

Spring Recipes

A Problem-Solution Approach



Gary Mak

Spring Recipes: A Problem-Solution Approach

Copyright © 2008 by Gary Mak

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): 978-1-59059-979-2

ISBN-10 (pbk): 1-59059-979-9

ISBN-13 (electronic): 978-1-4302-0624-8

ISBN-10 (electronic): 1-4302-0624-1

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.

Java™ and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc., in the US and other countries. Apress, Inc., is not affiliated with Sun Microsystems, Inc., and this book was written without endorsement from Sun Microsystems, Inc.

Lead Editors: Steve Anglin, Tom Welsh

Technical Reviewers: Sam Brannen, Kris Lander

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: Kylie Johnston

Copy Editor: Damon Larson

Associate Production Director: Kari Brooks-Copony

Production Editor: Laura Esterman

Compositor: Dina Quan

Proofreader: April Eddy

Indexer: Ron Strauss

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>.

Contents at a Glance

About the Author	xv
About the Technical Reviewers	xvii
Acknowledgments	xix
Introduction	xxi

PART 1 ■ ■ ■ Core

■ CHAPTER 1	Inversion of Control and Containers	3
■ CHAPTER 2	Introduction to Spring	21
■ CHAPTER 3	Bean Configuration in Spring	41
■ CHAPTER 4	Advanced Spring IoC Container	93
■ CHAPTER 5	Dynamic Proxy and Classic Spring AOP	135
■ CHAPTER 6	Spring 2.x AOP and AspectJ Support	167

PART 2 ■ ■ ■ Fundamentals

■ CHAPTER 7	Spring JDBC Support	209
■ CHAPTER 8	Transaction Management in Spring	247
■ CHAPTER 9	Spring ORM Support	287
■ CHAPTER 10	Spring MVC Framework	321
■ CHAPTER 11	Integrating Spring with Other Web Frameworks	395
■ CHAPTER 12	Spring Testing Support	417

PART 3 ■ ■ ■ Advanced

■ CHAPTER 13	Spring Security	465
■ CHAPTER 14	Spring Portlet MVC Framework	511
■ CHAPTER 15	Spring Web Flow	545
■ CHAPTER 16	Spring Remoting and Web Services	583
■ CHAPTER 17	Spring Support for EJB and JMS	625
■ CHAPTER 18	Spring Support for JMX, E-mail, and Scheduling	663
■ CHAPTER 19	Scripting in Spring	697
■ INDEX	709

Contents

About the Author	xv
About the Technical Reviewers	xvii
Acknowledgments	xix
Introduction	xxi

PART 1 ■ ■ ■ Core

■ CHAPTER 1	Inversion of Control and Containers	3
	1-1. Using a Container to Manage Your Components	4
	1-2. Using a Service Locator to Reduce Lookup Complexity	9
	1-3. Applying Inversion of Control and Dependency Injection	11
	1-4. Understanding Different Types of Dependency Injection	13
	1-5. Configuring a Container with a Configuration File	17
	1-6. Summary	20
■ CHAPTER 2	Introduction to Spring	21
	2-1. Introducing the Spring Framework	21
	2-2. Installing the Spring Framework	26
	2-3. Setting Up a Spring Project	28
	2-4. Installing Spring IDE	30
	2-5. Using Spring IDE's Bean-Supporting Features	32
	2-6. Summary	39
■ CHAPTER 3	Bean Configuration in Spring	41
	3-1. Configuring Beans in the Spring IoC Container	41
	3-2. Instantiating the Spring IoC Container	45
	3-3. Resolving Constructor Ambiguity	48
	3-4. Specifying Bean References	51
	3-5. Checking Properties with Dependency Checking	55
	3-6. Checking Properties with the @Required Annotation	58

3-7. Auto-Wiring Beans with XML Configuration	60
3-8. Auto-Wiring Beans with @Autowired and @Resource	64
3-9. Inheriting Bean Configuration	71
3-10. Defining Collections for Bean Properties	74
3-11. Specifying the Data Type for Collection Elements	81
3-12. Defining Collections Using Factory Beans and the Utility Schema	84
3-13. Scanning Components from the Classpath	86
3-14. Summary	92

■ CHAPTER 4 **Advanced Spring IoC Container** 93

4-1. Creating Beans by Invoking a Constructor	93
4-2. Creating Beans by Invoking a Static Factory Method	97
4-3. Creating Beans by Invoking an Instance Factory Method	98
4-4. Creating Beans Using Spring's Factory Bean	100
4-5. Declaring Beans from Static Fields	102
4-6. Declaring Beans from Object Properties	104
4-7. Setting Bean Scopes	106
4-8. Customizing Bean Initialization and Destruction	108
4-9. Making Beans Aware of the Container	114
4-10. Creating Bean Post Processors	115
4-11. Externalizing Bean Configurations	120
4-12. Resolving Text Messages	121
4-13. Communicating with Application Events	123
4-14. Registering Property Editors in Spring	126
4-15. Creating Custom Property Editors	129
4-16. Loading External Resources	131
4-17. Summary	134

■ CHAPTER 5 **Dynamic Proxy and Classic Spring AOP** 135

5-1. Problems with Non-Modularized Crosscutting Concerns	136
5-2. Modularizing Crosscutting Concerns with Dynamic Proxy	144
5-3. Modularizing Crosscutting Concerns with Classic Spring Advices	150
5-4. Matching Methods with Classic Spring Pointcuts	160
5-5. Creating Proxies for Your Beans Automatically	163
5-6. Summary	165

CHAPTER 6	Spring 2.x AOP and AspectJ Support	167
	6-1. Enabling AspectJ Annotation Support in Spring	168
	6-2. Declaring Aspects with AspectJ Annotations	170
	6-3. Accessing the Join Point Information	176
	6-4. Specifying Aspect Precedence	178
	6-5. Reusing Pointcut Definitions	180
	6-6. Writing AspectJ Pointcut Expressions	182
	6-7. Introducing Behaviors to Your Beans	187
	6-8. Introducing States to Your Beans	190
	6-9. Declaring Aspects with XML-Based Configurations	192
	6-10. Load-Time Weaving AspectJ Aspects in Spring	196
	6-11. Configuring AspectJ Aspects in Spring	201
	6-12. Injecting Spring Beans into Domain Objects	202
	6-13. Summary	206

PART 2 ■ ■ ■ Fundamentals

CHAPTER 7	Spring JDBC Support	209
	7-1. Problems with Direct JDBC	209
	7-2. Using a JDBC Template to Update a Database	216
	7-3. Using a JDBC Template to Query a Database	221
	7-4. Simplifying JDBC Template Creation	227
	7-5. Using the Simple JDBC Template with Java 1.5	230
	7-6. Using Named Parameters in a JDBC Template	233
	7-7. Modeling JDBC Operations As Fine-Grained Objects	236
	7-8. Handling Exceptions in the Spring JDBC Framework	240
	7-9. Summary	245
CHAPTER 8	Transaction Management in Spring	247
	8-1. Problems with Transaction Management	248
	8-2. Choosing a Transaction Manager Implementation	253
	8-3. Managing Transactions Programmatically with the Transaction Manager API	255
	8-4. Managing Transactions Programmatically with a Transaction Template	257
	8-5. Managing Transactions Declaratively with Classic Spring AOP	260

8-6. Managing Transactions Declaratively with Transaction Advices . . .	263
8-7. Managing Transactions Declaratively with the @Transactional Annotation	265
8-8. Setting the Propagation Transaction Attribute	266
8-9. Setting the Isolation Transaction Attribute	271
8-10. Setting the Rollback Transaction Attribute	279
8-11. Setting the Timeout and Read-Only Transaction Attributes	280
8-12. Managing Transactions with Load-Time Weaving	282
8-13. Summary	285

■ CHAPTER 9 Spring ORM Support 287

9-1. Problems with Using ORM Frameworks Directly	288
9-2. Configuring ORM Resource Factories in Spring	299
9-3. Persisting Objects with Spring's ORM Templates	306
9-4. Persisting Objects with Hibernate's Contextual Sessions	312
9-5. Persisting Objects with JPA's Context Injection	315
9-6. Summary	319

■ CHAPTER 10 Spring MVC Framework 321

10-1. Developing a Simple Web Application with Spring MVC	321
10-2. Mapping Requests to Handlers	333
10-3. Intercepting Requests with Handler Interceptors	336
10-4. Resolving User Locales	339
10-5. Externalizing Locale-Sensitive Text Messages	342
10-6. Resolving Views by Names	343
10-7. Mapping Exceptions to Views	346
10-8. Constructing ModelAndView Objects	348
10-9. Creating a Controller with a Parameterized View	351
10-10. Handling Forms with Form Controllers	353
10-11. Handling Multipage Forms with Wizard Form Controllers	366
10-12. Grouping Multiple Actions into a Controller	375
10-13. Creating Excel and PDF Views	381
10-14. Developing Controllers with Annotations	385
10-15. Summary	393

CHAPTER 11	Integrating Spring with Other Web Frameworks	395
11-1.	Accessing Spring in Generic Web Applications	395
11-2.	Integrating Spring with Struts 1.x	400
11-3.	Integrating Spring with JSF	407
11-4.	Integrating Spring with DWR	412
11-5.	Summary	416
CHAPTER 12	Spring Testing Support	417
12-1.	Creating Tests with JUnit and TestNG	418
12-2.	Creating Unit Tests and Integration Tests	423
12-3.	Unit Testing Spring MVC Controllers	433
12-4.	Managing Application Contexts in Integration Tests	438
12-5.	Injecting Test Fixtures into Integration Tests	444
12-6.	Managing Transactions in Integration Tests	449
12-7.	Accessing a Database in Integration Tests	455
12-8.	Using Spring's Common Testing Annotations	459
12-9.	Summary	461

PART 3 ■ ■ ■ Advanced

CHAPTER 13	Spring Security	465
13-1.	Securing URL Access	465
13-2.	Logging In to Web Applications	476
13-3.	Authenticating Users	480
13-4.	Making Access Control Decisions	490
13-5.	Securing Method Invocations	494
13-6.	Handling Security in Views	497
13-7.	Handling Domain Object Security	499
13-8.	Summary	509
CHAPTER 14	Spring Portlet MVC Framework	511
14-1.	Developing a Simple Portlet with Spring Portlet MVC	511
14-2.	Mapping Portlet Requests to Handlers	519
14-3.	Handling Portlet Forms with Simple Form Controllers	529
14-4.	Developing Portlet Controllers with Annotations	537
14-5.	Summary	543

CHAPTER 15	Spring Web Flow	545
15-1.	Managing a Simple UI Flow with Spring Web Flow	545
15-2.	Modeling Web Flows with Different State Types	552
15-3.	Securing Web Flows	565
15-4.	Persisting Objects in Web Flows	568
15-5.	Integrating Spring Web Flow with JSF	574
15-6.	Summary	580
CHAPTER 16	Spring Remoting and Web Services	583
16-1.	Exposing and Invoking Services Through RMI	584
16-2.	Exposing and Invoking Services Through HTTP	588
16-3.	Choosing a Web Service Development Approach	592
16-4.	Exposing and Invoking Web Services Using XFire	595
16-5.	Defining the Contract of Web Services	601
16-6.	Implementing Web Services Using Spring-WS	605
16-7.	Invoking Web Services Using Spring-WS	612
16-8.	Developing Web Services with XML Marshalling	615
16-9.	Creating Service Endpoints with Annotations	621
16-10.	Summary	622
CHAPTER 17	Spring Support for EJB and JMS	625
17-1.	Creating EJB 2.x Components with Spring	625
17-2.	Accessing EJB 2.x Components in Spring	631
17-3.	Accessing EJB 3.0 Components in Spring	637
17-4.	Sending and Receiving JMS Messages with Spring	640
17-5.	Creating Message-Driven POJOs in Spring	655
17-6.	Summary	661
CHAPTER 18	Spring Support for JMX, E-mail, and Scheduling	663
18-1.	Exporting Spring Beans As JMX MBeans	663
18-2.	Publishing and Listening to JMX Notifications	675
18-3.	Accessing Remote JMX MBeans in Spring	677
18-4.	Sending E-mail with Spring's E-mail Support	680
18-5.	Scheduling with Spring's JDK Timer Support	688
18-6.	Scheduling with Spring's Quartz Support	691
18-7.	Summary	696

■ CHAPTER 19 Scripting in Spring	697
19-1. Implementing Beans with Scripting Languages	697
19-2. Injecting Spring Beans into Scripts	702
19-3. Refreshing Beans from Scripts	705
19-4. Defining Script Sources Inline	706
19-5. Summary	707
■ INDEX	709

About the Author



GARY MAK has been a technical architect and application developer on the enterprise Java platform for six years. In his career, Gary has developed a number of Java-based software projects, most of which are core application frameworks and software tools. He enjoys designing and implementing the complex parts of software projects.

Gary is a Sun-certified Java programmer and has a master's degree in computer science. His research interests include object-oriented technology, aspect-oriented technology, design patterns, and software reuse.

Gary specializes in building enterprise applications on frameworks including Spring, Hibernate, JPA, Struts, JSF, and Tapestry. He has been using the Spring framework in his projects for four years, since Spring version 1.0. Gary is also an instructor of courses on enterprise Java, Spring, Hibernate, web services, and agile development. He has written a series of Spring and Hibernate tutorials as course materials, parts of which are open to the public, and they're gaining popularity in the Java community. In his spare time, he enjoys playing tennis and watching tennis competitions.

About the Technical Reviewers



SAM BRANNEN is a senior software engineer at SpringSource, where he serves as a core developer of the Spring framework. Sam is also a member of the SpringSource team for Spring and Tomcat Integrated Products. He has been developing Java applications since 1996 and enterprise applications since 1999. During this time, Sam has enjoyed designing complex software architectures and implementing custom solutions with a focus on scalable, testable, multitiered web and client-server applications using Java (J2EE/Java EE) and the Spring framework. Sam recently designed and implemented the new annotation-driven Spring TestContext framework included in Spring 2.5.

Prior to joining SpringSource, Sam gained experience building applications for customers in various business sectors ranging from e-commerce to banking, retail, automotive, and social communities. Sam has a degree in computer science from the Georgia Institute of Technology and currently resides in the United Kingdom with his wife, Vera.



From the moment his parents gave him a Spectrum 48K for his seventh birthday, it became clear that **KRIS LANDER** was always going to be an early adopter when it came to technology. Upon leaving school, with a computer addiction and a mild vitamin A deficiency, he decided to turn his childhood passion into a serious vocation, embarking on a degree in software engineering from the University of Wales.

Kris's constant thirst for emerging Java technologies has become a trademark throughout his professional career. He has been a Java web enterprise (J2EE) specialist from day one and a developer of applications using Spring since 2003, which has led him to work on many large-scale IT projects for corporate blue chip and successful new technology companies on both sides of the Atlantic. Based and brought up in London, in his spare time he enjoys good food and music production.

Acknowledgments

I'd like to thank my family, especially my wife, Ivy. She gave me mental support in writing this book and did a great job in taking care of our family. Without her love and support, I could never have finished this book.

Additionally, I'd like to thank the Apress team who worked for this book, in particular the following individuals:

Steve Anglin, my acquisition editor, who recognized me and believed in me writing this book. He offered splendid ideas for this book's outline and organized a great team for the book.

Tom Welsh, my development editor, who provided excellent support on both the literary and technical aspects of this book. He often encouraged me when I got into writing trouble, and offered solutions to me.

Kylie Johnston, my project manager, who did a great job in managing the book schedule and responded to my requests very quickly. Her experience made the schedule very flexible so that I could do my job in a comfortable way.

Kris Lander, who reviewed this book from a technical standpoint, pointed out my faults, and gave valuable suggestions based on his rich experience in Spring.

Sam Brannen, my other technical reviewer, who gave a large number of professional comments that directly and immediately improved the book. His role as a core Spring developer made me very confident in his comments.

Damon Larson, my copy editor, who did a nice job in correcting my grammatical mistakes, improving my words, and keeping the conventions consistent throughout the book.

Laura Esterman, my production editor, who made this book look as nice as it does and handled my late modification requests.

Dina Quan, Ron Strauss, and April Eddy, who were also instrumental in making this a great book.

Introduction

Since my first time using it in 2004, I have been a big fan of Spring and have used it in almost all my projects. I'm deeply attracted to Spring's simplicity and reasonableness. Spring is the simplest and most powerful Java/Java EE application framework I have ever used, and its ability to solve difficult problems in simple ways has made a strong impression on me. The solutions offered by Spring may not be the best all over the world, but they are the most reasonable ones that I can think of.

Spring addresses most aspects of Java/Java EE application development and offers simple solutions to them. By using Spring, you will be lead to use industry best practices to design and implement your applications. The releases of Spring 2.x have added many improvements and new features to the 1.x versions. This book focuses on the latest Spring 2.5 features for building enterprise Java applications.

As a course instructor on programming technologies, I always find that the biggest challenge my students face is how to get their experimental projects running. Many programming books include code examples, but most only include code fragments, not complete projects. Most of these books provide complete projects for you to download from their web sites, but they don't provide a chance for you to build the projects step by step on your own. I believe that you'll learn a lot from the project-building process, and that you'll gain confidence once you have gotten your projects running—this was my inspiration for writing this book.

As active Java developers, we often have to master a new technology or framework. Since we are only developers who use a technology, not students who have to take an exam, we don't need to keep everything in our mind. We only need an efficient way of making references when necessary. To benefit experienced readers, and beginner readers who have read this book from cover to cover, I've organized each chapter with multiple problem-solution-based recipes. This way, you'll be able to easily look up a solution for a particular problem.

The topics in this book are introduced by complete and real-world code examples that you can follow step by step. Instead of abstract descriptions on complex concepts, you will find live examples in this book. When you start a new project, you can consider copying the code and configuration files from this book, and then modifying them for your needs. This can save you a great deal of work over creating a project from scratch.

Who This Book Is For

This book is for Java developers who would like to gain hands-on experience rapidly on Java/Java EE development using the Spring framework. If you are already a developer using Spring in your projects, you can also use this book as a reference, and you'll find the code examples very useful.

You don't need much Java EE experience to read this book. However, it assumes that you know the basics of object-oriented programming with Java (e.g., creating a class/interface, implementing an interface, extending a base class, running a main class, setting up your

classpath, and so on). It also assumes you have basic knowledge on web and database concepts and know how to create dynamic web pages and query databases with SQL statements.

How This Book Is Structured

This book covers Spring 2.5 from basic to advanced, and it introduces several common Spring projects that will bring significant value to your application development. It's divided into 19 chapters organized into 3 parts:

- *Part 1: Core*: This part focuses on the core concepts and mechanisms of the Spring framework. The chapters in this part aim to familiarize you with Spring's core so that you can learn about other topics and uses of Spring quickly.
 - *Chapter 1: Inversion of Control and Containers*: This chapter introduces the core concept of Spring—the IoC design principal—and the importance of containers. If you are already familiar with IoC, feel free to skip this chapter.
 - *Chapter 2: Introduction to Spring*: This chapter gives you an overview of Spring's architecture and related projects. It also demonstrates how to set up Spring in your development environment.
 - *Chapter 3: Bean Configuration in Spring*: This chapter introduces basic bean configuration in the Spring IoC container. Understanding the features in this chapter is required for reading the subsequent chapters.
 - *Chapter 4: Advanced Spring IoC Container*: This chapter covers the advanced features and internal mechanisms of the Spring IoC container. Although these features may not be used as frequently as those in Chapter 3, they are indispensable to a powerful container.
 - *Chapter 5: Dynamic Proxy and Classic Spring AOP*: This chapter explains why you need AOP and how you can modularize crosscutting concerns with dynamic proxies and classic Spring AOP. If you already understand AOP and want to use Spring AOP in Spring 2.x directly, feel free to skip to Chapter 6.
 - *Chapter 6: Spring 2.x AOP and AspectJ Support*: This chapter covers Spring 2.x AOP usage and some advanced AOP topics, including how to integrate the AspectJ framework into Spring applications.
- *Part 2: Fundamentals*: This part involves the fundamental topics of the Spring framework. The topics covered in this part are used frequently in developing enterprise applications.
 - *Chapter 7: Spring JDBC Support*: This chapter shows how Spring can simplify JDBC's uses through its JDBC accessing framework. It also serves as an introduction to Spring's data access module.
 - *Chapter 8: Transaction Management in Spring*: This chapter discusses Spring's different transaction management approaches and explains transaction attributes in detail.

- *Chapter 9: Spring ORM Support:* This chapter focuses on how to integrate popular ORM frameworks, including Hibernate and JPA, into Spring applications.
- *Chapter 10: Spring MVC Framework:* This chapter covers web-based application development using the Spring Web MVC framework, with both the traditional approach and the new annotation-based approach.
- *Chapter 11: Integrating Spring with Other Web Frameworks:* This chapter introduces how to integrate the Spring framework with several popular web application frameworks, including Struts, JSF, and DWR.
- *Chapter 12: Spring Testing Support:* This chapter covers basic testing techniques in Java applications and the testing support features offered by the Spring framework.
- *Part 3: Advanced:* This part covers advanced topics of the Spring framework and related projects. However, thoroughly covering each of these topics would require an entire book. The aim of these chapters is to provide you with the useful basics and usages specific to Spring.
 - *Chapter 13: Spring Security:* This chapter introduces how to secure applications using the Spring Security framework, formerly known as Acegi Security. This chapter focuses on securing web applications using Spring Security 2.0.
 - *Chapter 14: Spring Portlet MVC Framework:* This chapter covers portlet application development using the Spring Portlet MVC framework, and focuses on the portlet-specific features that are different from Web MVC.
 - *Chapter 15: Spring Web Flow:* This chapter introduces how to use Spring Web Flow to model and manage your web application's UI flows. This chapter focuses on using Spring Web Flow 2.0 in Spring MVC and JSF.
 - *Chapter 16: Spring Remoting and Web Services:* This chapter covers Spring's support for various remoting technologies, including RMI, Hessian, Burlap, HTTP Invoker, and Web Services. It also introduces developing contract-first web services using Spring Web Services.
 - *Chapter 17: Spring Support for EJB and JMS:* This chapter discusses how to develop EJB 2.x and 3.0 components with Spring's EJB support, and how to use Spring's JMS support to simplify sending, receiving, and listening for JMS messages.
 - *Chapter 18: Spring Support for JMX, E-mail, and Scheduling:* This chapter discusses how to export Spring beans as JMX MBeans and access remote MBeans. This chapter also introduces how to send e-mail and schedule tasks with Spring's e-mail and scheduling support.
 - *Chapter 19: Scripting in Spring:* This chapter discusses how to use popular scripting languages, including JRuby, Groovy, and BeanShell, in Spring applications.

Each chapter of this book discusses a Spring topic with multiple problem-solution recipes. You can look up a solution for a particular problem and see how the solution works in the “How It Works” section. Each chapter demonstrates a topic with a complete real-world example. The example within a chapter is coherent, but examples are independent between chapters.

Conventions

Sometimes when I want you to pay particular attention to a part within a code example, I will make this part's font bold. Please note that a bold part doesn't reflect a code change from the last version. In cases when a code line is too long to fit the page width, I will break it with a code continuation character (►). Please note that when you try out the code, you have to concatenate the line by yourself without any space.

Prerequisites

Because the Java programming language is platform independent, you are free to choose any supported operating system. However, I should let you know that I am using Microsoft Windows because the file system paths I present in this book are Windows-based. You can simply convert these paths into your operating system's format before trying out the code.

To make the most of the book, you should install JDK version 1.5 or higher. You should also have a Java IDE installed to make development easier. For this book, I have been using Eclipse Web Tools Platform (WTP) to develop my projects, and I recommend you install it also.

Downloading the Code

The source code for this book is available from the Apress web site (<http://www.apress.com>), in the Source Code/Download section. The source code is organized by chapters, each of which includes one or more independent Eclipse projects. Please refer to `readme.txt`, located in the root, for setting up and running the source code.

Contacting the Author

I always welcome your questions and feedback regarding to the contents of this book. You can send your comments to springrecipes@metaarchit.com and access the book discussion and updates on <http://www.metaarchit.com>.