for web pages and standard Java programming for the business logic.

The first part of the book covers these topics:

- Setting up your programming environment (Chapter 1)
- Connecting JSF tags to application logic (Chapter 2)
- Navigating between pages (Chapter 3)
- Using the standard JSF tags (Chapters 4 and 5)
- Converting and validating input (Chapter 6)

Starting with Chapter 7, we begin JSF programming in earnest. You will learn how to perform advanced tasks, and how to extend the JSF framework. Here are the main topics of the second part:

- Event handling (Chapter 7)
- Including common content among multiple pages (Chapter 8)
- Implementing custom components, converters, and validators (Chapter 9)
- Connecting to databases and other external services (Chapter 10)
- Ajax (Chapter 11)
- Open source technologies, with a focus on Facelets, Seam, and Shale (Chapter 12)
We end the book with a chapter that aims to answer common questions of the form “How do I . . . ?” (see Chapter 13). We encourage you to have a peek at that chapter as soon as you become comfortable with the basics of JSF. There are helpful notes on debugging and logging, and we also give you implementation details and working code for features that are missing from JSF, such as file uploads, pop-up menus, and a pager component for long tables.

JSF is built on top of servlets and JSP, but from the point of view of the JSF developer, these technologies merely form the low-level plumbing. While it can’t hurt to be familiar with other web technologies such as servlets, JSP, or Struts, we do not assume any such knowledge.

**Required Software**

All software that you need for this book is freely available. You need the Java Software Development Kit from Sun Microsystems and an application server that supports JSF, such as the excellent open source GlassFish project. The software runs identically on Linux, Mac OS X, Solaris, and Windows. We used Java 5 and GlassFish on both Linux and Mac OS X to develop the code examples in the book.

If you are looking for a development environment that supports JSF development, we can heartily recommend the freely available NetBeans IDE. Good JSF support for Eclipse is available from several vendors that sell Eclipse enhancements.

**Web Support**

The web page for this book is http://corejsf.com. It contains

- The source code for all examples in this book
- Useful reference material that we felt is more effective in browseable form than in print
- A list of known errors in the book and the code
- A form for submitting corrections and suggestions
First and foremost, we’d like to thank Greg Doench, our editor at Prentice Hall, who has shepherded us through this project, never losing his nerve in spite of numerous delays and complications.

We very much appreciate our reviewers for both editions who have done a splendid job, finding errors and suggesting improvements in various drafts of the manuscript. They are:

- Gail Anderson, Anderson Software Group, Inc.
- Larry Brown, LMBrown.com, Inc.
- Frank Cohen, PushToTest
- Brian Goetz, Sun Microsystems, Inc.
- Rob Gordon, Crooked Furrow Farm
- Marty Hall, author of Core Java Servlets and JavaServer Pages
- Charlie Hunt, Sun Microsystems, Inc.
- Jeff Langr, Langr Software Solutions
- Bill Lewis, Tufts University
- Jeff Markham, Markham Software Company
- Angus McIntyre, IBM Corporation
- John Muchow, author of Core J2ME
- Dan Shellman, BearingPoint
Acknowledgments

- Sergei Smirnov, principal architect of Exadel JSF Studio
- Roman Smolgovsky, Flytecomm
- Stephen Stelting, Sun Microsystems, Inc.
- Christopher Taylor, Nanshu Densetsu
- Kim Topley, Keyboard Edge Limited
- Michael Yuan, co-author of *JBoss Seam: Simplicity and Power Beyond Java EE*

Finally, thanks to our families and friends who have supported us through this project and who share our relief that it is finally completed.