

Cambridge University Press  
978-0-521-77849-7 - Building Application Servers  
Rick Leander  
Frontmatter  
[More information](#)

---

# *Building Application Servers*

**Advances in Object Technology Series**

Dr. Richard S. Wiener, Series Editor  
and Editor-in-Chief of  
*Journal of Object-Oriented Programming*  
SIGS Publications, Inc.  
New York, New York  
and  
Department of Computer Science  
University of Colorado  
Colorado Springs, Colorado

1. Object Lessons: Lessons Learned in Object-Oriented Development Projects  
• *Tom Love*
2. Objectifying Real-Time Systems • *John R. Ellis*
3. Object Development Methods • *edited by Andy Carmichael*
4. Inside the Object Model: The Sensible Use of C++ • *David M. Papurt*
5. Using Motif with C++ • *Daniel J. Bernstein*
6. Using CRC Cards: An Informal Approach to Object-Oriented Development  
• *Nancy M. Wilkinson*
7. Rapid Software Development with Smalltalk • *Mark Lorenz*
8. Applying OMT: A Practical Step-By Step Guide to Using the Object Modeling  
Technique • *Kurt W. Derr*
9. The Smalltalk Developer's Guide to VisualWorks • *Tim Howard*
10. Objectifying Motif • *Charles F. Bowman*
11. Reliable Object-Oriented Software: Applying Analysis & Design • *Ed Seidewitz &  
Mike Stark*
12. Developing Visual Programming Applications Using Smalltalk • *Michael  
Linderman*
13. Object-Oriented COBOL • *Edmund C. Arranga & Frank P. Coyle*
14. Visual Object-Oriented Programming Using Delphi • *Richard Wiener & Claude  
Wiatrowski*
15. Object Modeling and Design Strategies: Tips and Techniques • *Sanjiv Gossain*
16. The VisualAge for Smalltalk Primer • *Liwu Li*
17. Java Programming by Example • *Rajiv Sharma & Vivek Sharma*
18. Rethinking Smart Objects: Building Artificial Intelligence with Objects  
• *Daniel W. Rasmus*
19. The Distributed Smalltalk Survival Guide • *Terry Montlick*
20. Java for the COBOL Programmer • *E. Reed Doke, Ph.D. and Bill C. Hardgrave, Ph.D.*
21. Building Application Servers • *Rick Leander*

*Additional Volumes in Preparation*

Cambridge University Press  
978-0-521-77849-7 - Building Application Servers  
Rick Leander  
Frontmatter  
[More information](#)

---

# *Building Application Servers*

Rick Leander



Cambridge University Press  
978-0-521-77849-7 - Building Application Servers  
Rick Leander  
Frontmatter  
[More information](#)

---

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE  
The Pitt Building, Trumpington Street, Cambridge, United Kingdom

CAMBRIDGE UNIVERSITY PRESS

The Edinburgh Building, Cambridge CB2 2RU, UK  
40 West 20th Street, New York, NY 10011-4211, USA  
10 Stamford Road, Oakleigh, Melbourne 3166, Australia  
Ruiz de Alarcón 13, 28014 Madrid, Spain

[www.cup.cam.ac.uk](http://www.cup.cam.ac.uk)  
[www.cup.org](http://www.cup.org)

Published in association with SIGS Books

© 2000 Cambridge University Press

All rights reserved.

This book is in copyright. Subject to statutory exception  
and to the provisions of the relevant collective licensing agreements,  
no reproduction of any part may take place without  
the written permission of Cambridge University Press.

Any product mentioned in this book may be a trademark of its company.

First published in 2000

Design and composition by Andrea Cammarata  
Cover design by Andrea Cammarata

Printed in the United States of America

*A catalog record for this book is available from the British Library.*

*Library of Congress Cataloging-in-Publication Data is on record with the publisher.*

ISBN 0 521 77849 2 paperback

Cambridge University Press  
978-0-521-77849-7 - Building Application Servers  
Rick Leander  
Frontmatter  
[More information](#)

---

*To Barb*

## Contents

<b>Acknowledgments</b>	xix
<b>Introduction</b>	xxi
Who Should Read This Book	xxi
Organization	xxii
Part 1—Architecture	xxii
Part 2—Design	xxiii
Part 3—Programming	xxiii
How to Get the Program Code	xxiii
<b>PART 1 • ARCHITECTURE</b>	1
<b>Chapter 1: What Is an Application Server and Why Do I Need One?</b>	3
Two-Tiered vs. Multi-Tiered Computing	3
Why I Chose Multi-Tiered Client/Server	4
What Can an Application Server Do?	6
Scalability	7
Distributed processing	7
Reusable business objects	8
Business rule processing	8
Cross-platform integration	9
Costs and Disadvantages of Application Servers	9
Long-term commitment	10
Middleware acquisition	10
New ways to twist the brain	11
The end of the coding cowboy	11
Software reuse	12
Moving from Traditional Client/Server to N-Tier Computing	12

viii *Building Application Servers*

---

Summary	13
References	14
<b>Chapter 2: Anatomy of an Application Server</b>	15
Overview of the Application Server Architecture	16
Middleware: The Glue That Holds It Together	18
Middleware Categories	19
Remote database protocols	21
Remote procedure calls	21
Distributed objects	23
Transaction processing monitors	26
Message brokers	27
Commercial application servers	27
Applying Middleware to the Application Server Architecture	28
Your best face forward: presenting a clean application interface	29
Business objects: modeling your business in software	30
Persistence: talking to the database	32
Alternative Application Server Architectures	34
The fourth layer	34
Data-centric application servers	34
Web server-based approaches	37
Putting It All Together	38
Summary	38
References	39
 <b>PART 2 • DESIGN</b>	 41
<b>Chapter 3: Designing Application Servers</b>	43
Joint Application Design	44
Business Object Design	45
Modeling business processes	46
Reuse	46
Design standards	47
Iterative Development	47
Why combine design and programming?	48

	Contents	ix
Self-directed technical review	48	
Design Constraints	49	
Layered design	49	
Middleware matters	50	
Integrating existing applications	51	
A Brief Introduction to UML Notation	51	
Diagrams and symbols	52	
Use case diagrams	53	
Class diagrams	53	
Sequence diagrams	58	
Meeting the End User's Needs	61	
Summary	62	
References	63	
<b>Chapter 4: Service Interface Design</b>	<b>65</b>	
What Is a Service Interface?	66	
Design by Interface	66	
More on JAD: Developing Use Cases	68	
Describe the context	69	
Describe the actors	71	
Describe the procedure	71	
Describe exceptions	72	
Use common language	73	
Iterate and refine	73	
A brief example	74	
Making use cases work	76	
Turning Use Cases into Services	76	
The service is application-specific	78	
The service is self-contained	79	
The service handles all exceptions	80	
The service hides the business object layer	81	
The service conforms to standards	82	
Bundling services into interfaces	83	
Handling Errors and Exceptions	84	
User interface errors	85	



<b>x</b>	<b><i>Building Application Servers</i></b>	
	Application errors	86
	System and network errors	87
	Exceptions and interface design	87
	Summary	88
	References	89
	Further Reading	89
	<b>Chapter 5: Designing Business Objects</b>	<b>91</b>
	Moving from Interfaces to Objects	92
	From data models to business objects	92
	Choosing a design approach	93
	What Exactly Is a Business Object?	94
	Finding the Objects in Your Business	95
	Defining the Objects	98
	Designing the Objects	101
	Attributes	101
	Methods	102
	States	103
	Events	104
	Business object specifications	105
	Object Interaction	107
	Aggregation	107
	Generalization and specialization	109
	Association	112
	Collections	114
	Creating the class diagram	115
	Application Server Issues and Constraints	116
	Short business cycles	116
	Reuse	117
	Concurrency and synchronization	119
	Repositories	119
	Persistence	120
	Linking Business Objects to the Service Interface	120
	Developing sequence diagrams	121
	Creating new business objects	122

	Contents	xi
Implementing services	123	
Business Object Architecture	125	
Summary	126	
References	127	
Further Reading	127	
<b>Chapter 6: Designing the Persistent Object Layer</b>	<b>129</b>	
The Role of the Persistence Layer	130	
Relational Database Principles	132	
Database history	132	
The relational data model	133	
Structured query language (SQL)	134	
Database middleware	136	
Designing a Persistent Object Layer	137	
Persistence layer example	138	
Generalized object servers	140	
Tracking the objects	143	
Objects and relational databases	144	
Scalability	149	
Using Object-Oriented Databases	152	
Using Objects to Represent External Applications	154	
Summary	154	
References	155	
Further Reading	155	
<b>Chapter 7: Integrating Existing Systems and Legacy Software</b>	<b>157</b>	
Design Issues for Application Integration	158	
What Do We Have—Application Mining	159	
Turning Subroutines into Services	160	
Proxy Objects	160	
How to access remote software	162	
Input and Output Streams	164	
Message-oriented middleware	165	
Advanced sneaker net	167	
Accessing Application Databases	168	

<b>xii</b>	<b><i>Building Application Servers</i></b>	
	Direct database access	168
	Replication	169
	Synchronizing Transactions	170
	Fun with Punch Cards: What to Do with Legacy Software	171
	Summary	172
	References	173
	Further Reading	173
	 <b>PART 3 • PROGRAMMING</b>	 175
	<b>Chapter 8: Implementing an Application Server Framework</b>	177
	The Application Server Framework	178
	Initializing the framework	178
	Processing service requests	180
	Commercial frameworks	182
	Choosing a framework strategy	182
	Additional Framework Requirements	183
	Scalability	183
	Concurrency	184
	Security	184
	Fault tolerance	185
	Development Strategies	185
	Communications support	186
	Development environment	187
	Tools	188
	Training	192
	Metrics	194
	Summary	194
	References	195
	 <b>Chapter 9: Using Java to Build Business Objects</b>	 197
	Using Java to Illustrate Programming Principles	198
	Overview of the Distributed Java Architecture	198
	Object-Oriented Programming in Java	201
	Java class definitions	201

	Contents	xiii
Class composition in Java	203	
Class association in Java	203	
Class generalization in Java	206	
Coding Guidelines in Java	206	
Using Interfaces to Package Objects	207	
Distributing Java Objects with RMI	211	
Creating the remote interface	211	
Creating a remote object	212	
Creating the stub and skeleton	214	
Registering the remote object	214	
Accessing the remote object	217	
Comparing Distributed Java with Other Middleware Architectures	219	
Distributed objects in CORBA	219	
Microsoft's DCOM	221	
Summary	223	
References	224	
Further Reading	224	
<b>Chapter 10: Persistent Objects: Communicating with Databases</b>	<b>227</b>	
An Overview of JDBC	228	
JDBC architecture	228	
SQL basics	230	
Basic JDBC programming	233	
Other database middleware	238	
Creating a Persistent Object Framework	238	
A Simple Persistent Object Server	239	
Joining business objects with relational databases	240	
Tracking business objects	245	
Serving up customer objects	248	
Extending the Simple Object Server	250	
Adding more objects	250	
Serving up multiple objects from the same query	251	
Serving up complex objects	255	
Optimizing the Persistence Layer	258	
Capacity planning	259	

<b>xiv</b>	<b><i>Building Application Servers</i></b>	
	Minimizing database connections	259
	Distributing business objects	262
	Concurrency and synchronization	263
	Optimizing throughput	266
	Summary	266
	References	267
	Further Reading	267
	<b>Chapter 11: Interfaces and Client-Side Communication</b>	<b>269</b>
	Client/Server Communication	270
	Establishing remote communication	271
	Processing remote communication	272
	Server-side communication	272
	Requesting services	274
	Creating a Service Interface	274
	Defining the service interface	275
	Implementing the interface	276
	Registering the service interface	279
	Using the Service Interface	280
	Accessing the services	281
	Locating the data	282
	Storing the data	284
	Releasing the remote object	285
	Passing Data, Objects and Properties	286
	Primitives	286
	Objects	286
	Properties	288
	Returning errors	290
	Messages, events, and asynchronous communication	294
	Summary	295
	References	295
	Further Reading	296
	<b>Chapter 12: Enforcing Business Rules</b>	<b>297</b>
	What is a Business Rule?	298

	Contents	xv
Turning Business Rules into Code	299	
Structure-based rules	300	
Rules in code	301	
Rules in data	306	
Classification	313	
Maintaining rule and classification tables	316	
Where to Put the Code	317	
User interface	318	
Service interface	319	
Business objects	320	
Persistence	321	
Database server	321	
Standardized Error Handling	322	
Standardized messages	322	
Exception objects	322	
Message handling	323	
Error logs	324	
Commercial Business Rule Engines	325	
Security and Authorization Strategies	326	
Organizing security rules	327	
Where to implement security	328	
Summary	329	
References	330	
Further Reading	330	
<b>Chapter 13: Multiprocessing, Concurrency, and Transactions</b>	<b>331</b>	
The Trouble with Multiprocessing	332	
Multi-tasking	333	
Multi-threading	334	
Multiple objects	335	
Multiple, synchronized data	336	
Multiple data sources	336	
Multiprocessing Within the Application Server	337	
User interfaces	338	
Service interfaces	338	

<b>xvi</b>	<b><i>Building Application Servers</i></b>	
	Business objects	338
	Persistent objects	339
	Database servers	339
	The Class Factory Model	340
	Applying the class factory model	341
	Creating the class factory object	342
	Using the class factory	345
	When to use the class factory	346
	Multi-Threading	346
	Implementing multi-threading	347
	Synchronizing execution	353
	Synchronizing Objects and Data	353
	Locking at the database level	354
	Locking at the object level	355
	Locking at the persistence level	357
	Resolving deadlocks	358
	Transactions	360
	Transaction basics	360
	Implementing transaction objects	361
	Commit or rollback	363
	Two-phase commit	365
	Commercial transaction monitors	366
	Summary	367
	References	368
	Further Reading	368
	<b>Chapter 14: The Next Generation of Business Applications</b>	<b>369</b>
	Clues from the Past	371
	Increased automation	372
	Ease of use	373
	Business intelligence	374
	Communications	375
	How much farther can we go?	376
	Emerging Component Standards	376
	Microsoft's Distributed Internet Architecture	377

	Contents	xvii
Enterprise JavaBeans	379	
CORBA object monitors	380	
Other contenders	380	
The Application Software Marketplace	382	
Off the shelf applications	383	
The component marketplace	384	
The open source bazaar	385	
The Emerging Business Platform	387	
Cheap computers	387	
Palm-tops and cell phones	388	
Pervasive computing	389	
Where is it all going?	390	
Final Thoughts	390	
References	392	
<b>Appendix: Setting up a Development Environment</b>	<b>393</b>	
Development Using a Single Computer	394	
Hardware requirements	395	
Software requirements	395	
Development on the Network	396	
Network hardware	396	
Software	397	
Compiling and Testing Java and RMI	398	
Step 1: set up a project directory	398	
Step 2: compile the server and applet	399	
Step 3: use rmic to create the stub and skeleton classes	399	
Step 4: start the Web server and RMI registry	400	
Step 5: start the application server	400	
Step 6: run the applet	401	
Running on a network	401	
Where to get help	402	
Setting up JDBC	402	
Summary	403	
Sources for Software	403	
<b>Index</b>	<b>405</b>	



## **Acknowledgments**

Many thanks to Lothlórien Homet and the people at SIGS and Cambridge University Press for all of their guidance and help in making this book possible. Thanks also to technical editor Lisa McCumber for her many insights and to copy editor Matt Lusher, who transformed my ramblings into readable prose. Thanks also to Dr. James Gerlach at the University of Colorado at Denver for his excellent course, Distributed Object Computing, that sparked my interest in middleware and distributed processing. Thanks also to RFB&D for quickly providing reference materials. Finally, special thanks to Barb, my wonderful wife, for her encouragement, support, and assistance.

## Introduction

---

You've read everything you can find about middleware, CORBA, transaction monitors, message brokers, enterprise JavaBeans, and other distributed technologies. Now it's time to put them to work. Time to build your company's first multi-tiered application. But where do you start? How do you structure the programs? How do you distribute the code? What about integrating existing applications and databases? This was the problem that I faced as I began working with multi-tiered development. There was plenty of information on the tools and technologies, but little on how to make them work in a business setting.

Application servers and related technologies offer great promise and potential for solving the issues that trouble corporate computing. Problems like scalability, application integration and code reuse. But before we can solve these grand problems, we have to figure out how to use the technology. How do we process orders, ship products, bill customers, approve loan applications and pay insurance claims.

My hope is that this book will offer some guidelines to start you on your way. Instead of focusing on middleware, the emphasis is on the design issues and programming techniques necessary to create an overall business application framework. The approach is user-centric, relying on joint development between developers and business people, using short, iterative design-program-review cycles. Object-oriented development is also stressed using designs illustrated with UML and programming examples written for the Java platform. Although Java and RMI are used, the framework will work with almost any language or distributed object platform.

## Who Should Read This Book

This book is primarily intended for software developers, the designers and programmers who have to take these new technologies and turn them into business solutions. It is written at a moderate technical level

and assumes that you, the reader, are familiar with client/server or main-frame development in a business environment. You do not need to understand middleware, object-oriented programming, or be a Java programming wizard, but you should be familiar with relational databases, user interface design and be able to read and understand program code. For those not familiar with some of the more technical topics, such as UML and distributed processing, the book provides enough background to get you started, then suggests additional references to fill in the details that are beyond the scope of this book.

Although the book is intended for software developers, the first two sections will be useful to business people working in a joint development team environment. These sections offer background on the development process and introduce the tools needed to create an effective design. Joint application design, use cases, and iterative development are concepts that must be understood by all team members. Managers can also read through these chapters to gain a better understanding of the benefits of the technology and the overall design process. Other information technology workers, such as network and database administrators, can also benefit from this book by gaining an understanding of these new technologies and processes.

## Organization

To fully understand application server technology, it must be examined from several different perspectives, first from a high level view, on to the user's perspective and finally to the programmer's vantage point. Not only does each perspective show different aspects of the technology, the three separate perspectives allow you, the reader, to ease into the many details that must be considered before you can understand how to make the technology work.

### Part 1—Architecture

The book begins by examining what an application server is and how it can benefit the business. Benefits and drawbacks are listed, followed by a general overview of the technology. Once these are understood, the three layers of the application framework, the service interface, business objects and persistent layer, are discussed in general terms.

---

## Part 2—Design

The design view looks at the application framework from a user-centric view, examining how the layers perform business functions. The emphasis in this section is on specifying the business requirements through use cases and then creating a software design that will meet these needs.

## Part 3—Programming

Once these layers are examined from a user-centric business perspective, the programming section examines each layer in even greater detail, offering techniques that can be used to create the program code that will perform the tasks specified during software design.

## How to Get the Program Code

The source code for the program examples, as well as the full implementation of each program, can be downloaded from the Cambridge University Press site:

<http://www.cup.org/Titles/77/0521778492.html>

In addition to this site, the files can also be obtained from my personal Website at:

<http://pages.prodigy.net/rleander>

Once expanded, the files are distributed into directories by chapter, with program listings in the main directory and additional program code included in subdirectories underneath each chapter directory. Check the `readme.txt` file included in the primary directory for additional information.