Introduction

We software architects and developers live in a fascinating time. With the release of the .NET Framework in 2000, Web services technology has swept into our programming toolset and into our collective consciousness. Web services are the killer application for XML. Web services are the "new way" to call distributed objects remotely. Web services will take all of our integration headaches away and allow formerly incompatible systems to communicate again. What Microsoft developer has not recently thought to himself, "should I be building my application with Web services?"

What .NET developer has not recently thought to himself, "I'm confused"?

Every tidal wave has a genesis, and a momentum, and a final destination where it typically crashes head-on into a stable landmass and causes havoc and confusion. Web services technology is a tidal wave.

The genesis is Microsoft's strategic decision to simplify SOAP-based Web services development using a seamless set of integrated classes in the .NET Framework. The momentum is provided by a relentless marketing machine that promotes Web services as the solution for many of our worst IT problems. One destination is us, the architects and the developers who must understand this technology and learn how to implement it. Another destination is the manager, who must make strategic decisions on how to put this technology to its best use.

The Web services technology tidal wave has created confusion for .NET developers because, quite simply, we do not know the best way to use it. We are wrapped up in misconceptions about what the technology is for, and this affects our judgment in using it properly. We will spend the first chapter clarifying these misconceptions, but let me reveal one:

> *Misconception*: Web services are for making remote procedure calls to distributed objects.
>
> *Reality*: Web services are not optimized for RPCs. This is not what they are best at. Web services work best when they respond to messages, not to instructions.

Until now, we could safely give developers time to absorb the new Web services technology. We needed time to play around with the .NET Framework and to get used to a new development approach. Web services development using the .NET Framework is stunning in its simplicity. It is equally stunning in its oversimplification of a deep and sophisticated technology. Play time is over; now it's time we grow up.

Web services play a key role in a greater whole known as service-oriented architecture (SOA). Quite simply, SOA is an architecture based on loosely coupled components that exchange messages. These components include the clients that make message-based service requests, and the distributed components that respond to them. In an SOA, Web services are critically important because they consume and deliver messages.

It is difficult to tackle topics such as SOA and Web services without invoking the ire of developers working on other platforms such as J2EE and IBM WebSphere. We have full respect for these platforms and for the efforts of the developers and the architects who use them. These guys and girls "get it," and they have been doing it for longer than we Microsoft-oriented developers have. Let's give credit where credit is due, but then move on. Because if you are reading this book, it is a safe assumption that you are interested in SOA the Microsoft way. If this describes you, then please buy this book and read on!

So why don't we Microsoft/.NET developers "get it"? It is not for lack of intelligence, nor is it for lack of an ability to understand sophisticated architectures. We don't get it because we have been misled as to why Web services are important. Let us roughly restate our original assertion:

> Web services work best with messages. They are not optimized to handle specific instructions (in the form of direct, remote procedure calls).

Most of us have been "trained" to this point to use Web services for implementing SOAP-based remote procedure calls. This is where we have been misled, because SOAP is about the worst protocol you could use for this purpose. It is verbose to the point where the response and request envelopes will likely exceed in size the actual input parameters and output response parameters that you are exchanging!

At this point, we hope we have left you with more questions than answers. We have stated things here that you can only take our word on, but why should you believe us?

This is exactly what we are trying to get at. We want to shake you out of your Web services comfort zone and to help you rethink the technology and think of the bigger picture that is SOA. We devote the first part of this book to clearing up the misconceptions. And we devote the second part of this book to showing you how to implement Web services in an SOA.

Free your mind.

Who This Book Is For

This book is a practical reference written for intermediate to advanced .NET solution developers and architects who are interested in SOA and Web services development. The book focuses on two key areas:

- How to build message-oriented and service-oriented Web services
- Understanding WSE 3.0

Solution developers and architects alike will find a lot in this book to hold their interest. The material in the book provides detailed conceptual discussions on SOA combined with in-depth C# code samples. The book avoids rehashing familiar concepts and focuses instead on how to rethink your approach to Web services development using today's best tools and industry-standard specifications. The book was written using the production version of WSE 3.0 that was released shortly following Visual Studio 2005, so you have the benefit of the latest and greatest developments with both Visual Studio and WSE.

What This Book Covers

This book covers SOA and cutting-edge Web services development using the WS- specifications and WSE 3.0. The first half of the book shows you how to think in terms of messages rather than procedure calls. It shows you how to design and build message- and service-oriented Web services that provide the security and the functionality that companies and businesses will require before they are ready to widely adopt Web services technology.

The second half of the book focuses on WSE 3.0, which provides infrastructure and developer support for implementing industry-standard Web service specifications, including

WS-Security: Integrates a set of popular security technologies, including digital signing and encryption based on security tokens, including X.509 certificates.

WS-Policy: Allows Web services to document their requirements, preferences, and capabilities for a range of factors, though is mostly focused on security. For example, a Web service policy will include its security requirements, such as encryption and digital signing based on an X.509 certificate.

WS-Addressing: Identifies service endpoints in a message and allows for these endpoints to remain updated as the message is passed along through two or more services. It largely replaces the earlier WS-Routing specification.

WS-Messaging: Provides support for alternate transport channel protocols besides HTTP, including TCP. It simplifies the development of messaging applications, including asynchronous applications that communicate using SOAP over HTTP.

WS-Secure Conversation: Establishes session-oriented trusted communication sessions using security tokens.

The WS- specifications are constantly evolving as new specifications get submitted and existing specifications get refined. They address essential requirements for service-oriented applications. This book aims to get you up to speed with understanding the current WS- specifications and how the WSE 3.0 product works and where Web services technology is headed for the next few years.

If you are interested in taking your Web services development to the next level, you will find this book to be an invaluable reference.

Chapter Summary

This book is broken into nine chapters, progressing from introductory conceptual information to advanced discussions of the WS- specifications and their implementation in WSE 3.0. You will get the most out of this book if you read at least the first five chapters in sequence. These chapters contain reference information and conceptual discussions that are essential to understanding the material in the second half of the book. The remaining chapters of the book cover all of the WS- specifications that are implemented by WSE 3.0. Finally, the book closes with a chapter on the Windows Communication Foundation (WCF), which is the name for a managed communications infrastructure for building service-oriented applications. The purpose of the WCF chapter is to show you the direction that service-oriented application development is headed, and to show you how your work with WSE 3.0 will help you make the transition to WCF very smooth.

The summary of the chapters is as follows:

Chapter 1, Introducing Service-Oriented Architecture: This chapter introduces the concepts behind SOA and the characteristics of a Web service from the perspective of SOA. This chapter reviews the following topics:

- SOA concepts and application architecture.
- The WS-I Basic Profile.
- The WS- specifications.
- WSE 3.0 (an introduction).

Chapter 2, The Web Services Description Language: This chapter reviews the WSDL 1.1 specification and the elements of a WSDL document. This information is essential to understanding what makes up a service. The concepts that are presented here will come up repeatedly throughout the book, so make sure you read this chapter! This chapter includes the following:

- The seven elements of the WSDL document (types, message, operation, portType, binding, port, and service), which together document abstract definitions and concrete implementation details for the Web service.
- How to work with WSDL documents using Visual Studio .NET.
- How to use WSDL documents.

Chapter 3, Design Patterns for Building Message-Oriented Web Services: This chapter shows you how to build message-oriented Web services, as opposed to RPC-style Web services, which most people end up building with ASP.NET even if they do not realize it. The goal of this chapter is to help you rethink your approach to Web services design so that you can start developing the type of message-oriented Web services that fit into an SOA framework. This chapter covers the following:

- Definition of a message-oriented Web service.
- The role of XML and XSD schemas in constructing messages.
- How to build an XSD schema file using the Visual Studio .NET XML Designer.
- Detailed review of a six-step process for building and consuming a message-oriented Web service. This discussion ties into the sample solutions that accompany the chapter.

Chapter 4, Design Patterns for Building Service-Oriented Web Services: This chapter extends the discussion from Chapter 3 and shows you how to build Web services that operate within a service-oriented application. This chapter includes the following:

- A discussion on building separate type definition assemblies that are based on XSD schema files.
- How to build a business assembly for delegating service processing.

- Detailed review of a six-step process for building and consuming a service-oriented Web service. This discussion ties into the sample solutions that accompany the chapter.
- How to build a service agent, which is unique to SOA.

Chapter 5, Web Services Enhancements 3.0: This chapter provides a detailed overview of WSE 3.0. This chapter covers the following:

- Overview of the WS- specifications.
- Introduction to WSE 3.0—what it contains, what it does, how it integrates with ASP.NET, and how to install it.
- Overview of X.509 certificates—the WSE sample digital certificates are used frequently throughout the sample applications. Certificate installation can be difficult, so this section shows you what you need to do.

Chapter 6, Secure Web Services with WS-Security: This is the first of three chapters that provide detailed discussions on the WSE implementations of the WS- specifications. *Security* typically refers to two things: authentication and authorization. This chapter contains the following:

- Overview of the WS-Security specification and implementation, including the enhanced declarative model in WSE 3.0.
- Review of common security scenarios, including an overview on important security objects and concepts such as security tokens, digital signatures, and encryption.
- How to implement WS-Security using WSE 3.0 and the username-ForCertificateSecurity turnkey security assertion.
- Review of declarative vs. imperative authorization.

Chapter 7, Extended Web Services Security with WS-Security and WS-Secure Conversation: This chapter reviews how WSE 3.0 can secure other common Web service deployment scenarios. This chapter covers the following:

- Overview of the direct and brokered authentication models.
- How to implement brokered authentication using Kerberos and mutual certificates.
- How to prevent reply attacks, using time stamps, digital signatures, and message correlation.
- Overview of the WS-Secure Conversation specification, which is enhanced in WSE 3.0.
- How to implement a secure conversation between a Web service and its client, using a security token service provider.

Chapter 8, SOAP Messages: Addressing, Messaging, and Routing: This chapter covers several WS- specifications that work together to provide a new messaging framework for Web services. Traditional Web services are built on the HTTP request/response model. WSE 3.0 provides a messaging framework that expands the supported transport protocols to include TCP and an optimized in-process transport protocol, in addition to HTTP. These protocols are not natively tied to a request/response communications model, so you can implement alternative models, such as asynchronous messaging solutions. This chapter also reviews the WS-Addressing specification, which enables messages to store their own addressing and endpoint reference information. This chapter includes the following:

- Overview of communication models for Web services.
- Overview of the WS-Addressing specification, including a discussion of message information headers vs. endpoint references.
- Overview of how WSE implements the WS-Addressing specification.
- Overview of the WS-Messaging specification and the WSE implementation, which provides support for alternate message transport protocols and communication models.
- How to implement a TCP-based Web service using SOAP sender and receiver components.
- Overview of the WS-Routing and WS-Referral specifications, which allow messages to be redirected between multiple endpoints.
- How to build a SOAP-based router using WSE, WS-Routing, and WS-Referral.
- How to integrate MSMQ with Web services in order to implement one form of reliable messaging.

Chapter 9, Beyond WSE 3.0: Looking Ahead to Windows Communication Foundation (WCF): WCF (formerly code named *Indigo*) provides infrastructure and programming support for service-oriented applications. WCF will be released in late 2006 as part of the upcoming Vista operating system. It focuses on messages, providing support for creating messages, for delivering messages, and for processing messages. With WCF there is less ambiguity in your services: the infrastructure forces you to be message oriented and to work with well-qualified XML-based data types. WSE 3.0 and its future revisions will provide you with excellent preparation for working with WCF in the future. This chapter contains the following:

- Overview of WCF architecture, including the Indigo service layer, the WCF connector, hosting environments, messaging services, and system services.
- Understanding WCF Web services.
- Understanding WCF applications and infrastructure.
- How to get ready for WCF.
- WSE 3.0 and WCF.

Notes on the Second Edition

This book is the second edition release of *Expert Service-Oriented Architecture: Using the Web Services Enhancements 2.0*. Readers of the previous edition will find that about 60 percent of the material has been rewritten to cover breaking changes and new features in WSE 3.0. The five introductory chapters of this book are similar to the first edition, although all code samples and screen captures have been updated to reflect WSE 3.0 and Visual Studio 2005.

The most significant change in WSE 3.0 is in the area of security implementation, with the introduction of the *turnkey security scenarios*, which are natively supported, common security scenarios that can be implemented using straightforward policy declaration files. Policy files were important in WSE 2.0, but in WSE 3.0 they assume an even greater importance, to the point that in most cases you will not need to write custom code with the WSE 3.0 API. Correspondingly, the second edition of this book reduces the amount of .NET code compared to what was presented in the first edition, and instead focuses more on how to achieve your goals using declarative policy files. The exception is in the area of SOAP messaging, which allows you to build custom SOAP senders and receivers that operate over alternate protocols instead of HTTP. This area is still code-intensive compared to other functional areas that are supported by WSE 3.0.

It is important to note that the WSE 3.0 product is not a full upgrade to WSE 2.0; rather it is a complementary product that improves on certain areas (such as security implementation) while leaving other areas essentially untouched (such as SOAP messaging). The full WSE 2.0 functionality has been subsumed into the WSE 3.0 product, so you will not need to use both products. However, what this means is that you can leverage many aspects of your WSE 2.0 experience into WSE 3.0, which will prevent productivity disruption and will allow you more time to focus on important enhancements in WSE 3.0.

If you have already purchased the first edition of this book you will still find a lot of value in this second edition, particularly in Chapters 6 and 7 on security implementations, which are significantly enhanced in WSE 3.0. These chapters have been completely rewritten for this edition. If you are new to this book you will find it to be a comprehensive resource for building service-oriented Web services using the WSE 3.0 product.

Code Samples and Updates

This book is accompanied by a rich and varied set of example solutions. The sample solutions were built using the production version of WSE 3.0 that was released on November 7, 2005. The code examples are chosen to illustrate complicated concepts clearly. Although Web Services Enhancements are conceptually complicated, this does not mean that they translate into complex code. In fact, the situation is quite the opposite. You will be surprised at how clear and straightforward the code examples are, plus you will find that most WSE-supported functionality can be accessed and administered via declarative policy files that do not require you to write a single line of .NET code.

Note The sample solutions are available for download at `http://www.apress.com`.

Visit http://www.bluestonepartners.com/soa.aspx for updates to the book and sample solutions, and for errata corrections. Check there often, because WSE is expected to undergo several revisions between now and the release of the WCF. In addition, the topic of SOA continues to evolve rapidly, and every month brings new, interesting developments.

And now, once more into the breach, dear friends, once more . . .