The Definitive Guide to SOA

Oracle® Service Bus

SECOND EDITION

Jeff Davies, David Schorow, Samrat Ray, and David Rieber

Apress®

The Definitive Guide to SOA: Oracle Service Bus, Second Edition

ISBN-13 (pbk): 978-1-4302-1057-3

ISBN-13 (electronic): 978-1-4302-1058-0

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Lead Editor: Steve Anglin
Technical Reviewer: Jay Kasi
Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Project Manager: Richard Dal Porto
Copy Editor: Marilyn Smith
Associate Production Director: Kari Brooks-Copony
Production Editor: Laura Esterman
Compositor/Artist: Kinetic Publishing Services, LLC
Proofreader: Nancy Sixsmith
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`.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at `http://www.apress.com`.

Contents at a Glance

Contents

Foreword

The enterprise service bus (ESB) is a hot topic today. Many vendors are either building new products in this category or dressing up their existing products to pitch as ESBs. However, there is no clearly accepted definition of what an ESB is, what its architecture should be, or what its programming paradigm should be. Definitions range from saying that it is nothing and wholly unneeded to saying it is everything and has all the capabilities of a full integration suite with built-in orchestration, data aggregation, and web services management capabilities. Architectures range from being embedded in the clients and endpoints to being a central intermediary to being a decentralized intermediary. Programming paradigms for the ESB range from writing Java to being completely configuration-driven and pliable with graphical interfaces.

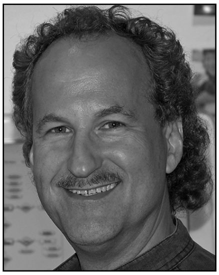
BEA Systems, the original creator of what is now the Oracle Service Bus, did not dress up one of its existing products and pitch it as an ESB. It built an ESB from scratch (first introduced in the summer of 2005), with a razor-sharp focus on where it is positioned as a component in an end-to-end service-oriented architecture (SOA). It complements a business process management or orchestration service, but serves a different and distinct role. Much of SOA is about componentization, interconnectivity, and reuse. The ESB is a component that serves as an intermediary, with the clear and distinct role of providing loose coupling between clients and services, a routing fabric, connectivity, and a central point of security enforcement, contributing to the manageability of your SOA network. It can be a central intermediary or a decentralized network of intermediaries. It is completely configuration-based with browser-based graphical interfaces.

In this book, the authors introduce you to ESBs in general and Oracle Service Bus in particular, with many examples and clear and understandable explanations of the product and its implementation in a number of ESB use cases. This book takes the very practical and useful approach of picking one of the leading products in the ESB category and does a show-and-tell, instead of dwelling on a lot of philosophical discussions and arguments of various contrasting architectures or definitions of an ESB. It is a very readable and instructive book. As one of the architects of the first release of the product, I feel this book is a fine introduction to Oracle Service Bus.

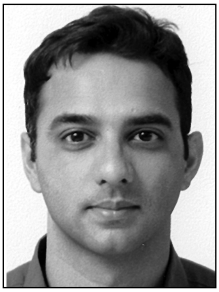
Jay Kasi
Director, Product Management,
Oracle Corporation

About the Authors

JEFF DAVIES has more than 25 years of experience in the software field. This includes developing retail applications, such as Act! for Windows and Macintosh, and a number of other commercially available applications, principally in the telecommunications and medical fields. His background also includes the development, design, and architecture of enterprise applications. Previous to joining BEA, Jeff was Chief Architect at a telecommunications company and ran his own consulting company for a number of years. Now at Oracle, Jeff is focused on the SOA.



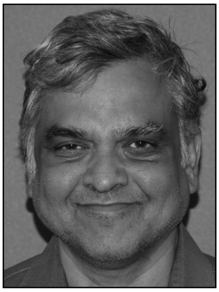
DAVID SCHOROW has more than 20 years experience working on enterprise software. David is currently the Director of Software Development, leading the Oracle Service Bus team. Previously, he was the Chief Architect for BEA AquaLogic Service Bus. He has guided the product's development and evolution from the first release through five (and counting) subsequent releases. Prior to joining BEA, David was the chief Java architect at the NonStop division of Hewlett-Packard, overseeing the development of a wide variety of Java projects, including the NonStop Java Virtual Machine, NonStop SQL JDBC drivers, the port of WebLogic Server to the NonStop platform, and other demanding Java products. David has extensive experience in high-performance, transaction-processing systems—the application environments used by the most demanding customers, such as stock exchanges, airline reservations, health care, and banking.



SAMRAT RAY has more than 10 years of experience in the architecture, design, and implementation of Java/J2EE-based enterprise software. Samrat is a Product Manager at Oracle, where he is responsible for RASP (Reliability, Availability, Scalability, Performance) aspects of multiple products in the SOA Suite. As the Performance Architect for AquaLogic Service Bus at BEA Systems, Samrat has been a key contributor to the architecture and design of the product. He is responsible for multiple innovative features that enable users to build scalable and flexible SOAs using Oracle Service Bus. Samrat has extensive experience in the areas of high-volume transaction processing and high-performance message-oriented systems.

DAVID RIEBER has more than 12 years of experience working on software development. David was a member of the BEA AquaLogic Service Bus team since its inception. As the Security Architect for AquaLogic Service Bus, he designed and implemented its security model and made major contributions to OSB's core runtime. Prior to joining BEA, David was a senior software developer at Sun Microsystems, where he worked on Sun's Java HotSpot Virtual Machine. David has a Master's Degree in Computer Science from Stanford University. David is now a software engineer at Google Inc.

About the Technical Reviewer



JAY KASI has been a software architect for about 20 years. He has worked for Hewlett-Packard as a relational database management system kernel architect, high-availability architect, and distributed OLTP architect. He was the Chief Architect at Commerce One for orchestration and ESB technologies, as well as B2B e-commerce infrastructure. He was the architect for the first few releases of OSB at BEA Systems, and later worked on designing and coordinating the integrations of OSB with other products. He is currently one of the Product Managers at Oracle for the SOA Suite.

Acknowledgments

Many people have helped me to make this book a reality. I want to thank my wife, Roberta, for her love and understanding as I spent hours on my computer, mumbling incoherently about “namespaces” and the like. There is no finer wife in the world. Similarly, I’d like to thank my children, Eric and Madeline, for putting up with my highly distracted nature while writing this book. Of course, I’d like to thank my parents and my aunt and uncle for enabling me to get to this point in my life with their constant love and support.

I’d like to thank Jay Kasi at Oracle for his help and tutelage while writing this book. I have never met a person with such a deep understanding of any software product in my life. Many times when I was stuck on a problem, Jay would quickly look at the code and deliver an exact analysis of the problem within moments.

I’d also like to thank the many folks who helped review the book and to provide me with technical answers to the more unusual scenarios. Specifically, I want to recognize (in alphabetical order) Deb Ayers, Stephen Bennett, Naren Chawla, George Gould, David Groves, Dain Hansen, Gregory Haardt, Karl Hoffman, Ashish Krishna, Usha Kuntamukkala, Saify Lanewala, Denis Pilipchuk, Michael Reiche, Kelly Schwarzhoff, Chris Tomkins, Tolga Urhan, Jeremy Westerman, Mike Wooten, and Bradley Wright.

Finally, I’d like to thank the great mentors in my life, Mark Russell and Gerry Millar. They taught me everything from how to tie a necktie to how to “listen to what they are feeling.” They both taught me that it’s the people that are important; the software is incidental. That’s a hard but invaluable lesson for a natural-born geek.

Jeff Davies

Chapter 17 describes how to extend OSB to communicate with additional applications by writing a custom transport using the Transport SDK. This useful extensibility mechanism was designed and implemented by Greg Fichtenholtz, a Senior Engineer on the OSB team. It is his design that enables OSB to be used in new and different environments not addressed in the original implementation. The usefulness of the Transport SDK is due to his good design work.

Greg is only one member of a very talented team that created the OSB product; however, their names are too numerous to mention (and I’d be afraid of leaving someone out). This group, with its engineering prowess and creative energy, working under the management of Ashok Aletty, who fosters a productive, cooperative, and enjoyable atmosphere, is responsible for making OSB such a fantastic product. I consider myself fortunate to have the opportunity to work with such a great team on this exciting product.

I'd like to thank my sister, Stephanie Schorow, for her thorough review of an early draft of the chapter. She is the real writer of the family. Chapter 17 is much more readable due to her efforts.

Lastly, I'd like to thank my wife, Mona, and my son, Marcus, for their understanding and support when working on this book required my nights and weekends (and a canceled ski trip).

David Schorow

My contributions to this book have been derived from my real-world experiences of developing OSB and enabling customer success with the product. It has been my privilege to be part of a great team that has created and nurtured an outstanding product like OSB.

There are a number of very talented and supportive individuals who directly or indirectly have helped me become an effective contributor to OSB and to this book. Special thanks go to Jay Kasi for being there whenever I have needed his guidance and his wisdom.

I would like to thank Ashok Aletty, David Schorow, and Deb Ayers for their unstinting support and encouragement. I would also like to thank Naren Chawla, Boris Chen, and Dain Hansen for their belief in me during my initial years with OSB. I want to thank Jeff Davies for giving me the opportunity to contribute to this book.

I would like to thank my wife, Devyani, for her support and understanding during all those weekends and nights that I spent on this book. Finally, I would like to thank my parents for being an immense source of inspiration throughout my life.

Samrat Ray

There are many people at BEA I'd like to thank for their support throughout the years. First of all, I'd like to thank David Schorow for being such a great manager and friend. I'd also like to thank Ashok Aletty for keeping OSB on track in the early years and for giving me the opportunity to play a bigger role in OSB security. Thanks to Jay Kasi for his thoughtful insights and to Kelly Schwarzhoff for his invaluable feedback. Special thanks go to Jeff Davies for giving me the opportunity to contribute the security chapter and for his immense patience.

I had the privilege of collaborating with a lot of bright people throughout BEA, beyond the OSB team. Special thanks go to Neil Smithline, John Herendeen, Craig Perez, Juan Andrade, David Garrison, and Denis Pilipchuk

I'd like to thank my wife, Astrid, for her love and support; my kids, Noemi, Joshua, and Raquel, for making me such a proud dad. Finally, I'd like to thank my parents, who taught me that everything is possible.

David Rieber

Introduction

Service-oriented architecture (SOA) is rapidly becoming the new standard for today's enterprises. A number of books discuss various aspects of SOA. Most (if not all) are high level in their discussions, providing some strategies for you to consider but very little tactical information. As software professionals, we are able to grasp these abstract concepts fairly quickly, as we're sure you can. However, the devil is always in the details. We know that once we begin to implement a new technology, we will discover a whole new dimension of bugs, design issues, and other problems that are never discussed in those strategy books.

SOA is not a technology; it is architecture and a strategy. In order for you to implement your own SOA, you will need to learn a new way of thinking about your enterprise and managing software assets. SOA is generally implemented using newer technologies—not a single new technology, but a whole series of different technologies. We thought we knew XML pretty well before we began walking the path to SOA. It didn't take long for us to figure out that there was a lot more to XML than we had previously thought. You can expect to need to learn the details of XML, XML Schema, Web Services Description Language (WSDL), XQuery, and XPath before you can begin to make informed design judgments.

While we enjoy reading about new strategies, we enjoy realizing them in code just as much. Code keeps you honest. A lot of things work very well on paper, but once you start flipping bits, the truth will emerge in all of its intolerant glory. What we really wanted to read was a detailed book on SOA development. Since we could not find one, we wrote one. We wrote this book under the assumption that there were thousands of other software developers like ourselves—people who enjoy writing code and love to put theory into practice.

This book is a mix of theory and working code samples. One of the reasons there are so few books on writing real code for an SOA is because there are so few SOA platforms that the average developer can download and use. Most SOA (and more specifically, enterprise service bus) vendors keep their software locked away, demanding that you purchase it before you can use it. This is like purchasing a car you have never seen or driven based solely on the description provided to you by the salesperson.

Fortunately, Oracle provides an enterprise-class service bus that anyone can download for free, called Oracle Service Bus. This book will walk you through many detailed examples of connecting Oracle Service Bus to legacy systems, show common design patterns for web services, and generally increase both your development and architectural expertise in enterprise service bus (ESB) and SOA.

About the Oracle Service Bus

The Oracle Service Bus (OSB) is a rebranded version of the AquaLogic Service Bus (ALSB) from BEA Systems. Readers of the first version of this book learned about ALSB version 2.6 in depth. ALSB was released by BEA Systems in 2005. In mid-2008, Oracle Corporation acquired BEA Systems. The ALSB product was rebranded to OSB.

The initial release of OSB is version 10.3, in compliance with Oracle naming standards. As we write this, the rebranding of the ALSB product to become OSB is still in progress. As a result, you may see differences in the text and the screenshots of the live product. The screenshots were taken from the "prebranded" version of ALSB. Wherever possible, we have used the new product name in an effort to avoid confusion over the long term.

Oracle had an ESB before the acquisition of BEA. That product is now called Oracle Enterprise Service Bus (OESB). OESB continues to be supported by Oracle.

What's New in OSB 10.3?

In late 2007, version 3.0 of ALSB was released. Now branded as OSB 10.3, it boasts a number of significant enhancements over its earlier versions. OSB incorporates new functionality and enhancements in many areas. Some of the key enhancements include the following:

New development environment: OSB now uses an Eclipse-based IDE called WorkSpace Studio. WorkSpace Studio provides software developers with a more traditional IDE and development life cycle. Oracle has not removed the web-based configuration environment; that is still in place and is a great tool for software quality assurance professionals and operations people. Oracle WorkSpace Studio provides a unified, collaborative design experience across development teams and Oracle products, promoting higher development productivity.

Advanced service pooling and endpoint failover: If a service endpoint is not responding, you can take that service endpoint URI offline automatically and route service requests to alternate service endpoints. When the endpoint URI comes back online, it can be automatically returned to the endpoint pool to handle requests.

Support for Web Services Reliable Messaging (WS-RM): Built on top of the proven WebLogic Server product, OSB 10.3 provides support for the WS-RM standard specified by OASIS.

Business service overload protection (aka throttling): You can limit the amount of throughput to business services to prevent overloading of those services.

Optimized transports for reliability and security propagation: Optimized transports are available when connecting different types of server technologies that are colocated on the same physical machine. These optimized transports are able to reduce the overhead associated with making distributed calls, thereby increasing their performance.

Navigational and metadata sharing in WorkSpace Studio: When working with the Oracle Enterprise Repository (OER) Service Assembly Modeler, you can quickly navigate from a Service Component Architecture (SCA) resource to the design view for an OSB service. OSB also supports the sharing of metadata with OER in WorkSpace Studio. This simplifies the process of ensuring that the metadata from each project is kept in sync with the OER.

Who This Book Is For

This book is for software professionals who are working in an SOA environment, or want to work in an SOA environment. It contains real-world information on SOA best practices, working code samples, and more than 10 years of combined experience from the authors solving real SOA problems.

How This Book Is Structured

This book contains a total of 18 chapters. We've written most of the chapters so that they may be read individually. However, we do recommend reading Chapters 2 and 3, which cover setting up your development environment and understanding the basic principles of an ESB.

Here's a brief summary of what you'll find in this book:

Chapter 1, Why Use a Service Bus?: This chapter describes the functions and benefits of an ESB.

Chapter 2, Installing and Configuring the Software: This chapter guides you through installing and configuring OSB and setting up a development environment. By installing the software as described in this chapter, you will be able to run all of the sample code contained in this book.

Chapter 3, Creating a Hello World Service: In the grand tradition of programming books, we write a web service, test it, and integrate it with OSB. Along the way, you'll get a quick tour of the WorkSpace Studio development environment.

Chapter 4, Message Flow Basics: In this chapter, you will learn how to create message flows and how they are used in OSB.

Chapter 5, A Crash Course in WSDL: WSDL is the language of modern web services. Creating (or just reading) a WSDL file requires a fair bit of skill beyond what is necessary for simple XML. This chapter teaches you the core of what you need to know about WSDL and leaves out the fluff!

Chapter 6, Intermediate Message Flows: In this chapter, we really put OSB through its paces, with sample code for almost every feature available.

Chapter 7, Asynchronous Messaging: In this chapter, you will learn how to loosely couple services with regard to time.

Chapter 8, Service Types and Transports: This chapter walks you through the many different service types and transports supported by OSB, and provides you with information on how to select the correct service type and transport for your needs.

Chapter 9, Advanced Messaging Topics: In this chapter, we cover the advanced messaging capabilities of OSB.

Chapter 10, Reporting and Monitoring: There is more to OSB than just messaging. It can keep you informed about the health of your enterprise, providing automated alerts and sophisticated status reports on both your services and the servers that host them. The chapter describes OSB's reporting and monitoring features.

Chapter 11, SOA Security: This chapter covers a topic that is often discussed but seldom understood. It will provide you with a solid understanding of how to implement security within your service bus.

Chapter 12, Planning Your Service Landscape: The move to SOA requires considerable planning. This chapter introduces a methodology that will simplify this planning process and provide you with a taxonomy by which you can quickly classify your services.

Chapter 13, Implementing Your Service Landscape: In this chapter, we put into action the service landscape methodology introduced in the previous chapter.

Chapter 14, Versioning Services: This is possibly the most controversial chapter in the book! Forget everything you've heard about versioning web services and brace yourself for some heresy!

Chapter 15, Performance: Tuning and Best Practices: This chapter provides tips on how to tune OSB for maximum scalability and performance.

Chapter 16, Administration, Operations, and Management: There is more to a service bus than just development. This chapter covers some best practices for managing your service bus.

Chapter 17, Custom Transports: While OSB provides many useful transport protocols out of the box, it also contains an API that allows you to create your own customer transports so it can integrate with any legacy system. This chapter describes how to create your own custom transport and details the Transport SDK.

Chapter 18, How Do I . . . ?: In this chapter, we answer some common questions about using OSB in the real world.

Downloading the Code

The code presented in this book is available for download in ZIP file format. You can download it from the Downloads section of this book's page on the Apress web site (`http://www.apress.com`).

Contacting the Authors

Readers can contact the authors as follows:

- For questions regarding security, please direct your e-mail to David Rieber at `drieber@gmail.com`.
- For questions on performance tuning, please contact Samrat Ray at `samrat.ray@oracle.com`.
- For questions on using the Transport SDK, please contact David Schorow at `david.schorow@oracle.com`.
- For all other questions, please contact Jeff Davies at `jeff.x.davies@oracle.com`. You can also visit Jeff's web site at `http://jeffdavies.org`.