

PROFESSIONAL JSP

Using JavaServer Pages, Servlets, EJB, JNDI, JDBC, XML, XSLT, and WML to create dynamic and customizable web content



Karl Avedal, Danny Ayers, Timothy Briggs, Carl Burnham, Ari Halberstadt, Ray Haynes, Peter Henderson, Mac Holden, Sing Li, Dan Malks, Tom Myers, Alexander Nakhimovsky, Stéphane Osmont, Grant Palmer, John Timney, Sameer Tyagi, Geert Van Damme, Mark Wilcox, Steve Wilkinson, Stefan Zeiger, John Zukowski

Professional JSP

Karl Avedal
Danny Ayers
Timothy Briggs
Carl Burnham
Ari Halberstadt
Ray Haynes
Peter Henderson
Mac Holden
Sing Li
Dan Malks
Tom Myers
Alexander Nakhimovsky
Stéphane Osmont
Grant Palmer
John Timney
Sameer Tyagi
Geert Van Damme
Mark Wilcox
Steve Wilkinson
Stefan Zeiger
John Zukowski

Wrox Press Ltd. ®

Professional JSP

© 2000 Wrox Press

Chapter 14: © 2000 Ari Halberstadt

All rights reserved. No part of this book may be reproduced, stored in a retrieval system or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical articles or reviews.

The authors and publisher have made every effort in the preparation of this book to ensure the accuracy of the information. However, the information contained in this book is sold without warranty, either express or implied. Neither the authors, Wrox Press nor its dealers or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

First Published

May 2000



Published by Wrox Press Ltd
Arden House, 1102 Warwick Road, Acock's Green, Birmingham B27 6BH, UK
Printed in USA
ISBN 1-861003-62-5

Trademark Acknowledgements

Wrox has endeavored to provide trademark information about all the companies and products mentioned in this book by the appropriate use of capitals. However, Wrox cannot guarantee the accuracy of this information.

Credits

Authors

Karl Avedal
 Danny Ayers
 Timothy Briggs
 Carl Burnham
 Ari Halberstadt
 Ray Haynes
 Peter Henderson
 Mac Holden
 Sing Li
 Dan Malks
 Tom Myers
 Alexander Nakhimovsky
 Stéphane Osmont
 Grant Palmer
 John Timney
 Sameer Tyagi
 Geert van Damme
 Mark Wilcox
 Steve Wilkinson
 Stefan Zeiger
 John Zukowski

Technical Editors

Richard Huss
 Greg Pearson

Development Editor

Timothy Briggs

Managing Editor

Paul Cooper

Project Manager

Chandima Nethisinghe

Technical Reviewers

Thomas Bishop
 Jason Bock
 Michael Boerner
 Carl Burnham
 Cosmo DiFazio
 Matthew Ferris
 Ethan Henry
 David Hudson
 Jim Johnson
 Rod Johnson
 Andrew Jones
 Jim MacIntosh
 Stephen Potts
 David Schultz
 John Timney
 Krishna Vedati
 Paul Warren
 Andrew H. Watt
 Mark Wilcox

Production Project Manager

Mark Burdett

Design / Layout

Tom Bartlett
 William Fallon
 Jonathan Jones
 Laurent Lafon

Index

Alessandro Ansa

Cover Design

Shelley Frazier

Cover Photographs

Left to right:

Danny Ayers, Carl Burnham, Ari Halberstadt, Ray Haynes
Sing Li, Dan Malks, Tom Myers, Alexander Nakhimovsky
Stéphane Osmont, Grant Palmer, John Timney, Geert van Damme
Mark Wilcox, Steve Wilkinson, Stefan Zeiger, John Zukowski

About the Authors

Karl Avedal

Karl Avedal has been a Java developer since the language was publically launched in 1995. With the advent of Java server side technologies like servlets he quickly turned his attention to the server and worked a lot with CORBA before his first contacts with EJB in 1998. He is now a developer with the Orion Application Server team (<http://www.orionserver.com>). He is also taking part in the development of the the J2EE 1.3 specification as well as the JSP 1.2 and Servlet 2.3 specifications as a member of the expert groups for these standards.

Danny Ayers

Since going freelance, Danny Ayers divides his time unequally between network consultancy and contract work, woodcarving, and creating dance-floor crazes. Contact: danny_ayers@yahoo.com

To Mary, my mother (because she grumbled when Caroline got the last dedication).

Carl Burnham

Carl is a web developer, Internet consultant, and founder of Southpoint.com, a popular U.S. travel portal. He has 15 years of experience working as an IT professional in the private and public sectors, and was recently a senior network administrator for a leading Internet company. His interests include developing web sites, writing, photography, and using the full potential of the Web. He travels extensively via RV, with his wife Rhonda, golden retriever Gus, blind mutt Rocky, and Peewee, the cat. He can be reached at burnhamc@southpoint.com.

Dan Malks

Dan Malks is an Enterprise Java Architect with Sun Microsystems, working in the Sun Java Center in McLean, VA. He received a Master of Science degree in Computer Science from Johns Hopkins University in 1996 after having earned a Bachelor of Science in Computer Science from The College of William and Mary in 1987. While focusing on Object-Oriented technologies, he has developed in a variety of environments, including Smalltalk and most recently Java. He has published a number of articles about Java in leading industry periodicals, in addition to being a contributing author to *Professional JSP*, Wrox, 2000. Currently he has been focusing on Distributed, Service-based architectural designs, patterns and implementations.

Thanks to the helpful group of editors and reviewers at Wrox Press and most of all to Beth, my wonderful wife.

Ray Haynes

I have been programming ASP using JavaScript for 3 years now. I discovered JSP about 6 months ago and have been intrigued since. As the Lead Web Developer of an Internet start-up company in Kansas City, MO, I have been told that I am the ultimate geek when it comes to web development. My full time job is as a web programmer, and when I get home, I work on my own web sites. I am also attending college, working for my Bachelor of Science in Computer Information Systems, and should graduate around October 2001. My girlfriend, while feeding and watering me so that I don't wither away, keeps my heart warm, and my cat keeps my feet warm on late nights.

Peter Henderson

Peter is a senior developer at Oyster Partners, a Web Consultancy in London. Since graduating in mathematics he moved back into computing. He has been having fun working with web technologies for the last 4 years.

Mac Holden

Mac Holden has over 15 years experience in the Information Technology Industry. For the last 10 years he has been running his own software house based in South East Asia. The company initially concentrated on development and implementation of Client Server systems and then moved increasingly into web-based applications written in Java. He first became interested in Java in its alpha days as a means of connecting remote locations such as mines and oil rigs to their head offices. Mac is also the chief designer and developer of a pure Java application development tool called JdJ Servlet Builder which creates database aware servlets from HTML forms.

Sing Li

First bitten by the computer bug in 1978, Sing has grown up with the microprocessor revolution. His first 'PC' was a \$99 do-it-yourself COSMIC ELF computer with 256 bytes of memory and a 1 bit LED display. For two decades, Sing has been an active author, consultant, speaker, instructor, and entrepreneur. His wide-ranging experience spans distributed architectures, multi-tiered Internet/Intranet systems, computer telephony, call center technology, and embedded systems. Sing has participated in several Wrox projects in the past, and has been working with (and writing about) Java and Jini since their very first alpha release. He is an active participant in the Jini community.

Tom Myers

Tom has a BA (cum laude), St. John's College, Santa Fe, New Mexico ("Great Books" program), 1975 and a PhD in computer science from the University of Pennsylvania, 1980. He taught computer science at the University of Delaware and Colgate before becoming a full-time consultant and software developer. He is the author of "Equations, Models, and Programs: A Mathematical Introduction to Computer Science" Prentice-Hall Software Series, 1988, several articles on theoretical computer science, and two joint titles with Alexander Nakhimovsky: *JavaScript Objects*, Wrox, 1998, and *Professional Java XML Programming with Servlets and JSP*, Wrox, 1999.

Alexander Nakhimovsky

Alexander Nakhimovsky received an MA in mathematics from Leningrad University in 1972 and a PhD in general linguistics from Cornell in 1978, with a graduate minor in computer science. He taught general and slavic linguistics at Cornell and SUNY Oswego before joining Colgate's computer science department in 1985. He published a book and a number of articles on theoretical and computational linguistics, several Russian language textbooks, a dictionary of Nabokov's *Lolita*, and, jointly with Tom Myers, *JavaScript Objects*, Wrox, 1998, and *Professional Java XML Programming with Servlets and JSP*, Wrox, 1999.

Stéphane Osmont

Stéphane Osmont is a Software Engineer at Netscape Communications. Stéphane worked on the team that developed the JSP engine for Netscape Enterprise Web Server 4.0, and has recently been working on the Online Procurement ecommerce solutions for the Sun-Netscape Alliance. Stéphane, whose interests include traveling to South of France to visit his family and hanging out with friends and his wife Karen around the sunny San Francisco Bay Area, will be starting a new adventure by joining the greenlight.com team. He can be reached at sosmont@seolan.com.

Grant Palmer

Grant has worked as a scientific programmer in the Space Technology Division at the NASA Ames Research Center for the past 15 years. He has worked with Java since 1996 developing programs for scientific applications as well as converting older FORTRAN and C codes to the Java platform.

Grant lives in Chandler, Arizona with his wife, Lisa, and his two sons, Jackson and Zachary. In his spare time, Grant enjoys skiing and gardening, and is a competitive swimmer. He also likes to watch movies and read historical fiction.

John Timney

John lives in the UK with his wife Philippa in a small town called Chester-le-Street in the north of England. He is a postgraduate of Nottingham University having gained an MA in Information Technology following a BA Honours Degree from Humberside University. John currently works for Syntegra at their Newcastle office and specialises in Internet development. John's hobbies include Martial Arts and he has black belts in two different styles of Karate. His computing expertise has gained him a Microsoft MVP (Most Valuable Professional) award.

Sameer Tyagi

Sameer writes regularly for online and print publications. He has over four years of experience in software design and development and specializes in server side Java based distributed applications. (N-tier architectures, JDBC, JNDI, EJB, JMS, RMI, JSP, Servlets *et al.*) He has a Bachelors in Electronic Engineering and numerous other certifications.

He is a Java addict who gets his fix by jumping head on into almost anything that compiles into bytecodes and is known to blame that newly discovered area of the brain called the Javasphere for such stimuli.

When he's not going through another latte flavor, he can be found flying around at 15000 ft in a small Cessna.

Geert van Damme

I live in Leuven (Belgium) with my wife Sofie and my little son Jules. I studied Mathematical Psychology and Philosophy but ended up working in the IT business after a short while. In 1997 I started my own development and consulting company Darling, currently focussing on server side Java. Since then I work as an independant consultant on a number of projects, mainly from my home office. I can be reached at geert.vandamme@darling.be.

You can leave your name on my Graffiti wall at <http://www.gojasper.be/wall.jsp>

Mark Wilcox

Mark is the Web Administrator for the University of North Texas. He's also a frequent author and speaker on a variety of Internet topics. You can reach him at mark@mjlwilcox.com.

To my lovely wife, Jessica. Thanks to our parents for bringing us up right.

Steve Wilkinson

Steven is a hands-on software developer with over 13 years experience. Steven is currently a Principal with Elkhorn Creek Software, Incorporated where he concentrates on design and implementation of web based applications using Java Technologies.

Steven has been using Java Technologies since 1996. He has worked on projects for companies that range from start-ups to fortune 500 companies like Sun Microsystems, MCI, BellSouth and IBM. In these previous positions, Steven has used Java Technologies as RMI, Java Servlets, JHTML, JavaServer Pages, Java Applications, and Java Applets.

Steven has a degree in Electrical Engineering from the University of Kentucky, and is currently finishing his Master's of Science in Computer Information Systems from Regis University in Denver, Colorado.

Steven is also a contributing author to *Developing Java Servlets* by James Goodwill where he developed two chapters on Servlets and Object Databases and Servlets as Distributed Clients using RMI and CORBA.

Steven would like to thank his partner Diana for her patience, understanding, proof reading, and encouragement during his writing experiences.

Stefan Zeiger

Stefan Zeiger has been working as a freelance Java programmer since 1997 and studying computer science at the Technical University of Darmstadt since 1996. He is the author of the *NetForge* web server software and the popular online servlet tutorial *Servlet Essentials*.

John Zukowski

John Zukowski is the Content Director with jGuru.com. He received a B.S. in computer science and mathematics from Northeastern University and M.S. in computer science from Johns Hopkins University. He is the author of *John Zukowski's Definitive Guide to Swing for Java 2* from APress, *Java AWT Reference* from O'Reilly & Associates, as well as *Borland's JBuilder: No Experience Required* and *Mastering Java 2* from Sybex. In addition, John has authored numerous Java technologies articles and serves on the Senior Advisory Board of JavaWorld. John also serves as the vice-chairman of ACM's WebTech user group and the *Focus on Java* guide for About.com (<http://java.about.com>).

Congratulations to my wife Lisa for completing her Masters.

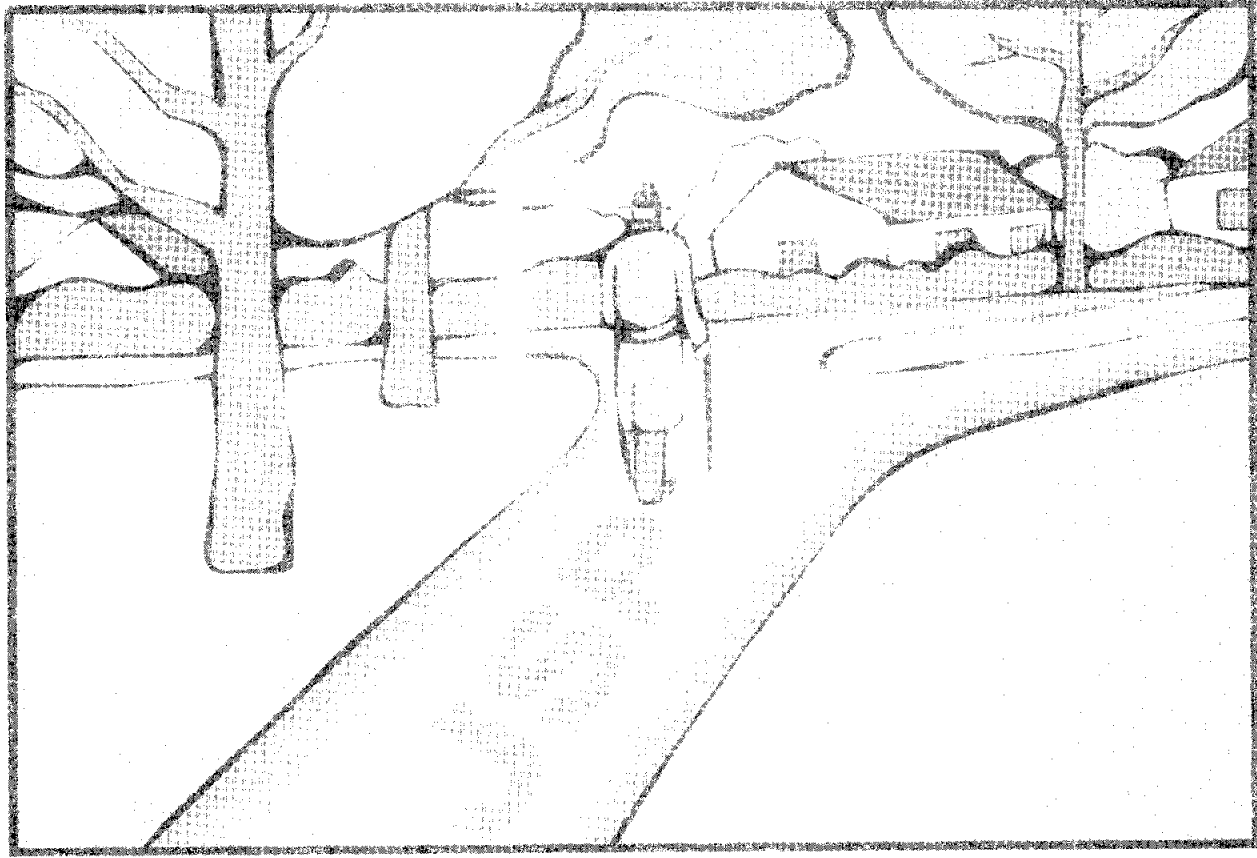


Table of Contents

Introduction	1
Welcome	1
Who is this Book For?	1
What's Covered in this Book	2
What You Need to Use this Book	2
Conventions	3
 Chapter 1: Introducing JavaServer Pages	 5
The Java 2 Enterprise Edition	6
JavaServer Pages	8
How Does a JSP Do Its Stuff?	8
What's Happening?	11
Comparison with Existing Technologies	14
CGI	14
Web Server APIs	15
ASP	15
Client-Side Scripting (JavaScript, JScript, VBScript)	16
Servlets	16
The Future of the Platform	16
Summary	17
 Chapter 2: The Basics	 19
What do JSPs contain?	19
Generated Implementation Class and the JSP Life Cycle	20
Element basics	24

Table of Contents

Directives	25
The page Directive	25
Example	27
The include Directive	29
The taglib Directive	31
Scripting Elements	32
Declarations	32
Scriptlets	34
Expressions	37
Standard Actions	39
jsp:useBean	40
jsp:setProperty	44
jsp:getProperty	50
Example	51
jsp:param	51
jsp:include	51
jsp:forward	55
jsp:plugin	56
Implicit Objects	59
Revisiting jspInit() and jspDestroy()	62
Summary	63
Chapter 3: Beneath JSP	65
Web Application Server Architecture	66
The Servlet Environment	67
Scalability	68
Important Servlet API features	68
Session tracking	68
Cookies	69
URL rewriting	69
Using Sessions	69
Form data parsing	71
Shared data	72

Internationalized character I/O	72
Initialization parameters	73
Request delegation	73
Logging	74
Summary	75

Chapter 4: JSP and JavaBeans 77

Introduction	77
JavaBeans	78
Bean Properties	79
Creating Accessor Methods for Properties	80
Indexed Properties	81
Bean Methods	82
Bound Properties and Bean Events	82
Bound Properties	82
Bean Events	86
Bean Persistence and Storage	86
JavaBeans and JDBC	87
Connecting to the Database	87
Running SQL Statements	88
ResultSets	90
Creating our Sample Application	90
Logging the Application	92
Defining Constants	93
Creating the Connection Manager	94
Building the Main Bean	96
Customizing Textual Output	101
Creating the JSP Documents	102
Managing Database Logon	102
Selecting the Customer	104
Populating the HTML Form with SQL Data	104
Preparing the Server	105
A Registration Form Example	105
Possible Enhancements	107
Summary	107

Chapter 5: JSP Sessions	109
A Sample HTTP Exchange	110
Persistent Connections	110
Techniques for Maintaining User Information across Pages	111
HTTP Information	111
Hidden Fields	111
Extended Path Info and URL-rewriting	112
Cookies	112
Working with Cookies and Java	114
Sessions	116
So What is a Session?	116
Servlet Support for Sessions and Session Lifecycle	117
More on URL Rewriting	118
Session Lifecycle	119
Contexts	121
Listening for Changes to the Session Object	123
Sessions in Action	124
Application Architecture	125
The JavaBeans	126
The Catalog Class	126
The Item Class	127
The Order Class	128
The User Class	130
The Login Bean	131
A HashMap	132
The JSPs	132
checkLogin.jsp	132
browse.jsp	134
processOrder.jsp	136
showDetails.jsp	137
logOut.jsp	140
console.jsp	140
Sessions, HTTPS, and SSL	142
Summary	143

Chapter 6: Error Handling with JSP **145**

Types of Errors and Exceptions	146
Handling Translation Time Errors	146
Handling Run Time Exceptions	148
Writing the Error Page	148
JSP-Specific Exception Classes	151
JspException	151
JspError	151
An Example Web Application	151
Main Page	153
Output Page	154
Calculator Bean	155
Error Page	156
Configuration	159
Summary	159

Chapter 7: Java Database Connectivity and Connection Pooling **161**

Relational Database Management Systems	162
Alien Object Model	163
Why All the Driver Types?	164
How JSP and JDBC Fit Together	166
The Bigger J2EE Picture	169
Creating Our Own MySQL Database	169
About Our Server Configuration	172
Type 4 JDBC Driver for MySQL and Tomcat Configuration	172
Alternative Setup for Type 1 Driver with Win32 JDBC-ODBC Bridging	173
Coding a Simple JSP using JDBC	174
Using JavaBeans to 'Fold Up' Commonly Used Code	177
More Advanced JDBC	179
Querying a Database	179

Table of Contents

Adding Records to a Database	182
Deleting Database Records	186
Modifying Records in a Database	189
Completing the MyCo Administrative System	196
Modularity and Optimization	196
Other JDBC Features: Prepared Statements and Callable Statements	197
Multiple users and the Need for Connection Pooling	198
PoolMan to the Rescue	200
A Word on JDBC 2.0 Standard Extension for Database Pooling	202
Limitations, Implementation Caveats and Gotchas	203
The "froggie in the well" Syndrome and Mysterious JVM Seizures	204
Testing and Improving Performance	206
Summary	211
 Chapter 8: Introducing Tag Extensions and Libraries	 213
The Need for a Tag Extension Mechanism	214
What is a Custom Tag?	215
How Pages and Tags Sit Together	215
The Simplest Tag	216
The taglib Tag	217
The pageVisit Tag	218
Setting Page Variables from Your Tag	220
Evaluating the Body of a Tag	223
Actions Implementing the BodyTag Interface	224
Using a Bean	225
Supporting Actions with Body	229
Nested Tag Extensions	234
Cooperating Tags	235
Changes to AlphabetTag	236
The RowTag Class	237
Recipes for Tags	240
Using Other People's Work	241
Resources	248
Summary	249

Chapter 9: Dynamic GUIs 251

Introduction	251
What is a Dynamic GUI?	251
Co-Branding Model (Customizing the Look)	252
Web Portal Model (Customizing the Content)	252
Creating a Co-Branding Modeled Web Site	253
What will be customized?	253
What are Zone Blocks?	254
How do we store and retrieve the template data?	255
How do we Decide what the "Template Template" looks like?	255
How do we Allow People to Change their Template?	258
Creating a Web Portal Modeled Site	259
Database-Driven Content	259
Storing the Layout Information	259
What are Content Objects?	260
What are Content Zones?	261
Building the Custom Content	262
User Customization Form	267
Summary	270

Chapter 10: Debugging JSP 273

Why Debugging JSP is so Hard	274
IDE vs. Notepad/vi	275
The IDE Approach	275
The Hardcore Camp	276
Different Types of Errors	276
Compilation Errors	276
Run Time Errors	278
HTML / JavaScript errors	279
Debugging Techniques	280
Eyeballing	280

Comments	280
Java Comments	280
JSP Comments	281
HTML Comments	281
out.println(), log() and System.out.println()	284
log()	284
out.println()	284
Singleton or static Debug object	285
System.out.println()	286
Debugging Concurrency	288
Where can we Expect Concurrency Problems in JSP	289
How to Debug Concurrency Problems	290
Separating your Code	291
JDB Type Debugging	293
Starting the Debugger	294
Setting Breakpoints	295
Stepping Through the Code	295
Future Directions	295
Summary	296

Chapter 11: Global Settings 299

Design Decisions	300
Information from Different Sources	300
String Values	300
Availability	300
ServletContext	300
Singleton Class / Static Methods	301
Implementing the Settings Object	301
Architecture	301
SettingsSource Interface	302
Settings	302
A First SettingsSource Implementation	305
The First Tests	305
Virtual Directories	307

SettingsServlet	308
Settings from the Database	310
How and Where to use These Settings	312
Other Possible Implementations	313

Chapter 12: JSP Architecture **315**

Code Factoring and Role Separation	317
Redirecting and Forwarding	318
Architectures	319
The "Page-Centric" Approach	319
Page-View	321
Page-View with Bean	325
Looping in JSP	328
Further Refinements	329
The "Dispatcher" Approach	329
Mediator-View	331
The RequestDispatcher API	335
Mediator-Composite View	336
Service to Workers	340
Servlets versus JSPs	345
Summary	347

Chapter 13: Case Study: A Web Interface for "The Mutual Fund Company" **349**

Introduction	349
Outline of Case Study	351
Definition of the Business Problem	351
Description of the Task	352
Technology Selection	353
Functional User Interface Design	353
loginStep1.html	354
loginStep2.html	354
displayAccountSummary.html	355

Table of Contents

UML Package Definition	357
Object Design	358
Beans package: "CustomerInfoBean" class.	358
JSP Package: JSP servlet classes	360
Server Classes package	360
Sequence Diagrams	363
Implementation	367
loginStep1.jsp	368
getMethod.jsp	371
validateSSN.jsp	371
CustomerInfoBean.java	373
UserSSNServer.java	373
UserProfileServer.java	376
BaseHttpServlet.java	381
loginStep2.jsp	382
processingDelay.jsp	385
checkResponse.jsp	386
processingError.jsp	386
errorPage.jsp	387
InvestmentSummary.java	387
enterKeySubmit.js	388
ProfileMgr.java	389
CustomerProfile.java	392
SSNList.java	392
Address.java	393
Company.java	393
AccountSummary.java	394
Constants.java	395
CustomerProfile.java	395
CategorySummary.java	396
displayAccountSummary.jsp	397
Setting up the Tomcat Servlet Engine	401
Summary	405
Chapter 14: Case Study: Publishing Data to the Web	407
Overview	407
User Interface	408

Selecting Technologies	413
JDK Version	413
Presentation Layer	414
Data Storage	414
Image Files	414
Database	415
Relationships Among Tables	416
Foreign Keys	416
Database Portability	417
JSP Pages	418
Simple JSP Page	419
Header and Footer Pages	422
Category and Folder Pages	423
category-frame.jsp	424
category-list.jsp	424
category-index.jsp	425
category-gallery.jsp	426
category-edit.jsp	427
Login and Logout Pages	429
Error Page	432
Index and Test Pages	434
Java Source Code	434
Data Access	434
Object-Relational Mapping	435
Optimistic Locking	435
Transactions	435
Persistent Objects	436
Querying for Multiple Objects	440
Request Processing	441
Query Requests	442
Thumbnail Requests	443
Edit Requests	444
Image File Attributes	446
PhotoServlet	447
Utilities	447
Security	449

Table of Contents

Installation	449
Requirements	449
Installation Steps	450
Create a Database	450
PostgreSQL	450
mySQL	451
Configure Servlet	451
Point servlet container to PhotoDB files	451
Test PhotoDB	452
Search Engines	452
Using PhotoDB	453
Naming Items	453
Image Files	453
Using Thumbnails in HTML Pages	454
Summary	455
 Chapter 15: Security and Personalization with JNDI	 457
Security	458
What is Security?	458
Security Boundaries	459
Physical Security	459
System Security	460
Application Security	460
Access Control	460
Message Verification	461
Message Ownership	461
Tying this together	461
Personalization	462
A Brief Introduction to Cryptography	462
Symmetric vs Asymmetric Encryption	463
Symmetric Encryption	463
Asymmetric Encryption	463
Message Digests	464
Digital Signatures	464

User Authentication	465
Something the User Knows	465
Something the User Has	465
Something Inherent to the User	466
Variations	466
What is a Digital Certificate?	466
Public Key Infrastructure	467
Introducing the Secure Sockets Layer	467
How SSL Works	467
How a Client Determines a Server's Trustworthiness	468
Client Authentication with SSL	469
Using SSL with JSPs	469
Authentication and Authorization	469
Authentication	470
Web Server Authentication	470
Servlet Engine Authentication	470
Individual Applications	472
Authorization	472
Data Integrity and Privacy	473
Securing JSPs	473
Standard JSP Security	473
A Note On Realms	475
Default Tomcat Authentication	476
Naming and Directory Services	479
Naming Services	479
Directory Services	479
Why LDAP?	481
LDAP Data	481
Introducing JNDI	483
JNDI Service Providers, aka JNDI Drivers	486
How to obtain JNDI Service Providers	486
Developing your own service provider	487
Basic LDAP Operations	487
Standard LDAP Operations	487

Security Interceptors	488
Writing the Code	488
Configuring Tomcat	497
More SSL	499
Putting it Together	499
The Code	501
Summary	510
Other Resources	510

Chapter 16: Case Study: Implementing a Membership Based E-Commerce Application 513

Definition of our Business Problem	514
Why is Membership Important?	514
Designing a Membership Based Web Application	514
The Business Logic	515
Business Processes	515
The Registration Process	515
The Login/RegistrationFlow	517
Defining our Object Model	518
Defining the Data Structure of our Business Objects	519
Properties of the User Object	519
Properties of the Catalog Object	520
Properties of the Product Object	520
Properties of the Cart Object	520
Properties of the ProductReferencer Object	521
Choosing the Right Technology	521
Selection of a Back-End Repository	521
LDAP Server: A Great Choice for Storing our User Information	521
Why JSP, Servlets, and LDAP are Great Together	521
Installing and Running the Netscape Directory Server	522
Setting up a Servlet-JSP Compliant Web Server	524
Running our Case Study on the JSWDK 1.0	524
Running our Case Study on the iPlanet Web Server 4.1	525

Putting It All Together	528
Implementing our Business Objects	528
Our Main Business Object: The User Bean	528
The Login Process in Detail	530
Creating a New User: Writing to the LDAP Repository	531
The SaveNewUserProfile() Method	533
Making our Information Page a Reusable Template	535
The Login Process: Authenticating a Member using the LDAP Server	538
The Authentication Process	540
The Information Retrieval Process	542
Instantiating the User Bean for our Existing Member	544
The Dynamic Online Grocery Store	544
The Online Catalog	545
The Pagination Mechanism	548
The Online Cart	548
The Checkout Process	550
Enhancing our Membership Based Application	551
Summary	551

Chapter 17: Case Study: J2EE, EJBs, and Tag Libraries 553

Introduction to J2EE	553
Overview	554
Web applications	555
Enterprise applications	556
Sun J2EE Blueprints	556
Platform roles	556
Platform Contracts	556
Scalability and Fault tolerance	557
Future directions	557
Enterprise Java Beans	557
Overview of EJB	557
The Components	558
Session Beans	558

Table of Contents

Entity Beans	558
EJB container	559
Limitations when writing EJBs	559
Stateful and Stateless Operation	559
JMS	559
JMS Capabilites	560
Servlets/JSP	560
The Java Transaction API and the Java Transaction Service	560
Transactions and Enterprise Java Beans	560
JDBC	561
JNDI	561
JavaMail	562
XML and XSL	562
Introduction to the Case Study	562
Requirements for the Webstore Application	562
Interface Requirements	563
Feature Requirements	563
Design Requirements	563
Limitations	563
Webstore Design	564
Basic Architectural Design	564
Layers	564
Designing the JSPs	565
The Welcome Page	565
The Product Browsing Page	566
The Toolbar/Cart Page	566
The Checkout Page	566
Pages for functions	567
Designing the EJBs	567
CartBean	567
ProductBean	567
CustomerBean	568
Designing the Tag Extension Libraries	568
User Tag Extension Library	568
Cart Tag Extension Library	569

Implementing Webstore	569
Implementing the EJBs	569
CartBean	570
ProductBean	574
CustomerBean	575
Deployment Descriptors	575
Implementing the Tag Extension Libraries	577
Implementing the JSPs	580
The Product Browsing Page	580
The Toolbar/Cart Page	581
The Logout Page	583
Implementing the XSL	583
Packaging the Web Application	585
Packaging the Enterprise Application	585
Summary	586
Further Reading	586

Chapter 18: Case Study: Streaming Data with JSP **589**

Multimedia and the Web	589
Streaming in Brief	590
HTTP Streaming	590
Streaming and JSP	591
Java Media Framework	591
Interactive JSP Streaming Media System	592
Media Receiver	592
Media Transmitter	598
JSP Control	605
The Bad News	610
A JavaScript Tier	610
Troubleshooting	612
File locations	613
Other resources	613

**Chapter 19: Case Study: Weather with JSP, XSLT,
and WAP****615**

Introduction	615
A WAP Application	616
Installation Instructions	616
The web.xml File	617
Web and WAP examples	618
An Architecture For a Web JSP Application	618
An Example: Database Lookup and Email	619
The Main Page and the Bean	619
Output Files	620
Weather Reports	622
The Entry Page	623
The Main JSP Page	624
The Configuration Page	624
A Look at the Back-End	626
The DBHandler Class	626
The Dict Class	627
The Main Bean	628
configure()	629
doCommand() and whereTo()	630
What About the Query?	630
JSP Pages for Output	631
Updating the Database and Adding Queries	632
Adding Queries	633
From a Wired to a Wireless Environment	633
Wireless Application Environment (WAE)	634
Internet Applications and the Protocol Stack	634
HTTP Reminders	635
The Request	635
The Response	635
Setting Response Headers	636
The WAP Programming Model	636

WML	638
A Simple Example	638
The Composition of the <wml> Element	640
The <head> and <template> Elements	640
What's in the Cards? Text and Action	640
The <p> and <pre> Elements	641
Tasks and Events	642
What Tasks are There?	642
What Kinds of Events are There?	643
The oneventforward and oneventbackward Event Handlers	644
The ontimer Event Handler	644
User Events and the <do> Element	645
Summary of WML Tags	646
The WAP Weather Application	648
More Targets	648
The WML Entry Page	649
JSP Pages for WML Output: TimeTemp.jsp	651
Declarations and the <head>	651
The Table and the Form	652
JSP Pages for WML Output: AllTable.jsp	653
JSP Pages for WML Output: AllText.jsp	654
A Different Approach: XSL	656
XML Parsing	656
The Parser and the Application	657
More on SAX	657
A Simple Example	658
XSLT, XPath and Document Transformations	660
How XSLT Came About	660
What's XPath?	661
Where Does XSLT Happen?	661
The XSLT Processing Model	662
A Simple Example	662
Running the Example with XSLServlet	667
Configuring Tomcat and XSLServlet	668
Same for WML?	671

Web-WAP with XSLT	671
The Configuration Page	672
doCommand() in the Main Bean	673
XSLPrune.java	674
XSLSessionServlet and XMLOutputHandler	675
XSLSessionServlet	676
xslTimeTemp.xml	676
Declarations and top-level elements	677
Overview of the Rest	677
An Extension-Function Call	678
The Entire Stylesheet	678
A Stylesheet for WML	679
Declarations	679
The Rest of the Stylesheet	680
xslWmlAllText and Control of Output	681
Feeding a NodeIterator into xt	683
XmlQueryStringDoc	684
Reminder: QueryResult and QueryResultTable	684
Reminder: the Resolver class and the InputSource	685
Summary	686

Chapter 20: Case study: Porting ASP to JSP **689**

The System	690
Requirements	690
Analysis	690
Content Entry and Management	691
Content Preview and Display	691
User Administration	692
Architecture	692
Content Architecture	692
Initial Development	693
Technologies	694
ASP	694
COM Business Objects	694
Database Connectivity	695
Site Server	696

Initial Architecture	696
Current Status of Initial Development Phase	697
Phase Two	697
Porting ASP to JSP 0.91	697
Error Handling	698
Lack of Application Object	699
Inline Functions	701
File Upload	702
Useful Functions	703
Freeing System Resources	704
Case Sensitive Filenames	704
Where Have We Got To?	704
Porting the Business Object Layer	705
Databases	706
Database Independence	706
Unique Numbers	706
Syntax Differences	708
Connection Pooling	708
Porting Connectivity	709
Searching	709
Where we're at	710
Phase Three	711
JSP 1.0	711
Summary	712
Appendix A: Configuring Apache and Tomcat	715
Installing Apache	715
Installing Tomcat	716
Installing mod_jserv	717
server.xml	718
web.xml	718
Where to Put JSPs and Beans in a Web Application	720
Using ant and build.xml	721
Summary	723

Appendix B: JSP Syntax Reference 725

The page Directive	725
taglib Directive	726
Include Tags	727
Bean-handling Tags	728
jsp:useBean Tag	728
jsp:setProperty Tag	729
jsp:getProperty Tag	729
Comments	729
Declarations	730
Scriptlets	730
Expressions	730
Supporting Applets and Beans	730
Forwarding the Request	731
Implicit Objects	731

Appendix C: JSP and Servlet API Reference 733

The javax.servlet Package	733
Servlet Interfaces	734
RequestDispatcher Interface	734
Servlet Interface	735
ServletConfig Interface	737
ServletContext Interface	739
ServletRequest Interface	743
ServletResponse Interface	747
SingleThreadModel Interface	749
Servlet Classes	749
GenericServlet Class	749
ServletInputStream Class	752
ServletOutputStream Class	752
The javax.servlet.http Package	754
HTTP Servlet Interfaces	754
HttpServletRequest Interface	754
HttpServletResponse Interface	756
HttpSession Interface	762
HttpSessionBindingListener Interface	766
HttpSessionContext Interface	766

HTTP Servlet Classes	767
Cookie Class	767
HttpServlet Class	771
HttpSessionBindingEvent Class	776
HttpUtils Class	776
The javax.servlet.jsp Package	778
JSP Interfaces	778
HttpJspPage Interface	778
JspPage Interface	778
JSP Classes	779
JspEngineInfo Class	779
JspFactory Class	779
JspWriter Class	781
PageContext Class	783
The javax.servlet.jsp.tagext Package	787
Tag Interfaces	787
BodyTag Interface	787
Tag Interface	788
Tag Classes	790
BodyContent Class	790
BodyTagSupport Class	791
TagAttributeInfo Class	793
TagData Class	794
TagExtraInfo Class	795
TagInfo Class	796
TagLibraryInfo Class	798
TagSupport Class	800
VariableInfo Class	802
getScope() Method	803

Appendix D: HTTP 805

URL Request Protocols	805
HTTP Basics	806
Client Request	806
HTTP Request Methods	807
Server Response	808
HTTP Headers	809
Server Environment Variables	812

Appendix E: JSP for ASP developers 819

Common Ground	820
JSP Scripting Elements	821
Expression Scriptlets	821
Code Scriptlets	822
Declaration Scriptlets	823
What's the Difference?	823
Declarations	824
Expressions	824
Comments	824
The Include Directive	825
The Response Object	826
The Request Object	828
Session and Application Objects	830
Application Object	833
Making Use of JavaBeans	835
A Different Approach to Datatypes	838
Databases	843
Adding a Business Layer	846
Handling Errors in ASP and JSP	851
Summary	853

Appendix F: Support, Errata, and p2p.wrox.com 855

The Online Forums at p2p.wrox.com	856
How To Enroll For Support	856
Why this system offers the best support	857
Checking The Errata Online at www.wrox.com	857
Wrox Developer's Membership	857
Finding an Errata on the Web Site	859
Add an Errata : E-mail Support	860
Customer Support	860
Editorial	860
The Authors	860
What We Can't Answer	860
How to Tell Us Exactly What You Think	860

Index 863

