

Pro Spring 2.5



Jan Machacek, Aleksa Vukotic,
Anirvan Chakraborty, and Jessica Ditt

Pro Spring 2.5

Copyright © 2008 by Jan Machacek, Aleksa Vukotic, Anirvan Chakraborty, and Jessica Ditt

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): 978-1-59059-921-1

ISBN-10 (pbk): 1-59059-921-7

ISBN-13 (electronic): 978-1-4302-0506-7

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.

SpringSource is the company behind Spring, the de facto standard in enterprise Java. SpringSource is a leading provider of enterprise Java infrastructure software, and delivers enterprise class software, support and services to help organizations utilize Spring. The open source–based Spring Portfolio is a comprehensive enterprise application framework designed on long-standing themes of simplicity and power. With more than five million downloads to date, Spring has become an integral part of the enterprise application infrastructure at organizations worldwide. For more information, visit www.springsource.com.

Lead Editors: Steve Anglin and Tom Welsh

Technical Reviewer: Rick Evans

Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Kevin Goff, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Project Manager: Sofia Marchant

Copy Editors: Heather Lang, Damon Larson

Associate Production Director: Kari Brooks-Copony

Production Editor: Kelly Winkquist

Compositor: Kinetic Publishing Services

Proofreader: April Eddy

Indexer: Broccoli Information Management

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>. You will need to answer questions pertaining to this book in order to successfully download the code.

To Marc, who still thinks that beans grow on trees

—Jan

*To my parents, Andja and Slobodan, who have guided me through life and encouraged me
to follow my own path*

—Aleksa

To my parents, Sujata and Kalyan, for their unconditional love and support

—Anirvan

*To Ioannis, for the vital nudges when I needed them most (and so much more), and to my
family, just for being my family*

—Jessica

Contents at a Glance

Foreword	xxi
About the Authors	xxiii
About the Technical Reviewer	xxiv
Acknowledgments	xxv
Introduction	xxvii

PART 1 ■ ■ ■ Getting Started with Spring

■ CHAPTER 1	Introducing Spring	3
■ CHAPTER 2	Getting Started	13
■ CHAPTER 3	Introducing Inversion of Control	31
■ CHAPTER 4	Beyond the Basics	73
■ CHAPTER 5	Introducing Spring AOP	147
■ CHAPTER 6	Advanced AOP	199
■ CHAPTER 7	Spring Schemas and Namespaces	259
■ CHAPTER 8	Spring Patterns	279

PART 2 ■ ■ ■ Data Access

■ CHAPTER 9	JDBC Support	319
■ CHAPTER 10	iBATIS Integration	373
■ CHAPTER 11	Hibernate Support	399

PART 3 ■ ■ ■ Enterprise Application Components

■ CHAPTER 12	Job Scheduling with Spring	445
■ CHAPTER 13	Mail Support in Spring	477
■ CHAPTER 14	Dynamic Languages	511

PART 4 ■ ■ ■ Java EE 5

■ CHAPTER 15	Using Spring Remoting	533
■ CHAPTER 16	Transaction Management	575
■ CHAPTER 17	Web Applications with Spring MVC	611
■ CHAPTER 18	Spring Web Flow	711
■ CHAPTER 19	Spring and AJAX	757
■ CHAPTER 20	JMX with Spring	771
■ CHAPTER 21	Testing with Spring	793
■ CHAPTER 22	Spring Performance Tuning	829
■ INDEX	857

Contents

Foreword	xxi
About the Authors	xxiii
About the Technical Reviewer	xxiv
Acknowledgments	xxv
Introduction	xxvii

PART 1 ■ ■ ■ Getting Started with Spring

■ CHAPTER 1	Introducing Spring	3
	What Is Spring?	3
	Beyond Dependency Injection	6
	Aspect-Oriented Programming with Spring	6
	Accessing Data in Spring	6
	Simplifying and Integrating with Java EE	7
	Job Scheduling Support	7
	Mail Support	7
	Dynamic Languages	8
	Remoting Support	8
	Managing Transactions	8
	The Spring MVC Framework	8
	Spring Web Flow	9
	AJAX	9
	Internationalization	9
	Simplified Exception Handling	9
	The Spring Project	10
	Origins of Spring	10
	The Spring Community	10
	Spring for Microsoft .NET	10
	The Spring IDE	11
	The Spring Security (Formerly Acegi)	11
	Alternatives to Spring	11
	The Sample Code	12
	Summary	12

CHAPTER 2	Getting Started	13
	Obtaining the Spring Framework	13
	Checking Out Spring from CVS	13
	Building Spring from Source Code	14
	Verifying Your Spring Distribution	15
	Spring Packaging	15
	Spring Dependencies	17
	Spring Sample Applications	20
	Setting Up Spring and Your IDE	21
	Hello, World	24
	Putting Spring Into “Hello, World”	27
	Dependency Injection	28
	The Impact of Spring	30
	Summary	30
CHAPTER 3	Introducing Inversion of Control	31
	IoC and DI	31
	Types of IoC	32
	Contextualized Dependency Lookup	33
	Constructor DI	35
	Setter DI	36
	Injection vs. Lookup	37
	Setter Injection vs. Constructor Injection	38
	IoC in Spring	39
	DI with Spring	39
	Beans and BeanFactories	40
	BeanFactory Implementations	40
	XML Bean Definition	41
	Using Constructor Injection	43
	Injection Parameters	46
	Understanding Bean Naming	56
	Bean Instantiation Modes	59
	Resolving Dependencies	62
	Automatically Wiring Your Beans	65
	Checking Dependencies	68
	Bean Inheritance	70
	Summary	72

CHAPTER 4	Beyond the Basics	73
	Spring's Impact on Application Portability	74
	Bean Life Cycle Management	74
	Hooking into Bean Creation	75
	Hooking into Bean Destruction	82
	Making Your Beans Spring Aware	87
	Using the BeanNameAware Interface	87
	Using the BeanFactoryAware Interface	89
	Using Method Injection	91
	Lookup Method Injection	91
	Method Replacement	96
	Using FactoryBean	100
	The MessageDigestFactoryBean	100
	Accessing a FactoryBean Directly	102
	The BeanFactoryPostProcessor	103
	Implementing a BeanFactoryPostProcessor	106
	JavaBeans PropertyEditor	110
	The Built-in PropertyEditors	110
	Creating a Custom PropertyEditor	113
	The BeanPostProcessor	118
	Implementing a BeanPostProcessor	120
	When to Use BeanPostProcessor	124
	The Spring ApplicationContext	125
	Implementations of ApplicationContext	126
	Using ApplicationContextAware	126
	Controlling Bean Initialization	127
	Using Annotation-Based Configuration	129
	Internationalization with MessageSource	133
	Using MessageSource in Stand-Alone Applications	141
	The MessageSourceResolvable Interface	141
	Using Application Events	141
	Considerations for Event Usage	143
	Accessing Resources	144
	Summary	145
CHAPTER 5	Introducing Spring AOP	147
	AOP Concepts	148
	Types of AOP	149
	Static AOP	149
	Dynamic AOP	149
	Choosing an AOP Type	149

AOP in Spring	149
The AOP Alliance	150
“Hello, World” in AOP	150
Spring AOP Architecture	152
The ProxyFactory Class	153
Creating Advice in Spring	153
Advisors and Pointcuts in Spring	169
The Pointcut Interface	170
Using ComposablePointcut	187
Pointcutting Summary	191
All About Proxies	191
Understanding Proxies	192
Using JDK Dynamic Proxies	192
Using CGLIB Proxies	192
Comparing Proxy Performance	193
Which Proxy to Use?	196
Summary	196

CHAPTER 6 **Advanced AOP** 199

@AspectJ	199
@AspectJ Aspects in Detail	203
Pointcuts	203
Pointcut Expressions	207
Exploring the Pointcut Expressions	208
Using the @Pointcuts in XML	211
Types of Advice	212
Argument Binding	220
Introductions	221
The Aspect Life Cycle	227
Framework Services for AOP	228
Creating Our First Aspect Using the aop Namespace	229
Pointcuts in the aop Namespace	230
Creating Advice Using the aop Namespace	231
Introductions in the aop Namespace	237
Which Style Should You Use?	240
Working with Spring AOP Proxies	241
Impact of Proxies	242
AspectJ Integration	246
Creating Your First AspectJ Aspect	247
Compiling the Sample Application	249
AspectJ’s Aspect Scope	251

Load-Time Weaving	251
Your First Load-Time Weaving Example	252
LoadTimeWeaver Lookup Strategies	254
Practical Uses of AOP	254
Performance and Health Monitoring	255
Summary	257
CHAPTER 7	
Spring Schemas and Namespaces	259
Why the New Configuration?	259
Schemas Included in Spring 2.5	262
The beans Schema	262
The context Schema	263
The util Schema	263
The tx Schema	266
The aop Schema	266
The jee Schema	267
The lang Schema	267
Behind the Schema Scenes	268
Custom Schemas	270
IDE Configuration	274
Summary	277
CHAPTER 8	
Spring Patterns	279
Directory Structure	279
Simple Applications	279
Complex Applications	281
Packaging and Naming	281
Design Patterns Introduction	282
Programming Against Interfaces	282
Creational Patterns	283
Structural Patterns	287
Behavioral Patterns	290
Template Method	292
Spring Application Patterns	293
Layered Design	294
High-Performance Paging	295
Multiple Error Reporting	298
User Interface Transactions	301
Background Processes	304
E-mail Notifications	309
Error Collecting and Logging	311
Summary	315

PART 2 ■ ■ ■ Data Access

CHAPTER 9	JDBC Support	319
	Key JDBC Concepts	319
	Using the DriverManager and Connections	322
	Using PreparedStatements	323
	Using CallableStatements	326
	Other JDBC Concepts	327
	Concepts in Spring Data Access Support	327
	JDBC Data Access Support	328
	Using the JdbcTemplate	329
	JdbcTemplate.execute	330
	JdbcTemplate.query and Friends	334
	JdbcTemplate.update	339
	JdbcTemplate.batchUpdate	340
	RdbmsOperation Subclasses	342
	SqlUpdate	343
	BatchSqlUpdate	348
	SqlCall and StoredProcedure	349
	SqlQuery and Its Subclasses	351
	JdbcTemplate or RdbmsOperation?	358
	Large Binary Objects	358
	JdbcDaoSupport	361
	Simple Spring JDBC	364
	SimpleJdbcTemplate	365
	SimpleJdbcCall	367
	SimpleJdbcInsert	369
	SimpleJdbcDaoSupport	370
	Summary	371
CHAPTER 10	iBATIS Integration	373
	What Is iBATIS?	373
	iBATIS Versions	374
	Infrastructure and Configuration	374
	Mapping Files	375
	sqlMap Files	376
	Configuring iBATIS and Spring	378

Selecting Data	380
Simple Selects	380
One-to-One Selects	384
One-to-Many Selects	388
Many-to-Many Selects	390
Updating Data	391
Deleting Data	393
Inserting Data	394
What's Missing from iBATIS?	397
Overall Performance	397
Summary	398
CHAPTER 11 Hibernate Support	399
Hibernate Primer	399
Packaging	400
Introduction to Hibernate Support	401
Using Hibernate Sessions	403
Using HibernateDaoSupport	407
Deciding Between HibernateTemplate and Session	409
Using Hibernate in Enterprise Applications	413
Preventing Stale Data Updates	413
Object Equality	416
Transactional Behavior	419
Lazy Loading	424
Dealing with Large Data Sets	435
Handling Large Objects	437
Combining Hibernate with Other DAO Code	441
Summary	441

PART 3 ■ ■ ■ Enterprise Application Components

CHAPTER 12 Job Scheduling with Spring	445
Scheduling Jobs Using JDK Timer	446
Trigger Types with Timer	446
Creating a Simple Job	447
Spring Support for JDK Timer Scheduling	449
Scheduling Jobs Using OpenSymphony Quartz	455
Introducing Quartz	455
Quartz Support in Spring	464

	Job Scheduling Considerations	470
	Choosing a Scheduler	470
	Packaging Job Logic Separately from the Job Class	470
	Task Execution and Thread Pooling	471
	Summary	475
CHAPTER 13	Mail Support in Spring	477
	The Spring Mail API Structure	479
	Configuring Mail Settings Using Spring	479
	Sending Simple E-mails	479
	Constructing and Sending E-mail Programmatically	479
	Constructing E-mail Declaratively	482
	Constructing and Sending MIME Messages	485
	Insight into Enterprise-Level E-mail Handling	500
	Summary	509
CHAPTER 14	Dynamic Languages	511
	Supported Languages Primer	511
	BeanShell	511
	Groovy	513
	JRuby	514
	Using Dynamic Languages As Spring Beans	515
	Behind the Scenes of Dynamic Language Support	517
	Proxying Dynamic Language Beans	518
	Performance	519
	Refreshable Beans	521
	BeanShell Beans	523
	JRuby Beans	525
	Groovy Beans	526
	Typical Usage for Dynamic Languages in Spring Applications	526
	Summary	530
 PART 4 ■ ■ ■ Java EE 5		
CHAPTER 15	Using Spring Remoting	533
	Spring Remoting Architecture	535
	Remote Method Invocation	535
	Exposing Arbitrary Services	536
	Accessing an RMI Service Using Proxies	538
	Exposing CORBA Services	540
	Accessing a CORBA Service	543

Web Services with JAX-RPC	544
Introducing Apache Axis	545
Creating a Web Service with ServletEndpointSupport	545
Accessing RPC-Style Web Services using Proxies	549
Working with JavaBeans in Axis Services	552
Using JAX-WS Web Services	556
Exposing Web Services Using SimpleJaxWsServiceExporter	556
Exposing a Web Service Using XFire	557
Accessing JAX-WS Web Services	559
Accessing Java Web Services from Other Clients	560
Creating Web Services with HTTP Invoker	563
Exposing Simple Services	564
Accessing an HTTP Invoker Service Using Proxies	567
Using Arbitrary Objects in HTTP Invoker Services	567
Using HTTP Basic Authentication	569
Choosing a Remoting Architecture	573
Summary	574

CHAPTER 16 Transaction Management	575
Exploring the Spring Transaction Abstraction Layer	575
Analyzing Transaction Properties	576
Exploring the TransactionDefinition Interface	576
Using the TransactionStatus Interface	578
Implementations of the PlatformTransactionManager	579
Exploring a Transaction Management Sample	579
Nontransactional Code	580
Programmatic Transaction Management	588
Using the TransactionTemplate Class	590
Programmatic Transaction Management Summary	591
Declarative Transaction Management	591
Using the TransactionProxyFactoryBean	591
Implications of Using Proxies in Transaction Management	593
AOP Transaction Management	594
Using Annotation-Based AOP Transaction Management	594
Using XML AOP Transaction Management	596
Working with Transactions Over Multiple Transactional Resources	599
Implementing Your Own Transaction Synchronization	601
Summary	610

CHAPTER 17	Web Applications with Spring MVC	611
	MVC Architecture	611
	Spring MVC	613
	MVC Implementation	613
	Using Handler Mappings	614
	Spring Controllers	616
	AbstractController	616
	ParameterizableViewController	617
	MultiActionController	618
	Interceptors	621
	Views, Locales, and Themes	622
	Using Views Programmatically	622
	Using View Resolvers	625
	Using Localized Messages	629
	Using Locales	630
	Using Themes	630
	Command Controllers	633
	Using Form Controllers	633
	Exploring the AbstractWizardFormController	640
	File Upload	644
	Handling Exceptions	647
	Spring and Other Web Technologies	650
	Using JSP	651
	Using Velocity	669
	FreeMarker	674
	Using XSLT Views	678
	Using PDF Views	680
	Using Excel Views	682
	Using Tiles	684
	JasperReports	696
	Spring Conventions Over Configuration	701
	Controller Conventions	701
	MultiActionController Conventions	702
	Model Conventions	703
	View Conventions	704
	Using Annotations for Controller Configuration	705
	@Controller	705
	@RequestMapping	706
	@RequestParam	707
	@ModelAttribute	707
	Using Annotations with the Command Controller	708
	Summary	709

CHAPTER 18	Spring Web Flow	711
	Introducing Spring Web Flow	712
	Core Concepts	713
	Obtaining Spring Web Flow	716
	Spring Web Flow Dependencies	718
	Hello, Web Flow!	719
	Exploring States	723
	View State	724
	Decision State	724
	End State	725
	Working with Transitions	725
	Advanced Concepts	727
	Expression Languages and Scopes	727
	Implementing Actions	731
	Model Data Binding	732
	Partial Rendering of Views	736
	Mapping Flow Input and Output Parameters	736
	Using Subflows	737
	Spring Web Flow Behind the Scenes	738
	Flow Execution Architecture	739
	Flow Executor	741
	Flow Definition Registry	742
	Flow Execution Repository	745
	Integration with Spring MVC	746
	Flow Handling	746
	View Resolving	747
	Securing Flows with Spring Security	747
	Step 1: Adding the SecurityFlowExecutionListener	748
	Step 2: Basic Authentication and Authorization Handling	748
	Step 3: Defining Security Rules in Flow Definitions	750
	Problem Solver	752
	Stateful Navigational Control	753
	Browser Navigation Bar Support and Double Submit	753
	Testing Flow Definitions	753
	Summary	754

CHAPTER 19	Spring and AJAX	757
	DWR	758
	Installing DWR	758
	Spring Configuration for DWR	758
	About the Complete Example	760
	Testing the DWR Configuration	764
	Running the Complete Example	765
	DWR Scripting Basics	766
	Using Simple Callback Functions	766
	Calling Metadata Objects	767
	engine.js	767
	Call Batching	767
	Call Ordering	768
	Handling Errors and Warnings	768
	util.js	768
	Security in DWR	768
	Advantages and Disadvantages of DWR	770
	Summary	770
CHAPTER 20	JMX with Spring	771
	JMX Refresher	771
	Exposing Your Beans	772
	MBeanExporter	772
	MBeanServerFactoryBean	774
	Exposing Your Beans in an Existing MBean Server	775
	Bean Registration Behavior	776
	Controlling Object Names	776
	Controlling the Management Interface	777
	MBeanInfoAssembler Interface	777
	MethodNameBasedMBeanInfoAssembler Interface	777
	Using Java Interfaces to Control the Management Interface	780
	Using Source-Level Metadata	782
	Remoting with Spring JMX	785
	Exposing Remote MBeans	785
	Accessing Remote MBeans	786
	Proxying MBeans	786
	Notifications in Spring JMX	787
	Notification Listeners	787
	Publishing Notifications	788
	Summary	792

CHAPTER 21	Testing with Spring	793
	Unit Testing	793
	Unit Tests	795
	Unit Testing the Web Layer	798
	Integration Tests	800
	Using AbstractSpringContextTests	807
	Using AbstractDependencyInjectionSpringContextTests	809
	Using AbstractTransactionalSpringContextTests	811
	Using AbstractAnnotationAwareTransactionalTests	813
	JNDI	817
	Spring TestContext Framework	819
	Application Context and DI with the TestContext Framework	820
	Transactions in the TestContext Framework	822
	Support Classes	824
	Test Coverage	826
	Summary	827
CHAPTER 22	Spring Performance Tuning	829
	Performance vs. Responsiveness	829
	Exploring Enterprise Application Performance Issues	830
	Measuring Java EE Application Performance	830
	Finding Out What to Measure	830
	Determining the Necessary Data Sets	837
	Improving the Data Access Tier	837
	Improving Transaction Management	847
	Controlling the Performance of Remote Calls	849
	Understanding View Performance	849
	Using Caching	849
	Performance Testing	851
	Monitoring Application Health and Performance	853
	More Resources on Performance Tuning	855
	Summary	855
INDEX		857

Foreword

It was with a heavy heart that I made the decision not to participate in writing *Pro Spring 2.5*. I am deeply thankful that Jan was around to pick up this book and run with it. *Pro Spring* has been a big part of my life for over three years, and I didn't relinquish the reins lightly. When Juergen and I set out working on Spring Framework 2.0, I knew that I wouldn't have the time to participate in the writing process and write the software at the same time. Fortunately, Jan was there to step into the breach.

Jan and Apress had additionally planned to release *Pro Spring 2.0*, but Juergen and I inadvertently made it impossible for them to keep up by making many changes to the Spring Framework. I vividly remember cringing when updating all the JSP form tags, knowing that I was creating yet more work for Jan.

With the 2.5 release just on the horizon, Jan made the sensible choice to forego a 2.0 edition and head straight for 2.5. This was a wise move. The Spring Framework 2.5 release reflects the state of the art in both the Spring Framework and in enterprise Java frameworks as a whole. A guide book to this critical tool is necessary reading for any conscientious Java developer.

I recall, back in the early days of running Cake Solutions, when we decided we needed to hire another programmer. We were very inexperienced at hiring in general, and hiring programmers is fraught with problems. We knew that we wanted to get a graduate, but we never imagined that we would get someone as accomplished as Jan.

I remember, in his first week, he wrote a complete desktop mailing package from scratch—and it worked. Over the last five years, Jan has been at the center of most of the projects run at Cake, many of which are large-scale Java products based on the Spring Framework. His knowledge of Spring comes from an immense amount of practical experience: he has been in the trenches with Spring since version 1.0 and has delivered successful systems on top of it.

To his credit, Jan realized that writing *Pro Spring 2.5* was too big a job for just one man, so he roped in the rest of the Cake Solutions team to help him. This prospect excited me greatly—a team of real programmers, with real experience in Spring, passing along that knowledge. There is no doubt that many will find this book to be an indispensable reference.

And so, although I am disappointed at being unable to work on this book myself, I am glad that Jan was there to deliver what so many people have been asking for, an updated version of *Pro Spring*. Enjoy,

Rob Harrop
*Principal Software Engineer and Lead Engineer
of the SpringSource Application Platform*

About the Authors



■ **JAN MACHACEK** is the chief software architect at Cake Solutions, which places him at the center of all architectural decisions in all projects. Apart from the architecture, Jan is often called on to help with some of the most complex and challenging areas of the implementation. Since joining Cake, Jan has proven his expertise in Java not only by taking on a wide variety of complex projects but also through his collection of published works. In his free time, Jan is a keen cyclist and a member of the Manchester Wheelers' Club. He tries his best in various time trials and road races.

Jan authored Chapters 1–4, 6, 9, 11, 14, 16, and 22.



■ **ALEKSA VUKOTIC** is a software architect at Cake Solutions. He oversees the architecture as well as the implementation of most of Cake's projects. He has extensive experience with most Java EE technologies, particularly Spring MVC and Security. He also has the knack, which means he can solve virtually any technical problem. He is an excellent tutor and is in charge of directing a team of Cake Solutions developers, helping the team in solving the most complex problems. As well as his interest in Java and .NET platforms, Aleksa enjoys sports, music, and nights out. Aleksa works with Jan on all of the major projects at Cake Solutions.

Aleksa authored Chapters 5, 8, 10, 15, 17, and 21.



■ **ANIRVAN CHAKRABORTY** is a senior developer at Cake Solutions. His extensive experience with the Spring Framework and attention to detail puts him in charge of the implementation of some of the challenging aspects of Cake Solutions's projects. Anirvan takes great pride in his code and always makes sure the code can be used as an example to others. When he does not have his head buried in Java EE and Linux, he enjoys good food and drink with his friends. He is also an ardent follower of the sport of cricket and enjoys reading detective novels.

Anirvan authored Chapters 7, 13, 19, and 20.



■ **JESSICA DITT** has been a developer at London-based Avenue A | Razorfish since early 2008. Prior to that, she was part of the Cake Solutions team for 2.5 years. She has been working on numerous enterprise-level projects, all of which were written using the Spring Framework and Spring Web Flow. Jessica has acquired significant expertise in efficient indexing using Lucene and has efficiently addressed Java EE application scalability issues using Gigaspaces. Out of the office, Jessica is a keen volleyball player and enjoys spending time in the gym.

Jessica authored Chapters 12 and 18.

About the Technical Reviewer



■ **RICK EVANS** is an independent contractor based in the UK with many years of experience working in the health, financial, and retail sectors. Over the years, Rick has committed on a number of open source projects, including Spring and Spring.NET. A polished teacher and mentor, Rick often speaks professionally about technology and delivers training on a wide range of enterprise technologies and disciplines.

Acknowledgments

When writing a book, a substantial amount of work goes on behind the scenes, and authors are backed by an excellent team of editors, proofreaders, and technical reviewers. This book was no exception, and we would like to thank everyone who worked on the book. Our thanks goes to Rick, the technical reviewer, who has done enormous amounts of work to ensure that this book is of the highest quality. The great editorial team at Apress also deserves our thanks: most importantly, Sofia Marchant; our editor, Tom Welsh; Heather Lang; Kelly Winqvist; and many others. Without their help, we would not have been able to complete this book. I would also like to thank Rob Harrop for agreeing to write the Foreword. Finally, we all wish to thank the managing director of Cake Solutions, Guy Remond; he gave some of Cake Solutions's time to us to work on the book.

Introduction

Recently, the Java world has witnessed a dramatic shift away from so-called “heavyweight” architectures such as Enterprise JavaBeans (EJB) toward lighter weight frameworks such as Spring. Complex and container-dependent services, such as CMP, and transaction management systems have been replaced with simpler alternatives such as Hibernate and aspect-oriented programming (AOP). At the core, Spring provides a comprehensive, lightweight container based on the principle of Inversion of Control (IoC), on which you can build your own applications. On top of this container, Spring provides a myriad of useful services, bringing together a large range of highly competent open source projects into a single cohesive framework.

The quality of the Spring Framework has seen it replacing traditional Java EE architectures in many cases; as a result, more and more developers see the need for comprehensive Spring skills. Despite Spring having quite an extensive suite of documentation and examples, we feel that many developers are still struggling to understand how to use Spring and, more importantly, how to use it effectively. Because of this, we decided to write a new edition of *Pro Spring*.

At first, we thought we would just update a few chapters and call it *Pro Spring 2.5*. However, we quickly realized that Spring 2.5 brought so many new features and improvements that, although we kept the old *Pro Spring* name, this is a completely new book.

Through this book, you will learn how to use Spring to build better web and stand-alone applications and how to sift through the many choices available to you through the framework. Our aim is to provide you with all the knowledge you need to use Spring *effectively* in your own applications and to give you insight into what is happening behind the scenes in Spring.

For example, you will

- Learn the fundamentals of IoC in the context of AOP.
- Become aware of the seamlessness and power of Spring by referencing the easy-to-understand sample applications we provide.
- Learn how to replace common EJB features with Spring alternatives, including Spring’s comprehensive AOP-based transaction management framework.
- Effectively manage your Spring components and applications using Spring’s built-in JMX engine.
- Learn how to add scheduling to your Spring application with Quartz.

After reading this book, you will be equipped with all the knowledge you need to build applications effectively using Spring and its related open source projects.

