100%

ONE HUNDRED PERCENT

COMPREHENSIVE
AUTHORITATIVE
WHAT YOU NEED

ONE HUNDRED PERCENT

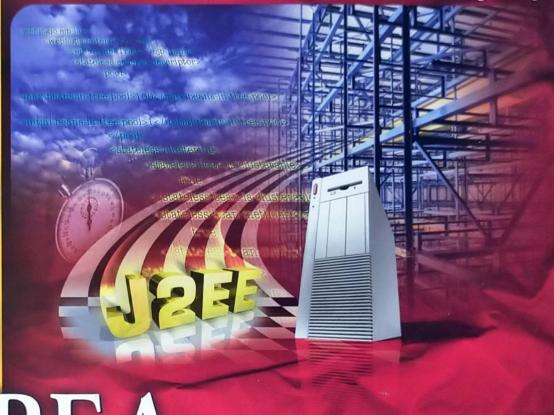
Discover powerful new WebLogic Server 7.0 features including Web services and WebLogic Workshop

Build enterprise applications that take advantage of WebLogic's strong J2EE™ standard support

Lickstart your productivity with examples, hints, and insight from top BEA experts

"I really enjoyed reading the WebLogic Bible. I've been using WebLogic off and on since before EJB existed, and I still learned a bunch of new tricks."

- Rick M. Hightower, Author of Java Tools for Extreme Programming



ebLogic Server

Bible Edition

Joe Zuffoletto and Lov Miranda

COMPANION WEB SITE

Features the author's sample code from the book

# BEA WebLogic Server Bible

2nd Edition

Joe Zuffoletto and Lou Miranda



GIFT OF THE ASIA FOUNDATION NOT FOR RE-SALE

QUÀ TẶNG CỦA QUỸ CHÂU Á KHÔNG ĐƯỢC BÁN LẠI





Wiley Publishing, Inc.

# **Foreword**

The computer industry innovates at an unprecedented pace producing multiple technological revolutions — involving changes so dramatic they shift the entire industry — within a relatively short time period. Thirty years ago, computers were enormous, complex, and expensive, making them only economically feasible to large corporations, universities, and government. With the personal computer revolution, that same technology has arrived in schools, homes, and nearly every business. Application software such as Microsoft's Office and its predecessors rode this revolution and dramatically increased productivity in almost every industry. More recently, we've experienced the network revolution. Instead of computers exchanging information via floppy disks and isolated private networks, we've built a worldwide distributed network bridging disparate systems via common protocols such as TCP/IP and HTTP. Just as word processors and spreadsheets became vital office tools, computer networking both in the office and across the Internet has transformed how we work and live.

Each major revolution inspires many smaller technological changes, producing a drastically different computing landscape. During the Internet revolution, Sun Microsystems released its Java programming language to a world of eager programmers. Until this time, most systems programming was done in C or C++, and every programmer endured countless hours debugging memory problems or porting to different architectures. Java rode the sweeping Internet changes and brought garbage collection, safe programming, and multi-platform support into widespread use. Along with Sun's J2EE APIs, BEA WebLogic Server brought the ideas of Java to server-side enterprise computing. Instead of writing low-level system code or dealing with proprietary interfaces and programming languages, WebLogic Server provided the platform for developing these new Java-based enterprise applications.

Many in the software industry believe we are at the forefront of another major revolution. Traditionally large enterprises use a variety of different architectures and vital corporate data lives in separate and incompatible systems. Connecting and sharing information between these systems requires expensive custom programming and gives only limited functionality. With the advent of XML and Web services, a new platform for enterprise integration has arrived. Instead of exchanging proprietary binary information over custom interfaces, Web services and XML provide structured data integration over standard, Internet protocols. With BEA WebLogic Server 7.0, you have the tools to embrace and ride this integration revolution.

The software industry continues to innovate at a rapid pace so it is vital to have guides like the *BEA WebLogic Server Bible* to lead you into this new world. This book provides you with the foundation to build these next-generation applications on the WebLogic platform. With the WebLogic Server tools and guidance from this book, I hope you are able to inspire the next technological revolution and bring us all to a new world.

Rob Woollen

Staff Software Engineer

BEA WebLogic Server

# **Preface**

elcome to *BEA WebLogic Serve Bible*, 2nd Edition, a book designed to help you understand and apply the power of BEA WebLogic Server to help solve your organization's most complex distributed computing problems. This book is written for developers who have a decent knowledge of Java and the Java 2 Platform, Enterprise Edition (J2EE), but are either working with WebLogic Server for the first time or further honing their existing WebLogic skills. In other words, this book assumes you have experience with Java and J2EE, but not necessarily with WebLogic.

## Why Do You Need This Book?

Perhaps you're thinking of buying this book because you've just downloaded and installed the trial version of WebLogic Server and have no idea what to do next. Or maybe you've become lost in the lush forest of WebLogic documentation on BEA's Web site and you're looking for a trail map. Or perhaps you've been programming WebLogic since the good old Tengah days and you're looking for additional insights on some of the newer features. Whatever your reasons for picking up this book, if you're looking for broad and fairly deep coverage of WebLogic Server, with a healthy balance between theory and hands-on learning, then this book is for you.

WebLogic Server is a large, complex product whose feature set can be positively overwhelming to newcomers. WebLogic's documentation, although excellent, is equally overwhelming, consisting of over 40 manuals and dozens of tutorials and samples. To the uninitiated, just figuring out how to learn WebLogic is a difficult job. This book is designed to take some of the pain out of that endeavor.

Although WebLogic Server is based on the Java 2 Platform, Enterprise Edition (J2EE), and although this book covers many of J2EE's features, this book will not teach you J2EE programming. In-depth coverage of that topic can be found on Sun's Web site (at http://java.sun.com/j2ee/tutorial/1\_3-fcs/index.html), or in a number of other books, such as the Java 2 Enterprise Edition Bible.

Like all software products, WebLogic Server and the J2EE are constantly changing and evolving. This book is based on WebLogic Server 7.0 and the J2EE SDK version 1.3.



Be sure to visit the WebLogic Bible Web site (www.wiley.com/compbooks/zuffoletto), where you will find the complete source code listings for the book, along with updates, errata, and more!

## **How This Book Is Organized**

Most computer scientists are familiar with the "divide-and-conquer" approach to programming, where programmers divide large programming problems into smaller ones that are

more easily understood and solved. The same approach works well for learning WebLogic Server, and the organization of this book reflects that.

There is no required order for reading the chapters in this book. In fact, it's so comprehensive, you'd have a hard time reading it from cover to cover. Start wherever your immediate interest lies, and then jump to whichever chapters you need for background material. If you do read it from cover to cover, rest assured that the chapters do build upon each other. For example, chapters on writing EJBs come after chapters explaining JDBC, JNDI, and RMI, all technologies relevant to EJBs.

Note

Whichever order you choose, we highly recommend you at least skim through Appendix B before doing any serious coding, building, or deploying. This appendix explains the basics about setting up a command line build environment. It can be very frustrating to get classpath and other build errors; Appendix B steers you through the requirements for getting productive right away. It's well worth the read if for no other reason than all the examples use Ant to build and run them, and Ant basics are explained in this appendix.

# Part I: Preparing Your Enterprise for WebLogic

The book begins by explaining what WebLogic Server is and what its capabilities are. It goes on to explain how to assemble a WebLogic development team and how to design good WebLogic applications. It closes by showing you how to construct an efficient WebLogic development environment on your network. This part of the book is completely non-technical, directed not only at prospective WebLogic developers, but also to managers who are thinking of adopting WebLogic Server and want to understand the impact it will have on their organization.

#### Part II: WebLogic and the J2EE APIs

This section begins the technical portion of the book. Its chapters explain, in depth, how WebLogic implements each of the J2EE APIs, including JDBC (database access), JTA (transaction processing), RMI (distributed applications), JNDI (network and directory lookups), JMS (messaging), and JavaMail (electronic mail processing). The approach of each chapter is to explain the background and architecture of each API, show how to write and deploy a simple package in WebLogic Server using the API, and then move on with more advanced examples. Unlike the WebLogic documentation, which often segregates development and administrative information into different manuals, our approach is to put all this information in one place, for easy reference by the developer.

#### **Part III: Developing Web Components**

Virtually everyone who buys WebLogic Server uses it to write Web components, which include Java servlets, Java Server Pages (JSPs), and tag libraries. This important part of the book shows how to build, deploy, and manage these key components in the WebLogic environment. Servlets were the first successful server-side Java Web technology. They solved problems with older non-Java server-side technologies, such as CGI, because servlets are multithreaded. Servlets often use many of the APIs that were discussed in Part II, such as JDBC for database access and JNDI for looking up resources. We discuss how to create servlets, including how to store information in a session object, and how to implement security via a login process. JSPs build upon servlets by factoring out commonality and separating design from content, thus making them easier to program and maintain. We explain how to set up WebLogic for easy development and debugging of JSPs, how to handle errors, and how

to use JSPs with servlets in an instance of a common design pattern. Tag Libraries expand on JSPs by allowing you to write tags — much like HTML tags — in which you can define and program their specific functionality. These are reusable, so you can keep tag libraries in different applications.

#### Part IV: Developing EJB Components

This part of the book builds on Part II, which covered the J2EE APIs. This section covers EJBs, which allow you to create easy to maintain, scalable, secure, transaction-aware code. It also builds on Part III, since EJBs are often the back end for a system fronted by Web applications. We start off with a broad overview of EJBs, since there are several types and subtypes. EJBs are powerful, and with that power comes significant complexity. Each of the following three chapters covers one of the major types of EJBs: session, entity, and message-driven. Session beans come in two flavors — stateless and stateful — and we discuss the similarities and differences. Then we explain how to code session beans with an example of each type. Even stateful session beans are only meant to store their information for a limited time; they're not considered persistent as are entity EJBs. Entity beans also come in two types: those with container-managed persistence (CMP) and those with bean-managed persistence (BMP). We discuss the pros and cons of choosing CMP or BMP, as well as some of the recent improvements to CMP in the EJB 2.0 specification. We go through an example of each type, including the deployment descriptors that are especially important in CMP. Message-driven EJBs are the newest type of enterprise bean, and we discuss their evolution from JMS. We contrast message-driven beans with using JMS, and then show an example of how to go about creating this kind of EJB.

#### Part V: Deploying and Testing Enterprise Applications

Designing, writing, and building WebLogic applications is important, but what happens when you're ready to go to production? Deploying applications on a local copy of WebLogic is easy, but complexities arise when packaging for deployment on test and production servers. We cover how WARs, JARs, and EARs are packaged, and some of the things to watch out for when packaging them. Classloading is a particularly thorny issue, and we give it a once-over along with an evaluation of different ways of attacking the problem. Once deployed, your application needs to be tested. Does it work? Is it fast enough? We cover the basics of performance tuning, and how to use WebLogic Server features to monitor the performance of your applications. Then we cover tuning WebLogic itself, along with JVM and application tuning. We conclude with an example of how you might go about tuning a real application.

#### **Part VI: Implementing Security**

WebLogic Server 7.0 introduces a completely new and robust security infrastructure that can stand on its own or integrate with outside security architectures. This part of the book explains some of the theory behind securing enterprise computing systems, and then explains WebLogic's new security architecture. It closes by showing how to secure WebLogic resources using a variety of approaches and methods.

### Part VII: WebLogic Server Administration

This part of the book introduces you to WebLogic's administration tools and shows you how to use them to manage WebLogic domains, servers, clusters, and security. WebLogic Server 7.0 introduces new features for clustering and a completely new security architecture, and these chapters will get you up and running right away.

#### Part VIII: Enterprise Application Integration

Web Services and SOAP are two of the newest additions to the enterprise developer's toolkit, and WebLogic Server supports them both. You can code them by hand or use the newly introduced WebLogic Workshop to automate much of the drudgery of designing, developing, testing, and deploying Web services. This part of the book shows you how to work with these emerging technologies in the context of WebLogic Server and Workshop, and it also shows how WebLogic can be used in conjunction with the J2EE Connector Architecture to integrate with existing corporate systems.

#### **Appendixes**

The first appendix explains the major differences between WebLogic 7.0 and earlier versions and gives tips for migrating older WebLogic applications to the newest version. New to this second edition of the book, Appendix B explains many of the bundled tools and how they can make your development and deployment easier and faster. It even includes a jumpstart for Ant, the build tool that will undoubtedly become your best friend.

#### What You'll Need

To gain the most benefit from this book, you'll need a workstation loaded up with the following software:

- ◆ BEA WebLogic Server 7.0 (a free, fully functional trial version is available for download at http://www.bea.com).
- ◆ Sun's J2EE SDK, version 1.3 (a copy of this is included with the WebLogic Server trial version).

Optional items include the following:

- ♦ A relational database, such as Microsoft SQL Server or Oracle. (WebLogic Server ships with a demo version of PointBase relational database, but if your organization uses one of the other two then you should figure out how to work with it right away.)
- ◆ An integrated development environment (IDE) for Java, such as Borland JBuilder (http://www.borland.com).
- An SMTP-compliant mail server, if you plan to write applications that will process incoming or outgoing electronic mail.

#### **Conventions Used in This Book**

As you go through this book, you will find a few unique elements in nearly every chapter, such as source code listings and special icons that draw your attention to important tidbits of information.

#### **Source Code**

Any time you encounter source code, it will be in a special font; this applies to code listings as well as code fragments. This way, it's easy to identify code throughout the book. Whether in a fragment or a numbered listing, code in the text appears in the following style: