"Explaining the intersection of these two worlds—service-orientation and .NET technologies—is exactly what this book does. Its team of specialist authors provides a concrete, usable guide to this combination, ranging from the fundamentals of service-orientation to the more rarified air of .NET services in the cloud and beyond. If you're creating serviceoriented software on the Microsoft platform—that is, if you're a serious .NET developer—mastering these ideas is a must." From the Foreword by David Chappell, Chappell & Associates

## SOA with . NE & Windows Azure Realizing Service-Orientation with the Microsoft Platform

Edited and Co-Authored by Thomas Erl, World's Top-Selling SOA Author Forewords by S. Somasegar David Chappell

David Chou, John deVadoss, Nitin Gandhi, Hanu Kommapalati, Brian Loesgen, Christoph Shittko, Herbjorn Wilhelmsen, Mickie Williams



With contributions from Scott Golightly, Daryl Hogan, Jeff King, Scott Seely With additional contributions by members of the Microsoft Windows Azure and AppFabric teams

# SOA with .NET and Windows Azure<sup>™</sup>



# Chapter 1

## Introduction

- **1.1** About this Book
- **1.2** Objectives of this Book
- **1.3** Who this Book is For
- 1.4 What this Book Does Not Cover
- 1.5 Prerequisite Reading
- **1.6** How this Book is Organized
- 1.7 How Principles and Patterns are Used in this Book
- 1.8 Symbols, Figures, and Style Conventions

"It's the transformation of our software, it's the transformation of our strategy and our offerings across the board to fundamentally embrace services."

-Ray Ozzie, Chief Software Architect, Microsoft

#### 1.1 About this Book

Documenting the intersection between the technology capabilities of a vendor productbased platform and the demands and requirements of the service-orientation design paradigm is always an interesting exercise. You find yourself exploring parts of the platform that go well beyond those labeled with the word "service" to discover strengths and weaknesses that have a direct bearing on the potential to realize the very specific goals associated with service-oriented computing.

The body of work provided in this book is the result of a three-year study of Microsoft platform technologies in relation to service-orientation. I was fortunate to work with authors and contributors (most with Microsoft) who collaborated and advanced the study at different intervals, each providing individual expertise and insights.

The use of design principles and patterns in this book proved especially helpful. Because each principle or pattern had been previously well-defined and documented in detail, there was no room for interpretation with its application to or comparison with Microsoft technologies.

One of the greatest revelations of this effort was the variety of options you have when working with Microsoft technologies. I had an initial expectation that we would be documenting very specific combinations of products and technologies for building serviceoriented solutions and service-oriented technology architecture implementations. It turns out that there are many choices, each with its own set of pros and cons. This type of diversity is important, as it can help you create and optimize individual services differently and independently, while still remaining within the same overall technology platform.

As you dive into the many topics covered by this book, I encourage you to keep the big picture in mind. Building services as units of service-oriented logic is about shaping

software programs in a specific way in support of achieving a specific target state. That target state is the big picture. The ultimate goal is for your services to collectively establish an environment that is naturally responsive to on-going business change.

Finally, another important consideration is that the big picture doesn't always have to be "big." The scope in which you choose to apply service-orientation can vary from a modest segment within the IT enterprise to an entire business domain and beyond. As stated in the SOA Manifesto: "Keep efforts manageable and within meaningful bound-aries." The definition of those boundaries is up to you, based on what you can realistically manage.

These factors will further influence how you choose to assemble service-oriented technology environments using Microsoft technologies. Having them in the back of your mind as you study the upcoming chapters will give you a constant context and also some baseline criteria as you consider each technology option in relation to service-orientation and your unique business objectives and requirements.

#### NOTE

The annotated version of the SOA Manifesto is published in Appendix E.

#### 1.2 Objectives of this Book

Collectively, these chapters were written with the following primary goals in mind:

- to provide coverage of contemporary Microsoft distributed technologies and modern service technologies
- to explain how Microsoft cloud computing technologies and platforms can be leveraged by service-oriented technology architectures
- to document the application of service-orientation principles to the Microsoft technology platform
- to explore the application of SOA design patterns to various .NET and Azure technologies and solutions built with these technologies
- to provide coverage of Microsoft infrastructure extensions and administration tools relevant to service-oriented solution deployment and governance

#### 1.3 Who this Book is For

This book can be used as a tutorial and a reference text and is intended for the following types of readers:

- Developers and architects new to .NET and Windows Azure who will supplement this book with additional tutorials to learn how to design and build service-oriented solutions using Microsoft platform technologies.
- Experienced .NET developers and architects who want to learn how to apply service-orientation principles and SOA design patterns in order to create services and service-oriented solutions.
- Developers and architects who want to learn more about Windows Azure and AppFabric.
- Enterprise architects that want to learn more about how to position and establish enterprise service bus and orchestration platforms within the IT enterprise.
- Developers who want to build solutions using modern Microsoft service technologies.

#### 1.4 What this Book Does Not Cover

This is not a "how-to" book for .NET or SOA. Although the six chapters in Part I contain a great deal of introductory coverage of modern .NET and Windows Azure technologies, the overall purpose of this book is to explore the intersection of Microsoft platform technologies and the application of service-orientation principles and SOA design patterns. This book intends to empower you with the knowledge required to properly utilize Microsoft products and technologies for the creation of services, service compositions, service-oriented solutions, and service-oriented technology architectures.

#### 1.5 Prerequisite Reading

This book assumes you have a basic knowledge of:

- the .NET framework
- fundamental XML concepts
- fundamental service-orientation

#### 1.5 Prerequisite Reading

If you have not yet worked with XML, you can read some of the brief tutorials published at www.soaspecs.com. If you are new to SOA, you can get a basic understanding of service-oriented computing, service-orientation, and related design patterns by studying the content at the following Web sites:

- www.whatissoa.com
- www.soaprinciples.com
- www.soapatterns.org
- www.soa-manifesto.com

To further ensure that you have a clear understanding of key terms used and referenced in the upcoming chapters, you can also visit the online master glossary for this book series at www.soaglossary.com to look up definitions for terms that may not be fully described in this book.

Even if you are an experienced SOA practitioner, we suggest you take the time to have a look at these online resources. A great deal of ambiguity has surrounded SOA and service-oriented computing and these explanations and definitions will ensure that you fully understand key terms and concepts in relation to this book and the book series as a whole.

Here are some recommendations for additional books that elaborate on key topics covered by this title:

- *SOA Principles of Service Design* A comprehensive documentation of the serviceorientation design paradigm with full descriptions of all of the principles referenced in this book.
- *SOA Design Patterns* This is the official SOA design patterns catalog containing descriptions and examples for most of the patterns referenced in this book. You can also look up concise descriptions for these patterns at www.soapatterns.org and in Appendix D.
- Web Service Contract Design & Versioning for SOA Any content pertaining to contract-first or WSDL, XML Schema, and WS-Policy development and design, development, and versioning will be aided by the detailed coverage in this title.
- *Service-Oriented Architecture: Concepts, Technology, and Design* The coverage of service-oriented analysis and design processes in this title supplements the technology-centric focus of this book with methodology-related topics.

The following titles are currently in development as part of the Prentice Hall Service-Oriented Computing Series from Thomas Erl:

- *SOA with Java* A book dedicated to building services and service-oriented solutions with Java development tools and technologies, with an emphasis on Web services and REST services.
- *SOA Governance* This book explores a wide range of organizational and technological governance topics, including Web service contract versioning and evolution.
- *SOA with REST* This book documents the convergence of REST and SOA by establishing how REST services can be realized in support of service-orientation. Salient topics are reinforced with comprehensive case studies using modern REST frameworks in combination with contemporary SOA models, patterns, practices, and concepts.
- Modern SOA Infrastructure The aim of this book is to explore modern infrastructure technologies and practices for mainstream service-oriented architectures and solutions. This book provides in-depth coverage of contemporary infrastructure technology components and further provides new design patterns that extend and build upon previously documented SOA design patterns.
- *Cloud Computing & SOA* This book is focused on the convergence of SOA and Cloud Computing. It will provide a comprehensive reference for the technologies and practices that are emerging around the adoption of Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) as they pertain private, public and community clouds in support of service-orientation and service-oriented solution design.

For the latest information regarding the release of these new books, visit www. soabooks.com.

#### 1.6 How this Book is Organized

This book begins with Chapters 1 and 2 providing introductory content and case study background information respectively. All subsequent chapters are grouped into the following parts:

- Part I: Fundamentals
- Part II: Services and Service Composition

#### 1.6 How this Book is Organized

- Part III: Infrastructure and Architecture
- Part IV: Appendices

#### Part I: Fundamentals

The first six chapters cover introductory topics related to SOA, service-orientation, and the broad range of past and present Microsoft distributed technologies.

#### Chapter 3: SOA Fundamentals

This chapter provides an overview of key terms and concepts associated with SOA, service-orientation, and service-oriented computing.

#### Chapter 4: A Brief History of Legacy .NET Distributed Technologies

This chapter begins with distributed computing basics, and then proceeds to summarize the evolution of Microsoft distributed technologies, including COM, DCOM, COM+ Services, .NET Enterprise Services, .NET Remoting, MSMQ, System.Transactions, and XML Web Services (ASMX), including Web Services Enhancements (WSE).

#### Chapter 5: WCF Services

This chapter introduces the Windows Communication Foundation (WCF) platform, with an emphasis on service technologies and implementation and hosting options. Areas of focus include service contracts and service consumer design requirements. WCF services are used in examples throughout subsequent chapters.

#### Chapter 6: WCF Extensions

The exploration of various architectural extensions to WCF provided in this chapter acts as a continuation of Chapter 5. Extensions covered include security, transactions, routing, discovery, management tools, and extensibility options.

#### Chapter 7: .NET Enterprise Services Technologies

Further architectural and infrastructure building blocks are provided by .NET Enterprise Service Technologies. This chapter provides introductory coverage of SQL Server, Windows Workflow Foundation (WF), Windows Server AppFabric, Application Blocks, and Software Factories.

#### Chapter 8: Cloud Services with Windows Azure

Following a section that covers basic cloud computing concepts and terminology, this chapter delves into the Windows Azure Platform to cover specific topics, such as Azure roles and storage options, as well as tutorial-style coverage of how Web services and REST services can be created for deployment within Windows Azure.

#### Part II: Services and Service Composition

When working with service-orientation, a service is a software program with specific characteristics. Many of these characteristics foster the ability of a service to be repeatedly aggregated into different service compositions. This part of the book contains a series of chapters that explore how service-orientation principles and patterns can be applied to build units of service-oriented solution logic (services) in support of strategic service-oriented computing goals, with a special emphasis on support for future composability requirements.

#### Chapter 9: Service-Orientation with .NET Part I: Service Contracts and Interoperability

The design and standardization of service contracts is a focal point when building service-oriented solutions. This chapter provides numerous examples and coverage of .NET technologies shaped by the application of the relevant design principles, patterns, and practices (including the Standardized Service Contract principle and the Canonical Schema, Data Model Transformation, Canonical Protocol, and Canonical Expression patterns).

#### Chapter 10: Service-Orientation with .NET Part II: Coupling, Abstraction, and Discoverability

This chapter explores the application of numerous patterns and principles in relation to service coupling, abstraction, and discoverability requirements and concerns. Patterns and principles covered include Service Loose Coupling, Decoupled Contract, Service Façade, Concurrent Contracts, Service Abstraction, Validation Abstraction, Service Discoverability, and Metadata Centralization.

#### Chapter 11: Service-Orientation with .NET Part III: Reusability and Agnostic Service Models

With an emphasis on fostering the reusability potential of services, this chapter explores the application of the Service Reusability principle from modeling, design, and development perspectives via the application of fundamental service identification and definition patterns, including Functional Decomposition, Service Encapsulation, Agnostic

#### 1.6 How this Book is Organized

Context, Agnostic Capability, Utility Abstraction, and Entity Abstraction. The latter sections in the chapter contain detailed case study examples for the development of utility and entity services using Web service and REST service technologies, respectively.

### Chapter 12: Service-Orientation with .NET Part IV: Service Composition and Orchestration Basics

This chapter covers fundamental service composition theory and discusses the importance of the Service Composability principle in relation to the Capability Composition, Capability Recomposition, Service Layers, Non-Agnostic Context, and Process Abstraction patterns. The latter half of the chapter acts as a prelude to Chapters 13 and 14 by providing introductory coverage of patterns relevant to Microsoft orchestration platforms, including Process Centralization, State Repository, Compensating Transaction, and the Orchestration compound pattern itself.

#### Chapter 13: Orchestration Patterns with WF

Using the Orchestration compound pattern (and its core member patterns) as a basis, this chapter explores relevant technologies and features of the Windows Workflow Foundation (WF) platform.

#### Chapter 14: Orchestration Patterns with BizTalk Server

Similar in structure to Chapter 13, this chapter covers technologies and features of the BizTalk Server product in relation to the Orchestration compound patterns and the core patterns that comprise it.

#### Part III: Infrastructure and Architecture

The chapters in this part of the book cover specialized topics and extensions that pertain to layers of technology architecture and infrastructure relevant to the development and governance of service-oriented solutions and inventories of services.

#### Chapter 15: Enterprise Service Bus with BizTalk Server and Windows Azure

The primary focus of this chapter is on how the BizTalk Server product is expanded via the ESB Toolkit and how .NET based enterprise service bus implementations can be further extended into the realm of cloud computing via Windows Azure. (Note that this chapter does not specifically explore the Microsoft technology platform from the Enterprise Service Bus compound pattern perspective, as that is covered separately in the *Modern SOA Infrastructure* book as part of this book series.)

#### Chapter 16: Windows Azure Platform AppFabric Service Bus

This chapter is dedicated to providing an overview of the Windows Azure Platform AppFabric Service Bus and contains various examples that explore its usage, especially with REST-based services.

#### Chapter 17: SOA Security with .NET and Windows Azure

Security is a primary concern when building services and service-oriented solutions because if the security of an agnostic service is breached, it can impact several service compositions. This chapter covers basic security patterns (such as Direct Authentication and Brokered Authentication) in relation to .NET technologies and further provides an overview of the Windows Identity Foundation platform. The chapter concludes with a section dedicated to security concerns raised by Windows Azure.

#### Chapter 18: Service-Oriented Presentation Layers with .NET

There are various ways to abstract, design, and develop presentation logic in support of service-orientation. This chapter describes the Windows Presentation Foundation and Prism Library and explores its usage from a patterns perspective.

#### Chapter 19: Service Performance Optimization

One of the common myths of service-oriented computing is that performance must always be sacrificed when building reusable and composable services. This chapter provides a number of techniques that demonstrate how, using the many modern technology and infrastructure advances, service performance and reliability can be optimized while continuing to support the application of service-orientation principles.

#### Chapter 20: SOA Metrics with BAM

This final chapter explains the business activity monitoring features provided by BizTalk Server, and how they can be applied specifically for the collection of metrics relevant to service usage, scalability, and general service governance.

#### Part IV: Appendices

#### Appendix A: Case Study Conclusion

This appendix provides a conclusion of the case study storyline, as it pertains to Standard Mold IT enterprise environment.

#### 1.7 How Principles and Patterns are Used in this Book

#### Appendix B: Industry Standards Reference

A list of referenced industry standards is provided in this appendix.

#### Appendix C: Service-Orientation Principles Reference

This appendix provides the profile tables (originally from *SOA Principles of Service Design*) for the service-orientation design principles referenced in this book.

#### Appendix D: SOA Design Patterns Reference

This appendix provides the profile tables (originally from *SOA Design Patterns*) for the SOA design patterns referenced in this book.

#### Appendix E: The Annotated SOA Manifesto

This appendix provides the annotated version of the SOA Manifesto declaration.

#### Appendix F: Additional Resources

A list of relevant Web sites and an article for the *SOA Magazine* are provided as supplementary resources.

#### 1.7 How Principles and Patterns are Used in this Book

As part of its exploration of service-orientation in relation to the Microsoft technology platform, this book references and uses established design principles and patterns throughout its chapters.

#### Sources

The principles of service-orientation were originally documented in the book *SOA Principles of Service Design*. Referenced design patterns originated in the following design patterns publications:

- [CJP] "Core J2EE Patterns" (Sun Microsystems, java.sun.com, 2001-2002)
- [DP] "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Ralph Johnson, Richard Helm, John Vlissides (Addison-Wesley, 1995, ISBN: 0201633612)
- [JG] "Tales from the Smart Client" by John Gossman (blogs.msdn.com, 2005)

- [MPP] Microsoft Patterns & Practices Group
- [PEA] "Patterns of Enterprise Application Architecture" by Martin Fowler (Pearson Education, 2003, ISBN: 0-321-12742-0)
- [SDP] "SOA Design Patterns" by Thomas Erl (Prentice Hall, 2009, ISBN: 978-0-13-613516-6)
- [TAL] "MVP: Model-View-Presenter The Taligent Programming Model for C++ and Java" by Mike Potel (Taligent, 1996)
- [TR] Trygve M. H. Reenskaug

#### **Reference Notation**

Note how each of the listed publications is associated with a code in square brackets. This code is repeated every time a pattern name is mentioned within body text or title text (with the exception of chapter and top-level section titles).

Profile tables for all 85 of the original patterns from the *SOA Design Patterns* book are provided in Appendix D. Therefore, instead of a code, the actual page number of the corresponding profile table is provided each time a pattern from this publication is referenced.

Similarly, because the profile tables of the eight service-orientation principles are also provided in Appendix C, referenced design principle names are further supplemented with the corresponding page number. However, in order to maintain a distinction between principles and patterns, the page number for each principle is placed in rounded parentheses instead of square brackets.

For example, the following statement first references a service-orientation design principle, then an SOA design pattern, and finally a pattern from another publication:

"...the Service Loose Coupling (695) principle is supported via the application of Decoupled Contract [735], which is comparable to the Separated Interface [PEA] pattern..."

Note, as also demonstrated in this sample statement, a principle or a pattern can be referenced with or without being qualified. In other words, the statement "...when the Decoupled Contract [735] pattern is applied..." has the same meaning as "...when Decoupled Contract [735] is applied..."

#### 1.8 Symbols, Figures, and Style Conventions

#### Symbol Legend

This book contains a series of diagrams that are referred to as *figures*. The primary symbols used throughout all figures are individually described in the symbol legend located on the inside of the front cover.

#### How Color is Used

The color red is occasionally used to highlight text, especially within code samples. Generally, the highlighted code will be related to the current topic being discussed.

#### **Additional Information**

The following sections provide supplementary information and resources for the *Prentice Hall Service-Oriented Computing Series from Thomas Erl.* 

#### Updates, Errata, and Resources (www.soabooks.com)

Information about other series titles and various supporting resources can be found at www.soabooks.com. You are encouraged to visit this site regularly to check for content changes and corrections.

#### Master Glossary (www.soaglossary.com)

To avoid content overlap and to ensure constant content currency, the books in this series do not contain glossaries. Instead, a dedicated Web site at www.soaglossary.com provides a master glossary for all series titles. This site continues to grow and expand with new glossary definitions as new series titles are developed and released.

#### Referenced Specifications (www.soaspecs.com)

The chapters throughout this book reference XML and Web services specifications and standards. The www.soaspecs.com Web site provides a central portal to the original specification documents created and maintained by the primary standards organizations.

#### SOASchool.com<sup>™</sup> SOA Certified Professional (SOACP)

This text book is an official part of the SOA Certified Professional curriculum and is used in conjunction with courses and exams for the SOA .NET Developer Certification program. The course materials that are part of this program provide additional content and lab exercises that further explore topics covered in this book. For more information, see www.soaschool.com.

#### The SOA Magazine (www.soamag.com)

The *SOA Magazine* is a regular publication provided by SOA Systems Inc. and Prentice Hall/PearsonPTR and is officially associated with the *Prentice Hall Service-Oriented Computing Series from Thomas Erl*. The *SOA Magazine* is dedicated to publishing specialized SOA articles, case studies, and papers by industry experts and professionals. The common criteria for contributions is that each explore a distinct aspect of service-oriented computing.

#### **Notification Service**

If you'd like to be automatically notified of new book releases in this series, new supplementary content for this title, or key changes to the previously listed Web sites, use the notification form at www.soabooks.com.