

The Definitive Guide to Apache MyFaces and Facelets



Zubin Wadia, Martin Marinschek,
Hazem Saleh, and Dennis Byrne

Contributing Authors: Bruno Aranda, Mario Ivankovits, Cagatay Civici,
Arvid Hülsebus, Detlef Bartetzko, and Allan Lykke Christensen

Apress®

The Definitive Guide to Apache MyFaces and Facelets

Copyright © 2008 by Zubin Wadia, Martin Marinschek, Hazem Saleh, Dennis Byrne, Mario Ivankovits, Gagatay Civici, Arvid Hülsebus, Detlef Bartetzko, Bruno Aranda, Allan Lykke Christensen

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-59059-737-8

ISBN-10 (pbk): 1-59059-737-0

ISBN-13 (electronic): 978-1-4302-0344-5

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editor: Steve Anglin

Technical Reviewer: Dr. Sarang Poornachandra

Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell,

Jonathan Gennick, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann,

Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Project Manager: Sofia Marchant

Copy Editor: Heather Lang

Associate Production Director: Kari Brooks-Copony

Production Editor: Kelly Winkquist

Compositor: Linda Weidemann, Wolf Creek Press

Proofreader: Martha Whitt

Indexer: John Collin

Artist: April Milne

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit <http://www.apress.com>.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at <http://www.apress.com/info/bulksales>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at <http://www.apress.com>.

*My parents, Rustom and Avan,
whose guidance and encouragement mean the world.
—Zubin Wadia*

*For the prophet Muhammad,
from whom I learnt all the good things in my life.
—Hazem Saleh*

*Ainhoa, my wife,
whose support and patience made this book possible.
—Bruno Aranda*

Contents at a Glance

About the Authors	xiii
About the Technical Reviewer	xv
Acknowledgments	xvii
Introduction	xix
CHAPTER 1 An Introduction to JavaServer Faces	1
CHAPTER 2 The Tomahawk Project	21
CHAPTER 3 Facelets	67
CHAPTER 4 The Trinidad Project	111
CHAPTER 5 Conversations and Scoping with Orchestra	165
CHAPTER 6 Layouts and Themes with Tobago	199
CHAPTER 7 Antipatterns and Pitfalls	229
APPENDIX A Facelets Tag Reference	271
APPENDIX B View State Encryption	285
APPENDIX C Custom Dependency Injection	287
INDEX	289

Contents

About the Authors	xiii
About the Technical Reviewer	xv
Acknowledgments	xvii
Introduction	xix
CHAPTER 1 An Introduction to JavaServer Faces	1
The Evolution of Java Web Frameworks	1
Servlets	2
JavaServer Pages	3
Struts and Web Development Patterns	4
A Brief History of JSF	4
JSF Features	5
The JSF Life Cycle	6
Life Cycle Phase 1: Restore View	7
Life Cycle Phase 2: Apply Request Values	7
Life Cycle Phase 3: Process Validations	7
Life Cycle Phase 4: Update Model Values	8
Life Cycle Phase 5: Invoke Application	8
Life Cycle Phase 6: Render Response	8
Moving from Struts Frameworks to JSF	8
JSF Shortcomings	10
Apache MyFaces	10
MyFaces Installation	11
Configuring MyFaces	12
Writing Your First MyFaces Application	13
Summary	19
CHAPTER 2 The Tomahawk Project	21
Tomahawk Setup	21
Tomahawk Features	22
The forceId Attribute	23
The displayValueOnly Attribute	23
Tomahawk's Major Components	24

Component Examples	25
The Calendar Component	25
The DataScroller Component	29
The DataList Component	34
The NewspaperTable Component	38
The JSCookMenu Component	40
The TabbedPane Component	42
The ValidateRegExpr Component	44
The Tree2 Component	47
The Schedule Component	52
The SubForm Component	57
Putting It All Together	58
Summary	65

CHAPTER 3	Facelets	67
	The Gap Between JSP and JSF	67
	Why Use Facelets?	68
	Creating an Application with Facelets	68
	Downloading Facelets	69
	Adding Dependencies	70
	Creating a Project Structure	70
	Configuring the Web Descriptor (web.xml)	71
	Faces Descriptor (faces-config.xml)	72
	Creating JSF Views	72
	Migrating an Existing Application from JSP to Facelets	79
	Unified Expression Language	80
	Inline Text	80
	Tag Libraries	81
	Other Tag Libraries	81
	Creating a Tag Library	81
	Functions	83
	The jsfc Attribute	85
	Facelets Templating and Template Clients	85
	Creating Composition Components	86
	Creating the Custom inputTextLabeled Component	86
	Creating the simpleColumn Custom Component	89
	Creating the scrollableDataTable Custom Component	91
	Creating the editableColumn Custom Component	96
	Reusing the Custom Tag Library	98

Extending Facelets	99
Facelets Architecture	99
Custom Tag Development	100
Using Metatags	103
Extending the ViewHandler	110
Summary	110
CHAPTER 4 The Trinidad Project	111
Trinidad Setup	111
Why Use Trinidad?	113
Extended JSF Capabilities	113
RequestContext	114
pageFlowScope	114
Partial Page Rendering (PPR)	121
Dialog Framework	130
Client-Side Conversion and Validation	140
Components	150
Chart	150
Table	153
Skinning	157
Creating a Custom Skin	158
Tuning and Configuration	161
Tuning Trinidad	163
Summary	164
CHAPTER 5 Conversations and Scoping with Orchestra	165
Why Use Orchestra?	165
Creating an Application with Orchestra	166
Dependencies	166
Configuring the Web Descriptor (web.xml)	167
Configuring the Spring Framework	168
Conversations	171
Conversation Examples	172
Conversation Lifetime	174
Managing Independent Windows	175
The Role of Spring	176
Configuring Conversation-Scoped Managed Beans	177
Conversation-Related JSF Tags	179

Managing Events with ViewController	181
Mapping Views to Beans	182
Finding the Life Cycle Methods	182
Mapping via Annotations	182
Customizing the ViewController	183
Requiring Conversations	184
Managing Persistence in Orchestra	185
Quick Overview of the Java Persistence API	185
Using Orchestra's Persistence Support	186
Using Persistence with Annotations	187
Using Persistence Without Annotations	188
Transaction Management	188
Using Non-JPA Persistence Implementations	189
JSF Converters and Persistence	190
The DynaForm Component	191
Orchestra's Architecture	194
ConversationManager	194
ConversationContext	194
FrameworkAdapter	195
RequestParameterProvider	195
RedirectTrackerNavigationHandler	195
UrlParameterNavigationHandler	196
Application Programming Interface	196
Summary	198
CHAPTER 6 Layouts and Themes with Tobago	199
Trying Out Tobago	199
Online Demo	200
Creating an Application with Tobago	200
Dependencies	200
Configuring the Web Descriptor (web.xml)	201
Configuring Tobago (tobago-config.xml)	202
Creating a Basic Tobago Page	202

Controls	203
Basic Controls	203
Sheet Control	206
TabGroup Control	209
Menu Control	209
Toolbar Control	210
Pop-ups	211
File Upload	211
Layout Management	212
Themes	214
Resource Management	216
Adding Markup	217
Creating Your Own Tobago Control	218
The UIComponent	219
The Renderer	220
The Tag and the Tag Declaration	223
Building and Using the Control	224
Security	224
Virtual Forms	226
Partial Rendering	226
Advanced Assembly	227
Summary	228
CHAPTER 7 Antipatterns and Pitfalls	229
N Plus One	229
The Map Trick	233
The Déjà Vu PhaseListener	234
XML Hell++	235
The Myth of Static Typing	236
Thread Safety	239
Facelets Migration Challenge: Tags with Behavior	240
The Law of Demeter	241
Master-Detail	242
Vendor Lock-In	250

AJAX 101	251
Code to Interface	260
View State Encryption	262
Portlet Issues	266
The Validating Setter	267
Summary	269
APPENDIX A Facelets Tag Reference	271
<ui:component/>	271
<ui:composition/>	272
<ui:debug/>	273
<ui:decorate/>	273
<ui:define/>	276
<ui:fragment/>	277
<ui:include/>	277
<ui:insert/>	278
<ui:param/>	280
<ui:remove/>	282
<ui:repeat/>	283
APPENDIX B View State Encryption	285
APPENDIX C Custom Dependency Injection	287
INDEX	289

About the Authors



■ **ZUBIN WADIA** has seven years of experience in JEE and .NET technologies. His experiences with enterprise content management, workflow, imaging, compression, and security allow him to appreciate a powerful framework that offers a high degree of flexibility and efficiency. In 2005, Zubin designed and deployed an award-winning solution for a criminal justice agency in the government space, leveraging next-generation technologies such as Ajax, Hibernate, Spring, BPEL, and, of course, MyFaces. His current focus is on architecting an enterprisewide contract management platform for the New York City Department of Health. Zubin is CTO at ImageWork Technologies, a leading government systems integrator on the East Coast.



■ **MARTIN MARINSCHEK** has eight years of experience in Java EE and open source technologies. He is a MyFaces PMC member and a popular speaker at major open source events such as JavaPolis and ApacheCon. Martin brings dynamism, experience, and innovation to the MyFaces project. Martin is a CEO of Irian, a fast-growing consulting company focused on providing JSF solutions to customers.



■ **HAZEM SALEH** has four years of experience in Java EE and open source technologies and is a MyFaces committer. He is the initiator of many components in the MyFaces Tomahawk sandbox such as CAPTCHA, pdfExport, media, and passwordStrength. He is also a yui4jsf committer. He is now working at IBM Egypt as a staff engineer.



■ **DENNIS BYRNE** works for ThoughtWorks, a global consultancy with a focus on end-to-end agile software development of mission-critical systems. Dennis is a committer and PMC member for Apache MyFaces. He is also an experienced public speaker and a committer for JBoss JSFUnit.

About the Technical Reviewer



■ **POORNACHANDRA SARANG, PHD**, in his long tenure of over 20 years, has worked in various capacities in the IT industry. Dr. Sarang provides consulting and training in enterprise architecting, solution architecting, design, and development to a large number of clients worldwide. He has been a consultant to Sun Microsystems for several years, and his recent engagements include Director, Architecture for Kynetia (a Spanish-origin MNC). He has been a recipient of Microsoft's Most Valuable Professional (MVP) award for two consecutive years. He has spoken in several international conferences on Java, CORBA, XML, and .NET organized by O'Reilly, SYS-CON, WROX, SUN, and Microsoft in various countries such as India, the United States, the United Kingdom, Switzerland, Singapore, and so on. He has been invited to deliver keynote speeches at highly acclaimed Microsoft Architect Summits and several other prestigious events. He has several research papers, journal articles, and books to his credit.

Dr. Sarang has been a visiting professor of computer engineering at University of Notre Dame and currently holds an adjunct faculty position at the Department of Computer Science at the University of Mumbai, where he teaches postgraduate courses, provides project guidance to postgraduate students, and guides PhD students. His current research interests include distributed systems, mobile computing, and algorithm development. He may be reached at profsarang@gmail.com.

Acknowledgments

This book would not have been possible without the MyFaces project founders—Manfred Geiler and Thomas Spiegl. Additionally, Jacob Hookom’s seminal work with Facelets was a prime driver to increasing JSF adoption in the Java developer community. Without their vision, this comprehensive book would not have an audience.

Additionally, we would like to acknowledge the contributions of Simon Kitching, who is a regular committer to MyFaces and the Orchestra projects. His insight and attention to detail have been valuable to the continuing growth and applicability of Orchestra. Arash Rajaeyan also provided valuable input, contributions, and insight into Tomahawk and the IDE setup. We are grateful for their inputs.

Finally, we would like to thank the MyFaces and JSF community as a whole for their tireless support and contributions to a variety of subprojects. Without this anarchic collaboration, MyFaces would cease to be competitively differentiated against the Sun Reference Implementation.

Introduction

In the four years that the Sun, IBM, and MyFaces implementations have been available, the MyFaces project has arguably been the front-runner when it comes to innovation and dexterity. It also is the most responsive to user issues and has a vibrant community that ensures progression (you can always find free support on the mailing lists). With JavaServer Faces (JSF) 2.0 around the corner, preparation is already under way as the committers quickly review the early access draft and plan for MyFaces 2.0.

This “anarchic collaboration,” a term coined by Roy Fielding, is the essence of the Apache MyFaces project: many talented people share information with a single-minded objective to make JSF a pleasurable developer experience for all. The Tomahawk, Trinidad, Tobago, Orchestra, and MyFaces implementations have been downloaded over a million times. Before the Sun JSF 1.2 implementation was released, we could safely say the MyFaces implementation was the only reliable way to build production JSF applications. Even today, many developers are happy with simply using MyFaces 1.1.x and Facelets to form a potent combination.

This book assumes you are interested in JSF and component-oriented development. It takes you through a journey beginning with the unique aspects of JSF, introduces you to the MyFaces specification, and continues with building your first application. Once you’ve mastered that, this book advances your skill set further by taking you into the world of the Tomahawk component library. Here, you learn the power of componentization in JSF (you will be able to build a complex application with a few lines of code) and see the advantages of having a vibrant community like MyFaces continually developing new components into the sandbox like CAPTCHA-, media-, exporter-, and AJAX-enabled UI widgets.

Next, we take you into one of the most important chapters in the book—learning to leverage Facelets’s powerful view-definition capabilities within your JSF application. After absorbing the material in this chapter, you will be able to write JSF view tiers with greater efficiency and yield more performance out of your applications, as compared to similar applications leveraging classic JSP. The patterns we encourage in this chapter are a result of our more than ten years of cumulative experience with JSF across two of the major Reference Implementations: Sun and MyFaces.

The second half of this book focuses on smaller but equally critical projects to the future of MyFaces: Trinidad, Tobago, and Orchestra. Trinidad and Tobago offer high-quality component libraries that are particularly adept at custom rendition, themes, and layouts. Trinidad’s presence is also a testament to the commercial support of MyFaces

by Oracle for its Application Development Framework (ADF). Although a relatively young project, Orchestra offers powerful functionality like conversation scope for persistence-rich JSF applications.

The final chapter weaves together all your newfound knowledge and shows how to avoid some of the common antipatterns new JSF developers often end up pursuing. JSF is a complex specification augmented by a significant number of support frameworks to assist with development. We make sure the details in this book allow you to make educated decisions on which ones to leverage.

Finally, be sure to take advantage of the book's modular nature. We tried to strike a natural balance between this book being a *guide* and a *reference*. Refreshing your memory on a particular topic is extremely easy because of the atomic nature of each chapter. Enjoy!