

Mastering Enterprise JavaBeans™

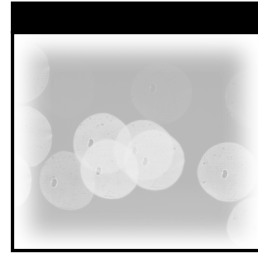
Third Edition

Ed Roman
Rima Patel Sriganesh
Gerald Brose



WILEY

Wiley Publishing, Inc.



Mastering Enterprise JavaBeansTM

Third Edition

Ed Roman
Rima Patel Sriganesh
Gerald Brose



WILEY

Wiley Publishing, Inc.

Mastering Enterprise JavaBeans™, Third Edition

Published by

Wiley Publishing, Inc.

10475 Crosspoint Boulevard

Indianapolis, IN 46256

www.wiley.com

Copyright © 2005 by Ed Roman, Gerald Brose, and Rima Patel Sriganesh

Published by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 0-7645-7682-8

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

3B/RX/RR/QU/IN

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Legal Department, Wiley Publishing, Inc., 10475 Crosspoint Blvd., Indianapolis, IN 46256, (317) 572-3447, fax (317) 572-4355, e-mail: brandreview@wiley.com.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Website is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Website may provide or recommendations it may make. Further, readers should be aware that Internet Websites listed in this work may have changed or disappeared between when this work was written and when it is read.

IN NO EVENT SHALL THE AUTHOR OR PUBLISHER BE LIABLE TO ANY PERSON FOR ANY INCIDENTAL, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION TO, LOSS OF PROFITS, LOSS OF DATA, BUSINESS INTERRUPTION OR ANY AND ALL OTHER SIMILAR DAMAGES OR LOSS, EVEN IF AUTHOR OR PUBLISHER OR THEIR SUPPLIERS OR THEIR AGENTS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

For general information on our other products and services or to obtain technical support, please contact our Customer Care Department within the U.S. at (800) 762-2974, outside the U.S. at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Library of Congress Control Number: 2004114268

Trademarks: Wiley, the Wiley Publishing logo and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Enterprise JavaBeans is a trademark of Sun Microsystems, Inc. in the U.S. or other countries. All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book.

To my wonderful wife, Christine, and to our boys, Johannes and Julius.

*Rima wishes to dedicate this book to her dearest and loving husband
Sriganesh, and her most wonderful parents.*



Credits

Acquisitions Editor

Robert M. Elliott

Development Editor

Sydney Jones

Technical Editor

Floyd Marinescu

Copy Editor

Michael Koch

Editorial Manager

Kathryn Malm Bourgoine

Vice President**& Executive Group Publisher**

Richard Swadley

Vice President and Publisher

Joseph B. Wikert

Project Coordinator

Erin Smith

Graphics and Production Specialists

Sean Decker

Kelly Emkow

Jennifer Heleine

Quality Control Technician

Brian H. Walls

Proofreading and Indexing

TECHBOOKS Production Services



Contents

Acknowledgments	xvi
Introduction	xvii
Part One Overview	1
Chapter 1 Overview	3
The Motivation for Enterprise JavaBeans	4
Component Architectures	7
Service-Oriented Architectures	8
Divide and Conquer to the Extreme with Reusable Services	9
Introducing Enterprise JavaBeans	11
Why Java?	12
EJB as a Business Tier Component	13
The EJB Ecosystem	15
The Bean Provider	16
The Application Assembler	16
The EJB Deployer	17
The System Administrator	17
The Container and Server Provider	18
The Tool Vendors	18
Summary of Roles	19
The Java 2 Platform, Enterprise Edition (J2EE)	21
The J2EE Technologies	22
Summary	26
Chapter 2 EJB Fundamentals	27
Enterprise Beans	27
Types of Beans	28
Distributed Objects: The Foundation for EJB	30
Distributed Objects and Middleware	32

Explicit Middleware	33
Implicit Middleware	34
What Constitutes an Enterprise Bean?	36
The Enterprise Bean Class	36
The EJB Object	37
The Home Object	42
The Local Interfaces	44
Deployment Descriptors	48
Vendor-Specific Files	49
Ejb-jar File	49
Summary of Terms	50
Summary	52
Chapter 3 Writing Your First Bean	53
How to Develop an EJB Component	54
The Remote Interface	55
The Local Interface	56
The Home Interface	57
The Local Home Interface	59
The Bean Class	61
The Deployment Descriptor	64
The Vendor-Specific Files	65
The Ejb-jar File	65
Deploying the Bean	66
The Optional EJB Client JAR File	67
Understanding How to Call Beans	68
Looking up a Home Object	68
Running the System	72
The Server-Side Output	73
The Client-Side Output	73
Implementing Component Interfaces	73
A Solution	74
Summary	75
Part Two The Triad of Beans	77
Chapter 4 Introduction to Session Beans	79
Session Bean Lifetime	79
Session Bean Subtypes	80
Stateful Session Beans	80
Stateless Session Beans	81
Special Characteristics of Stateful Session Beans	83
Achieving the Effect of Pooling with Stateful Beans	83
The Rules Governing Conversational State	84
Activation and Passivation Callbacks	85
Method Implementation Summary	88
A Simple Stateful Session Bean	88
Life Cycle Diagrams for Session Beans	98
Summary	102

Chapter 5	Writing Session Bean Web Services	103
	Web Services Concepts	103
	Web Services Standards	106
	XML Artifacts and Platform Independence	109
	Implementing a Web Service	110
	The JAX-RPC Service Endpoint Interface	111
	WSDL and the XML/Java Mapping	113
	Packaging and Deploying a Web Service Session Bean	113
	Implementing a Web Service Client	114
	Summary	117
Chapter 6	Introduction to Entity Beans	119
	Persistence Concepts	119
	Object-Relational Mapping	120
	What Is an Entity Bean?	122
	About the Files That Make Up an Entity Bean	124
	Features of Entity Beans	125
	Entity Beans Survive Failures	125
	Entity Bean Instances Are a View into a Database	126
	Several Entity Bean Instances May Represent the Same Underlying Data	127
	Entity Bean Instances Can Be Pooled	128
	There Are Two Ways to Persist Entity Beans	131
	Creation and Removal of Entity Beans	132
	Entity Beans Can Be Found	136
	You Can Modify Entity Bean Data without Using EJB	136
	Entity Contexts	137
	getEJBLocalObject() / getEJBObject()	138
	getPrimaryKey()	138
	Summary	139
Chapter 7	Writing Bean-Managed Persistent Entity Beans	141
	Entity Bean Coding Basics	141
	Finding Existing Entity Beans: Finder Methods	143
	Bean-Managed Persistence Example: A Bank Account	150
	Account.java	151
	AccountLocal.java	152
	AccountHome.java	153
	AccountLocalHome.java	155
	AccountPK.java	156
	AccountBean.java	158
	AccountException.java	170
	Client.java	171
	The Deployment Descriptor	173
	The Container-Specific Deployment Descriptor	175
	Setting up the Database	175
	Running the Client Program	175
	Server-Side Output	175
	Client-Side Output	177

	Putting It All Together: Walking through a BMP Entity Bean's Life Cycle	177
	Summary	180
Chapter 8	Writing Container-Managed Persistent Entity Beans	181
	Features of CMP Entity Beans	181
	CMP Entity Beans Are Subclassed	181
	CMP Entity Beans Have No Declared Persistent Fields	182
	CMP Get/Set Methods Are Defined in the Subclass	184
	CMP Entity Beans Have an Abstract Persistence Schema	186
	CMP Entity Beans Have a Query Language	187
	CMP Entity Beans Can Have ejbSelect() Methods	189
	Implementation Guidelines for Container-Managed Persistence	191
	Container-Managed Persistence Example: A Product Line	196
	Product.java	197
	ProductLocal.java	198
	ProductHome.java	198
	ProductLocalHome.java	200
	ProductPK.java	201
	ProductBean.java	203
	The Deployment Descriptor	207
	The Container-Specific Deployment Descriptor	210
	Client.java	212
	Running the Client Program	214
	The Life Cycle of a CMP Entity Bean	214
	Summary	216
Chapter 9	Introduction to Message-Driven Beans	217
	Motivation to Use Message-Driven Beans	217
	The Java Message Service	219
	Messaging Domains	220
	The JMS API	222
	Integrating JMS with EJB	226
	What Is a Message-Driven Bean?	227
	Developing Message-Driven Beans	231
	The Semantics	231
	A Simple Example	234
	Advanced Concepts	241
	JMS Message-Driven Bean Gotchas	244
	Message Ordering	245
	Missed ejbRemove() Calls	245
	Poison Messages	246
	How to Return Results Back to Message Producers	249
	The Future: Asynchronous Method Invocations	254
	Summary	254
Chapter 10	Adding Functionality to Your Beans	255
	Calling Beans from Other Beans	255
	Default JNDI Lookups	256
	Understanding EJB References	257

Resource Factories	259
Environment Properties	262
Understanding Handles	263
Home Handles	264
Summary	265
Part Three Advanced Enterprise JavaBeans Concepts	267
Chapter 11 EJB Best Practices	269
When to Use EJB	270
How to Choose a Web Application Framework to Work with EJB	272
Applying Model Driven Development in EJB Projects	275
Applying Extreme Programming in EJB Projects	277
Testing EJB	279
EJB Unit Testing	279
Use Frameworks for EJB Unit Testing	280
Implementing Client-Side Callback Functionality in EJB	282
JMS	282
Remote Method Invocation	283
Web Service	283
Choosing Between Servlets and Stateless Session Beans as Service Endpoints	284
Considering the Use of Aspect-Oriented Programming Techniques in EJB Projects	284
Aspect-Oriented Programming	285
When to Use AOP in EJB Applications	285
Reflection, Dynamic Proxy, and EJB	287
Deploying EJB Applications to Various Application Servers	288
Debugging EJB	290
Inheritance and Code Reuse in EJB	291
Writing Singletons in EJB	293
When to Use XML with EJB	293
When to Use Messaging Versus RMI-IIOP	294
Summary	297
Chapter 12 Transactions	299
Motivation for Transactions	300
Atomic Operations	300
Network or Machine Failure	301
Multiple Users Sharing Data	302
Benefits of Transactions	303
The ACID Properties	304
Transactional Models	306
Flat Transactions	306
Nested Transactions	308
Other Transactional Models	310
Enlisting in Transactions with Enterprise JavaBeans	310
Underlying Transaction System Abstraction	310
Declarative, Programmatic, and Client-Initiated Transactions	310
Choosing a Transaction Style	314

Container-Managed Transactions	317
EJB Transaction Attribute Values	318
Programmatic Transactions in EJB	324
CORBA Object Transaction Service	324
The Java Transaction Service	325
The Java Transaction API	325
Declarative versus Programmatic Transactions Example	328
Transactions from Client Code	330
Transactional Isolation	331
The Need for Concurrency Control	332
The Dirty Read Problem	334
The Unrepeatable Read Problem	336
The Phantom Problem	336
Transaction Isolation Summary	337
Isolation and EJB	338
Pessimistic and Optimistic Concurrency Control	339
Distributed Transactions	340
Durability and the Two-Phase Commit Protocol	340
The Transactional Communications Protocol	
and Transaction Contexts	342
Designing Transactional Conversations in EJB	343
J2EE Transaction Service and Extended Transactions	346
Summary	348
Chapter 13 Security	349
Introduction	350
Violations, Vulnerabilities, and Risk	351
Controls	351
Web Application Security	353
Authentication in Web Applications	354
Authorization	355
Confidentiality and Integrity	356
Understanding EJB Security	357
Authentication in EJB	357
Authorization in EJB	368
Security Propagation	377
Secure Interoperability	378
IIOP/SSL	379
CSIv2	379
Web Services Security	381
End-to-End Security	382
XML Digital Signature and XML Encryption	383
SAML	386
WS-Security	387
Summary	389
Chapter 14 EJB Timers	391
Scheduling	391
EJB and Scheduling	392

The EJB Timer Service	394
Timer Service API	395
Interaction between the EJB and the Timer Service	398
Timer Example: CleanDayLimitOrdersEJB	399
The CleanDayLimitOrders EJB Remote Interface	400
The CleanDayLimitOrders EJB Bean Class	400
The CleanDayLimitOrders EJB Home Interface	403
The CleanDayLimitOrders EJB Deployment Descriptor	403
The CleanDayLimitOrders EJB Client	404
Running the Client	405
Strengths and Limitations of EJB Timer Service	406
Summary	408
Chapter 15 BMP and CMP Relationships	409
The CMP and BMP Difference	410
Cardinality	411
1:1 Relationships	412
1:N Relationships	416
M:N Relationships	421
Directionality	428
Implementing Directionality with BMP	429
Implementing Directionality with CMP	430
Directionality May Not Map to Database Schemas	431
Bidirectional or Unidirectional?	433
Lazy Loading	433
Aggregation Versus Composition and Cascading Deletes	434
Relationships and EJB-QL	436
Recursive Relationships	437
Circular Relationships	438
Referential Integrity	439
Relationships, Referential Integrity, and Client Code	441
Summary	444
Chapter 16 Persistence Best Practices	445
Comparing Entity Beans with Other Persistence Approaches	446
Control	446
Data retrieval	446
Procedural versus Object-Oriented	447
Caching	448
Enforcement of Schema Independence	448
Migration	449
Rapid Application Development	449
Choosing Between CMP and BMP	450
Code Reduction and Rapid Application Development	450
Performance	450
Bugs	451
Control	451
Application Server and Database Independence	451
Relationships	452
Learning Curve and Cost	452

Choosing the Right Granularity for Entity Beans	453
Persistence Tips and Tricks	454
Beware the Object-Relational Impedance Mismatch	454
Hard-Coded versus Soft-Coded SQL	454
When to Use Stored Procedures	455
Normalizing and Denormalizing	457
Use Your EJB Object Model to Drive Your Data Model	459
Follow a Good Data Design Process	459
Use Surrogate Keys	460
Understand the Impacts of Database Updates	461
Versioning EJB Components	461
Living with a Legacy Database Design	463
Handling Large Result Sets	474
Summary	475
Chapter 17 EJB Integration	477
Why Does Integration Matter?	477
Integration Styles	478
EJB and Integration	479
J2EE Connector Architecture	480
Why J2EE Connectors?	480
Resource Adapter Interaction with J2EE Components	483
Resource Adapter Interaction with Application Server	484
The J2EE Connector API	486
The javax.resource Package	486
The javax.resource.cci Package	487
The javax.resource.spi Package	490
The javax.resource.spi.endpoint Package	492
The javax.resource.spi.security Package	493
The javax.resource.spi.work Package	493
System Contracts	494
Lifecycle Management	494
Connection Management	495
Security Management	498
Transaction Management	501
Work Management	504
Message In-flow	506
Connector Example: OutboundLoanRA	508
Example Architecture	508
JavaLoanApp.java	509
LoanApp.dll	510
OutboundLoanRA	511
LoanRatesEJB	535
LoanRatesClient	538
Running the Client	539
Extending OutboundLoanRA	541

Integration Best Practice: When to Use Which Technology	541
When to Use JMS and JMS-Based MDB	542
When to Use J2EE Connectors	542
When to Use Java Web Services	543
Summary	543
Chapter 18 EJB Performance Optimizations	545
It Pays to Be Proactive!	545
The Stateful Versus Stateless Debate from a Performance Point of View	547
How to Guarantee a Response Time with Capacity Planning	549
Use Session Façade for Better Performance	550
Choosing Between Local Interfaces and Remote Interfaces	552
Partitioning Your Resources	553
Tuning Stateless Session Beans	554
Tuning Stateful Session Beans	555
Tuning Entity Beans	556
Tuning Message-Driven Beans	563
Tuning Java Virtual Machine	563
Miscellaneous Tuning Tips	565
Choosing the Right EJB Server	567
Summary	568
Chapter 19 Clustering	569
Overview of Large-Scale Systems	569
What Is a Large-Scale System?	570
Basic Terminology	572
Partitioning Your Clusters	573
Instrumenting Clustered EJBs	578
How EJBs Can Be Clustered	578
The Concept of Idempotence	579
Stateless Session Bean Clustering	581
Stateful Session Bean Clustering	583
Entity Bean Clustering	584
Message-Driven Bean Clustering	588
Other EJB Clustering Issues	589
First Contact	589
Initial Access Logic	590
Summary	591
Chapter 20 Starting Your EJB Project on the Right Foot	593
Get the Business Requirements Down	593
Decide Whether J2EE Is the Right Choice	594
Staff Your Project	598
Design Your Complete Object Model	600
Implement a Single Vertical Slice	601
Choose an Application Server	603
Divide Your Team	604

	Invest in Tools	606
	Invest in a Standard Build Process	607
	Next Steps	607
	Summary	608
Chapter 21	Choosing an EJB Server	609
	J2EE Standard Compliance	610
	Pluggable JRE	610
	Conversion Tools	610
	Complex Mappings	611
	Third-Party JDBC Driver Support	611
	Lazy Loading	611
	Deferred Database Writes	611
	Pluggable Persistence Providers	611
	In-Memory Data Cache	612
	Integrated Tier Support	612
	Scalability	612
	High Availability	613
	Security	613
	IDE Integration	614
	UML Editor Integration	615
	Intelligent Load Balancing	615
	Stateless Transparent Fail-over	615
	Clustering	616
	Java Management Extension (JMX)	616
	Administrative Support	616
	Hot Deployment	617
	Instance Pooling	617
	Automatic EJB Generation	617
	Clean Shutdown	617
	Real-Time Deployment	618
	Distributed Transactions	618
	Superior Messaging Architecture	618
	Provided EJB Components	619
	Web Services	619
	Workflow	619
	Open Source	620
	Specialized Services	620
	Nontechnical Criteria	621
	Summary	621
Chapter 22	EJB-J2EE Integration: Building a Complete Application	623
	The Business Problem	623
	A Preview of the Final Web Site	624
	Scoping the Technical Requirements	630
	The Business Logic Tier	631
	The Presentation Tier	637
	Example Code	643
	Summary	651

Appendix A RMI-IIOP and JNDI Tutorial	653
Appendix B CORBA Interoperability	683
Appendix C Deployment Descriptor Reference	705
Appendix D The EJB Query Language (EJB-QL)	739
Appendix E EJB Quick Reference Guide	757
Index	801



Acknowledgments

This book has been a project spanning several years. Many have commented that the first edition was one of the best technical books they ever read. What's made this book a reality are the many people who aided in its development.

As a special thanks, we'd like to acknowledge the great folks at John Wiley & Sons. They have been absolutely outstanding throughout this book's evolution. In particular, we thank Bob Elliott, Sydney Jones, and Kathryn Malm for their incredible efforts. We also thank Floyd Marinescu of The Middleware Company for his insightful technical reviews, and Jörg Bartholdt of Xtradyne Technologies for technical discussions. Finally, we thank Theserverside.com community for providing us with their very helpful reviews.



Introduction

This book is a tutorial on Enterprise JavaBeans (EJB). It's about EJB concepts, methodology, and development. This book also contains a number of advanced EJB topics, giving you a practical and real-world understanding of the subject. By reading this book, you will acquire a *deep* understanding of EJB.

Make no mistake about it—what you are about to read is *not* easy. EJB incorporates concepts from a wealth of areas, including distributed computing, databases, security, component-driven software, and more. Combining them is a magnificent stride forward for the Java community, but with that comes a myriad of concepts to learn and understand. This book will teach you the concepts and techniques for authoring reusable components in Java, and it will do so from the ground up. You need only to understand Java to understand this book.

While you're reading this book, you may want to download the EJB specification, available at <http://java.sun.com/products/ejb/docs.html>.

Goals for This Edition

The first edition of this book came out in 1999, and the second edition in 2002. We had to make some tough calls when writing the second and third editions, and we are confident you'll like them. Here were our goals:

- **To update the book for EJB 2.1.** EJB 2.1 has many new useful features that we detail throughout the book.
- **To be broad and also deep.** We do not regurgitate the complete EJB specification in this book, nor do we cover every last detail of EJB. Rather, we cover the most important parts of EJB, leaving room to discuss advanced issues. For a complete reference while you are coding,

search through the EJB specification using Adobe Acrobat. Readers who are looking for a well-written book that is interactive and fun to read and covers the basics through advanced subjects have come to the right place.

- **To be concise.** Your time as a reader is extremely valuable, and you're likely waiting to read a stack of books besides this one. Given that most people don't have time to read 1,000-plus-page books, we actually wanted to reduce the size of this book as much as possible. So we've tightened things up and eliminated redundant examples. This way, you can get to actually program with EJB, rather than read a book for months on end. The irony of this story is that it was harder for us to write a shorter book than a long book!
- **To be a book for developers.** This book is not intended for high-level businesspeople. This is a technical book for a technical audience.
- **To write a book the right way.** This book's primary author, Ed Roman, has taken his skills in training and knowledge transfer and applied them to this book. Thus, we've infused this book with the following attributes:
 - **A conversational style.** When you read this book, sometimes you'll feel like you're almost having a discussion with us. We think this is far superior to spending eons trying to re-read a formal writing style over and over again.
 - **Use of diagrams and bulleted lists.** The adage "a picture is worth a thousand words" applies here. These tactics are great for breaking up blocks of text. They keep things varied and make the book a much faster read.
 - **A consistent voice.** Even though several co-authors wrote this book, you'll hear one voice. This was done to combine best-of-breed knowledge from several expert co-authors while maintaining a uniform look and feel throughout the book.
- **To be an introductory book, but also to get quickly into advanced topics.** We figured that the average developer has had enough of books that merely skim the surface. We wanted to write a book that pushed beyond the basics. Our approach when writing this book was always to err on the side of being advanced. To achieve this, we did an immense amount of research. We participated in the mailing lists, performed many real-world projects, attended conferences and seminars, and networked with the top experts throughout the world.
- **To be vendor-neutral.** All vendor-specific deployment steps are externalized to the book's accompanying source code. This makes this book useful for any EJB server.

- **To add useful EJB information garnered from our instructor-led training classes.** Having taught EJB/J2EE for years, we have learned significantly from our students. We have interlaced this book with many of our own students' questions and answers in relevant sections.
- **To take all the source code and make it available online.** Because we've made the code available on the Web, you know it's the latest version. This will ensure that the code you receive works right the first time.

Organization of the Book

The text is organized into the following five parts:

- **Part One** is a whirlwind introduction to EJB programming. Part One serves as a great overview for people in a hurry. While Part One is essential information to EJB newcomers, veterans will also find nuggets of useful knowledge. The following chapters are covered:
 - **Chapter 1** is a tour of enterprise computing. We'll talk about components, service-oriented architectures, distributed computing frameworks, and containers. In this regard, we'll introduce EJB and J2EE.
 - **Chapter 2** moves on to the fundamentals of building an EJB system, including the tricky concept of request interception. We'll also look at the various source code files that make up an enterprise bean.
 - **Chapter 3** shows you how to put together a simple enterprise bean. We'll also learn how JNDI is used in EJB and see how to call that bean from a client.
- **Part Two** devotes exclusive attention to programming with EJB. We'll see how to use the triad of beans: entity beans, session beans, and message-driven beans. We'll cover the basics of writing each type of bean, including an example as well as detailed lifecycle diagrams.
 - **Chapter 4** introduces session beans. We'll look at the difference between stateful and stateless session beans, how to code a session bean, and what's going on behind the scenes with session beans.
 - **Chapter 5** shows how Web services can be implemented using EJB. In particular, we show how a stateless session bean can be made available as a Web service.
 - **Chapter 6** is a conceptual introduction to entity beans. We'll look at persistence concepts, what makes entity beans unique, and the files involved when building entity beans.

- **Chapter 7** covers bean-managed persistent (BMP) entity beans. We'll see how to program a BMP entity bean, and also look at what's happening behind the scenes with BMP.
- **Chapter 8** covers container-managed persistent (CMP) entity beans. We'll focus on the exciting features of EJB 2.x CMP, learn how to program a CMP entity bean, and look at what's happening behind the scenes with CMP.
- **Chapter 9** covers message-driven beans. We'll first review the Java Message Service (JMS), which is the messaging framework used mostly with message-driven beans. We'll then dive in and understand how to program with message-driven beans.
- **Chapter 10** discusses the EJB environment, along with services provided by the container. This includes environment properties, resource factories, references between beans, and handles.
- **Part Three**, the most exciting part of the book, covers advanced EJB concepts. The following chapters are included:
 - **Chapter 11** explains guidelines for using various Web application frameworks, model-driven development tools, and so on, in EJB applications. It also presents proven best practices for EJB design, development, and testing.
 - **Chapter 12** tackles transactions. Transactions are a crucial topic for anyone building an EJB application that involves state. We'll discuss transactions at a conceptual level followed by a discussion on how to apply them to EJB. We'll also learn about the Java Transaction API (JTA) as well as J2EE Extended Transaction concepts.
 - **Chapter 13** provides an in-depth treatment of EJB security and covers *Java Authentication and Authorization Service* (JAAS), secure interoperability, and Web Services security.
 - **Chapter 14** introduces the new EJB timer service that lets you schedule tasks for automatic execution.
 - **Chapter 15** covers relationships between entity beans. This is a critical concept for anyone performing complex persistence. We'll understand the concepts of cardinality, directionality, referential integrity, and cascading deletes. We'll also see how to code relationships for *both* CMP and BMP entity beans.
 - **Chapter 16** covers persistence best practices. You'll learn exciting concepts such as how to choose between entity beans and other persistence techniques, how to choose between BMP and CMP, and you'll survey a collection of persistence best practices that we've assembled from our knowledge and experience.

- **Chapter 17** covers integration to and from EJB platform in-depth. It provides introduction to the various styles of integration, followed by a discussion of various techniques for integrating EJB with the outside world. It explains the J2EE Connector Architecture, a predominant framework for integrating EJB with back-end enterprise applications, and discusses a connector example.
- **Chapter 18** covers EJB tips and techniques for designing and deploying EJB for better performance. You'll learn about design strategies that will help you make decisions such as when to choose between stateful versus stateless session beans, when to choose between local and remote interfaces, and so on. The chapter also focuses a great deal on providing performance tuning tips for different types of beans.
- **Chapter 19** discusses clustering in large-scale EJB systems. You'll learn about how clustering works behind the scenes and learn a few strategies for how containers might achieve clustering. This is a critical topic for anyone building a system that involves several machines working together.
- **Chapter 20** covers EJB project management. We'll talk about how to get your project started on the right foot. This includes guidelines on choosing between J2EE and .NET frameworks for your projects, building a first pass of your system, dividing your development team, and many such concepts.
- **Chapter 21** provides guidelines for choosing the right EJB server for your needs. We'll describe our methodology for how an organization can compare and contrast different vendors' offerings. We'll also list our set of criteria for what we would want in an EJB server.
- **Chapter 22** shows how to build a real-world J2EE system using EJB components. We'll see how the EJB components should be used *together* in an enterprise, as well as how to connect them with clients using Java servlets and JavaServer Pages (JSP) technology. We'll also demonstrate how to design an EJB object model using UML.
- **The Appendixes** are a collection of ancillary EJB topics. Some developers may want to read the appendices, while some may not need to do so.
 - **Appendix A** teaches you Java Remote Method Invocation over the Internet Inter-ORB Protocol (RMI-IIOP) and the Java Naming and Directory Interface (JNDI). These technologies are prerequisites for using EJB. If you're just starting down the EJB road, you must read this appendix first.
 - **Appendix B** discusses how to integrate EJB and CORBA systems. We'll learn about how EJB and CORBA are interoperable through

RMI-IIOP and see sample code for calling an EJB component from a CORBA client.

- **Appendix C** is a deployment descriptor reference guide. This will be useful for you later, when you're working with deployment descriptors and need a guide.
- **Appendix D** covers the EJB query language (EJB-QL) in detail.
- **Appendix E** is an API and diagram reference guide. This is useful when you need to look up the purpose of a method or class in EJB.



Throughout the book, this icon will signal a tip, note, or other helpful advice on EJB programming.

In a similar paradigm to our training courses, the content of this book is very interactive. We have taken our knowledge of adult learning and scattered boxes like this throughout the book. Each box asks you a question to get you thinking. The answers to the questions are posted on the book's accompanying Web site. What do you think are the benefits of this paradigm?

Illustrations in the Text

Almost all of the illustrations in this book are written in the Unified Modeling Language (UML). UML is the de facto standard methodology for illustrating software engineering concepts in an unambiguous way. If you don't know UML, pick up a copy of *The Unified Modeling Language User Guide* (Addison-Wesley, ISBN 0201571684), which illustrates how to effectively use UML in your everyday software. UML is a highly important achievement in object-oriented methodology. It's a common mechanism for engineers to communicate and design with, and it forces you to abstract your object model prior to implementation. We cannot stress its use enough.

The Accompanying Web Site

This book would not be complete without a way to keep you in touch after it was published. A Web site is available for resources related to this book. There you'll find:

- All of the source code you see in this book. The code comes complete with Ant scripts, ready to build and run. It should be deployed on any application server that is J2EE 1.4-compliant.
- Updates to the source code examples.

- Error corrections from the text.
- A PDF copy of this book

The Web site is at www.wiley.com/compbooks/roman.

Feedback

When you begin your EJB programming, we're sure you'll have many experiences to share with other readers. Feel free to e-mail examples, case studies, horror stories, or tips that you've found helpful in your experiences, and we'll post them on the Web site.

Send bug reports to books@middleware-company.com.

From Here

Now that we've gotten the logistics out of the way, let's begin our exploration of Enterprise JavaBeans.

About the Authors

Ed Roman is one of the world's leading authorities on high-end middleware technologies. He has been heavily involved with Sun Microsystems' enterprise Java solutions from their inception and has designed, built, and deployed a variety of enterprise applications, including architecting and developing complete application server products. He devotes a significant amount of time to influencing and refining Sun's enterprise specifications, contributes regularly to middleware interest mailing lists, and regularly speaks at middleware-related conferences.

Ed is the founder of The Middleware Company (which can be found on the Web at www.middleware-company.com). The Middleware Company offers the world's leading knowledge network for middleware professionals. The Middleware Company enables developers, technology vendors, and enterprises to implement, innovate, and communicate emerging technology offerings. The Middleware Company solutions include TheServerSide Communities, MiddlewareREACH, and MiddlewarePRO. TheServerSide Communities inform over half a million professionals monthly using an open forum to discuss and solve the most challenging middleware issues. Clients of The Middleware Company include the world's top software organizations including BEA Systems, Compuware, HP, IBM, Microsoft, Oracle, Sun Microsystems, and VERITAS Software. Ed also is the founder of TheServerSide.com, which is the de facto J2EE community Web site. Every day, thousands of developers get

together on TheServerSide.com to share EJB design patterns, hear about the latest EJB news, ask and answer EJB development questions, and read articles. After you've read this book, visit TheServerSide.com to catch up on the latest EJB information. TheServerSide.com is a completely free service and is intended to help the community.

Rima Patel Sriganesh is a Member of Technical Staff presently working in the Technology Outreach group at Sun Microsystems, Inc. She specializes in Java, XML, and Integration platforms. Her areas of technology passion include Distributed Computing Models, Security and Trust Computing, Semantic web, Grid Computing, and Quantum Physics. She speaks regularly at premiere industry conferences such as JavaOne, Web Services Edge, SIGS 101, Sun Technology Days, and others. She also represents Sun at various security, choreography, and financial services technology standards.

Rima is a co-author of *Developing Java Web Services* (Wiley, 2002). She frequently publishes her take on technology and industry in the form of papers and blogs.

Rima graduated in Mathematics from M. S. University, Gujarat, India. She currently lives with her husband in the Greater Boston area.

To find out more about her work, use the Google queries "Rima Patel" Sun Microsystems or "Rima Patel Sriganesh."

Gerald Brose works as Security Software Architect at Xtradyne Technologies. Gerald is an expert in middleware security, including CORBA, J2EE, and Web Services. He is a regular speaker at international conventions and the author of several publications on middleware security and related issues. Gerald is a co-author of *Java Programming with CORBA* (Wiley, 2001).

As a member of the open source community, Gerald maintains JacORB, the most widely used Open Source ORB for Java, which is also part of the JBoss J2EE application server. Gerald holds a Ph.D. in computer science from Freie University, Berlin. He lives with his wife and two sons in Berlin, Germany.



Overview

In Part One, we introduce the server-side development platform, the *Java 2 Platform, Enterprise Edition* (J2EE), of which the *Enterprise JavaBeans* (EJB) component architecture is a vital piece. J2EE is a conglomeration of concepts, programming standards, and innovations—all written in the Java programming language. With J2EE, you can rapidly construct distributed, scalable, reliable, and portable secure server-side deployments.

- **Chapter 1** begins by exploring the need for server-side component architecture such as EJB. You'll see the rich needs of server-side computing, such as scalability, high availability, resource management, and security. We'll discuss how EJB architecture relates to the Service-oriented Architecture (SOA) paradigm. We'll also take a look at the J2EE server-side development platform.
- **Chapter 2** moves on to the fundamentals of Enterprise JavaBeans. We'll look at the concept of *request interception*, which is crucial for understanding how EJB works. We'll also look at the different files that go into a bean and how they work together.
- **Chapter 3** gets down and dirty with EJB programming. Here, we'll write our first simple bean. We'll explain how to code each of the files that compose the bean, and we'll also look at how to call that bean from clients.

CHAPTER 1 Overview

Enterprise JavaBeans (EJB) is a server-side component architecture that simplifies the process of building enterprise-class distributed component applications in Java. By using EJB, you can write scalable, reliable, and secure applications without writing your own complex distributed component framework. EJB is about rapid application development for the server side; you can quickly and easily construct server-side components in Java by leveraging a prewritten distributed infrastructure provided by the industry. EJB is designed to support application portability and reusability across any vendor's enterprise middleware services.

If you are new to enterprise computing, these concepts will be clarified shortly. EJB is a complicated subject and thus deserves a thorough explanation. In this chapter, we'll introduce EJB by answering the following questions:

- What plumbing do you need to build a robust distributed object deployment?
- What is EJB, and what value does it add?
- How does EJB relate to SOA?
- Who are the players in an EJB ecosystem?

Let's kick things off with a brainstorming session.

The Motivation for Enterprise JavaBeans

Figure 1.1 shows a typical business application. This application could exist in any vertical industry and could solve any business problem. Here are some examples:

- A stock trading system
- A banking application
- A customer call center
- A procurement system
- An insurance risk analysis application

Notice that this application is a *distributed system*. We broke up what would normally be a large, monolithic application and divorced each layer of the application from the others, so that each layer is completely independent and distinct.

Take a look at this picture, and ask yourself the following question based purely on your personal experience and intuition: *If we take a monolithic application and break it up into a distributed system with multiple clients connecting to multiple servers and databases over a network, what do we need to worry about now* (as shown in Figure 1.1)?

Take a moment to think of as many issues as you can. Then turn the page and compare your list to ours. Don't cheat!

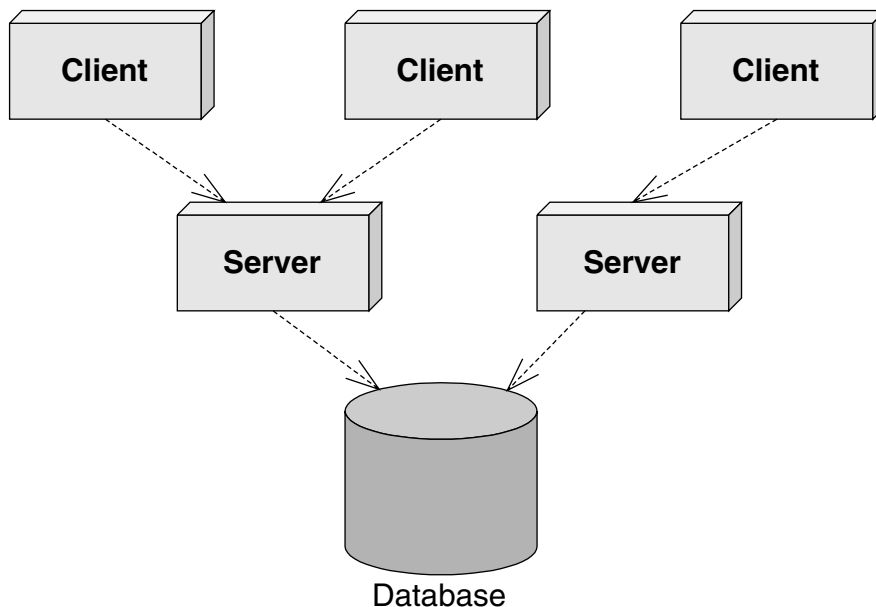


Figure 1.1 Standard multitier-only deployment.