Preface

WINDOWS COMMUNICATION FOUNDATION (WCF) is the unified programming model for writing distributed applications on the Microsoft platform. It subsumes the prior technologies of ASMX, .NET Remoting, DCOM, and MSMQ and provides an extensible API to meet a wide variety of distributed computing requirements. Prior to WCF, you needed to master each of those technologies to select the right approach for a particular distributed application requirement. WCF simplifies this considerably by providing a unified approach.

XML Web Services is the most common technique for distributed computing in modern applications. They're used to expose technical and business functions on private or public networks. Sometimes they use the SOAP specification, sometimes they don't. They typically transmit information as text documents containing angle brackets, but not always. They generally use HTTP for the transport, but again, not always. WCF is a framework for working with XML Web Services and is compatible with most technology stacks.

Rich, Chris, and I have each developed code with .NET from the beginning (circa 1999). We work at Microsoft in the field, helping customers use WCF to solve real-world problems. Our customers range from large multinational corporations to ISVs to Web startups. Each has different challenges, needs, and priorities that we individually address. We show them what's possible, recommend what works well, and steer clear of what doesn't. We have experience building distributed applications and leverage that experience in teaching others about WCF.

Our goal for this book is to present WCF in a way that can immediately be put to use by software developers. We cover the material in enough detail that you know how and why to use different features. We go a bit further in most cases, describing some of the subtleties in the framework, but not so far as to document the API.

The Blogosphere is rich with WCF details. Much of it comes from the .NET product team and much of it comes from other developers learning it along the way. We made extensive use of blogs as source material. This book brings order to that repository by organizing it in a way that can be easily consumed from your desk, sofa, or wherever you do your best reading.

Who Should Read This Book?

We wrote this book for software developers who want to build distributed applications on the .NET platform. As fellow developers, we know the importance of solid advice and clear examples on how to use new technology. We've trolled the Blogosphere, scoured internal Microsoft e-mail aliases, and wrote plenty of code to provide you with the best examples for doing the things you need to do.

Architects who need to understand WCF will also benefit from this book. The chapters covering basics, bindings, channels, behaviors, hosting, workflow, and security describe important aspects of designing and implementing services with WCF. Reading the two- to three-page introductions in each of these chapters may be the best way to get the 50,000-foot view of the technology.

Our goal in writing this book is to shorten your learning curve for WCF. We describe and demonstrate how to do the common tasks, addressing the basics as well as advanced topics. Throughout the book, we approach topics as a series of problems to be solved. Rather than documenting the API, we describe how to use WCF to accomplish your goals.

Prerequisites for this book are modest. If you're interested in WCF, you probably already have grounding in .NET. You're probably competent in C# or Visual Basic, or at least you were at one point. And, of course, you probably know your way around Visual Studio. So we're assuming that

you can write decent .NET code and are motivated to make the best use of your time in becoming proficient in WCF.

Installation Requirements

WCF is a key component of the Microsoft .NET Framework 3.x. WCF was first released with .NET 3.0 and has been enhanced in .NET 3.5. The delta between the two releases is modest: enhancements for non-SOAP Web services, integration between WCF and WF, and a healthy service pack. This book covers .NET 3.5. Unless there's a reason to use an older release, this is the clear recommendation.

.NET is packaged in two forms: the redistributable runtime libraries and the software development kit (SDK). The runtime libraries are meant for target machines—those machines that are not for development. This includes testing, staging, and production environments. The SDK is meant for your development machines. The SDK contains code samples, documentation, and tools that are useful for development. Each of these .NET packages, the runtime and SDK versions, can be downloaded from Microsoft's MSDN site at http://msdn2.microsoft.com/en-us/netframework/default.aspx. The .NET 3.5 SDK also ships with Visual Studio 2008.

The Microsoft .NET Framework 3.5 can be installed on Windows XP SP2, Windows Vista, Windows Server 2003, and Windows Server 2008.

Organization

We don't expect you to read the book cover to cover. If you're new to WCF, you may want to read and try the samples in Chapter 1, "Basics," first. Following that, each subsequent chapter covers a major feature set of WCF. We include a few introductory pages in each chapter to describe the motivation and some design goals, and then we cover subtopics within the chapter.

Chapter 1, "Basics," is where we cover the basics of building and consuming WCF services. We discuss and demonstrate how to implement different types of interfaces and why you may choose each. By the end of this chapter, you'll be able to produce and consume services using WCF.

Chapter 2, "Contracts," covers the three primary types of contacts in WCF: service contracts, data contracts, and message contracts. Each of these enables you to define complex structures and interfaces in code. Data contracts map .NET types to XML, service contracts expose service interface endpoints in WSDL that can be consumed in a cross-platform manner, and message contracts enable developers to work directly on the XML in a message, rather than working with .NET types. For each of these contracts, WCF tools generate and export standards-based WSDL to the outside world.

Chapter 3, "Channels," covers channels and channel stacks. The channel model architecture is the foundation on which the WCF communication framework is built. The channel architecture allows for the sending and receiving of messages between clients and services. Channel stacks can be built to exactly match your needs.

Chapter 4, "Bindings," describes how to configure the communication stack to use exactly the protocols you need. For instance, if you're communicating within an enterprise and won't be crossing firewalls, and you need the fastest performance, a binding named `netTcpBinding` will give you best results. If you're looking to communicate with every last Web client out there, then HTTP and text encoded XML is necessary, so `basicHttpBinding` is the way to go. A binding is synonymous with a preconfigured channel stack.

Chapter 5, "Behaviors," covers service behaviors. In WCF, behaviors are the mechanism for affecting service operation outside of the actual message processing. Everything that is done after a message is received but before it is sent to the service operation code is the domain of behaviors. In WCF, this is where concurrency and instance management is handled, as well as transactional support. This chapter also demonstrates how to build custom behaviors for additional service control.

Chapter 6, "Serialization and Encoding," describes the process by which data is serialized from a .NET Type (class) to an XML Infoset and the way that XML Infoset is represented on the wire. We typically think of XML as a text document with angle brackets around field names and values, but the XML Infoset is a more basic data structure. This chapter discusses ways of converting that structure into a format that can be exchanged over a network.

Chapter 7, "Hosting," describes the various options in hosting a WCF service. The most common environment, IIS, is described, but it is by far **not** the only option. WCF services can be hosted in Managed .NET applications, Windows Activation Services, or any other .NET program. This chapter discusses the options and techniques for hosting.

Chapter 8, "Security," is a large chapter and covers the multitude of security options. Different authentication schemes are discussed and demonstrated. Transport- and message-level security are compared, with examples of each. Intranet and Internet scenarios are also described.

Chapter 9, "Diagnostics," describes how to use the built-in trace facilities in .NET to capture WCF events. Trace Listeners are described, along with examples that show how to configure the settings for different events. The Trace Viewer, a powerful tool that is shipped with WCF, is also described, which enables you to trace activities across service call boundaries.

Chapter 10, "Exception Handling," offers practical guidance on handling exceptions within WCF. SOAP faults are described using fault contracts, and examples demonstrate how to throw and catch them to minimize errors.

Chapter 11, "Workflow Services," covers the integration points between WCF and Windows Workflow Foundation (WF) introduced in Visual Studio 2008 and .NET 3.5. We describe how to call WCF services from WF and how to expose WF workflows in WCF.

Chapter 12, "Peer Networking," shows how to build client-to-client applications that leverage a network mesh to enable clients to find each other. We cover mesh addressing and techniques for establishing point-to-point connections after the client addressing is resolved.

Chapter 13, "Programmable Web," covers how to use WCF for non-SOAP Web Services. Examples are shown with Asynchronous JavaScript and XML (AJAX) and JSON for simpler, JavaScript-friendly data formats. The hosting classes specific to non-SOAP protocols are described. Like WCF-WF integration, this is new with .NET 3.5.

Finally, the appendix, "Advanced Topics," covers advanced topics that we didn't fit into other chapters. Rather than burying them somewhere they don't belong, we include them separately.

Because of the broad nature of the WCF subject, not all topics are covered in equal depth. This book's goal is to help developers be super productive when working with WCF. If we do our job, readers will use this book as they learn the technology. This book does not attempt to document WCF—that's what the good tech writers at Microsoft have done with the help files and MSDN. But a combination of that documentation and the good guidance found in these pages should enable developers to quickly and productively build robust applications with WCF.

NOTE Code Continuation Arrows

When a line of code is too long to fit on one line of text, we have wrapped it to the next line. When this happens, the continuation is preceded with a code-continuation arrow (➥).