

Advanced .NET Remoting, Second Edition

INGO RAMMER AND MARIO SZPUSZTA

Apress®

Advanced .NET Remoting, Second Edition

Copyright © 2005 by Ingo Rammer and Mario Szpuszta

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN (pbk): 1-59059-417-7

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

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editor: Ewan Buckingham

Technical Reviewer: Kent Sharkey

Editorial Board: Steve Anglin, Dan Appleman, Ewan Buckingham, Gary Cornell, Tony Davis, Jason Gilmore, Jonathan Hassell, Chris Mills, Dominic Shakeshaft, Jim Sumser

Project Manager: Laura E. Brown

Copy Manager: Nicole LeClerc

Copy Editor: Ami Knox

Production Manager: Kari Brooks-Copony

Production Editor: Ellie Fountain

Compositor: Kinetic Publishing Services, LLC

Proofreader: Elizabeth Berry

Indexer: John Collin

Artist: Kinetic Publishing Services, LLC

Cover Designer: Kurt Krames

Manufacturing Manager: Tom Debolski

Distributed to the book trade in the United States by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013, and outside the United States by Springer-Verlag GmbH & Co. KG, Tiergartenstr. 17, 69112 Heidelberg, Germany.

In the United States: phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders@springer-ny.com, or visit <http://www.springer-ny.com>. Outside the United States: fax +49 6221 345229, e-mail orders@springer.de, or visit <http://www.springer.de>.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. 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> in the Downloads section.

*To Katja,
Who was courageous enough to marry me
even though she knew I would write another book.
—Ingo*

*To my parents—I am so happy that I have you!
And to my best friends Dominik and Edi—I enjoy every single moment with you!
—Mario*

Contents at a Glance

About the Authors	xvii
About the Technical Reviewer	xix
Acknowledgments	xxi
Introduction	xxiii

PART 1 ■ ■ ■ Understanding

CHAPTER 1	Introduction to Remoting	3
CHAPTER 2	.NET Remoting Basics	9
CHAPTER 3	.NET Remoting in Action	25
CHAPTER 4	Configuration and Deployment	75
CHAPTER 5	Securing .NET Remoting	123
CHAPTER 6	Creating Remoting Clients	161
CHAPTER 7	In-Depth .NET Remoting	185
CHAPTER 8	The Ins and Outs of Versioning	225
CHAPTER 9	.NET Remoting Tips and Best Practices	275
CHAPTER 10	Troubleshooting .NET Remoting	303

PART 2 ■ ■ ■ Extending

CHAPTER 11	Inside the Framework	321
CHAPTER 12	Creation of Sinks	349
CHAPTER 13	Extending .NET Remoting	359
CHAPTER 14	Developing a Transport Channel	421
CHAPTER 15	Context Matters	469

PART 3 ■ ■ ■ Reference

APPENDIX A	.NET Remoting Usage Reference	487
APPENDIX B	.NET Remoting Extensibility Reference	525
APPENDIX C	.NET Remoting Links	541
INDEX	549

Contents

About the Authors	xvii
About the Technical Reviewer	xix
Acknowledgments	xxi
Introduction	xxiii

PART 1 ■ ■ ■ Understanding

■ CHAPTER 1	Introduction to Remoting	3
	What Is Remoting?	3
	Scenarios for .NET Remoting	3
	Centralized Business Logic	4
	Physical Separation of Layers	4
	Accessing Other Platforms	4
	Evolution of Remoting	4
	DCE/RPC	5
	CORBA	5
	DCOM	5
	MTS/COM+	6
	Java RMI	6
	Java EJB	6
	Web Services/SOAP/XML-RPC	7
	.NET Remoting	7
	Summary	7
■ CHAPTER 2	.NET Remoting Basics	9
	Advantages of .NET Remoting	9
	Ease of Implementation	9
	Extensible Architecture	10
	Interface Definitions	11
	Serialization of Data	12
	Lifetime Management	12
	Multiserver/Multiclient	13

	Your First Remoting Application	13
	The First Sample	14
	Extending the Sample	20
	Summary	23
CHAPTER 3	.NET Remoting in Action	25
	Types of Remoting	25
	ByValue Objects	25
	MarshalByRefObjects	26
	Types of Invocation	46
	Synchronous Calls	47
	Asynchronous Calls	51
	One-Way Calls	55
	Multiserver Configuration	59
	Examining a Sample Multiserver Application	60
	Sharing Assemblies	67
	Shared Implementation	67
	Shared Interfaces	67
	Shared Base Classes	67
	SoapSuds-Generated Metadata	68
	Summary	74
CHAPTER 4	Configuration and Deployment	75
	Configuration Files	76
	Watch for the Metadata!	77
	The Problem with SoapSuds	77
	Porting the Sample to Use Configuration Files	82
	Standard Configuration Options	85
	What About Interfaces?	100
	Using the IPC Channel in .NET Remoting 2.0	102
	Deployment	108
	Console Applications	108
	Windows Services	108
	Deployment Using IIS	116
	Summary	121

CHAPTER 5	Securing .NET Remoting	123
	Building Secure Systems	123
	Authentication Protocols in Windows	124
	NTLM Authentication	124
	Kerberos: Very Fast Track	126
	Security Package Negotiate	128
	Security Support Provider Interface	128
	Identities and Principals: A Short Overview	129
	Securing with IIS	133
	Authentication with IIS	133
	Encryption and IIS	138
	Security Outside of IIS	140
	Using the MSDN Security Samples	140
	Implementing Authorization in the Server	149
	Security with Remoting in .NET 2.0 (Beta)	151
	Summary	160
CHAPTER 6	Creating Remoting Clients	161
	Creating a Server for Your Clients	161
	Creating a Console Client	163
	Creating Windows Forms Clients	167
	Creating Back-End–Based Clients	169
	ASP.NET-Based Clients	169
	Remoting Components Hosted in IIS As Clients	172
	Security Considerations	177
	Summary	184
CHAPTER 7	In-Depth .NET Remoting	185
	Managing an Object's Lifetime	185
	Understanding Leases	186
	Working with Sponsors	196
	Using the CallContext	209
	Best Practices	212
	Security and the Call Context	213
	Remoting Events	213
	Events: First Take	214
	Refactoring the Event Handling	217
	Why [OneWay] Events Are a Bad Idea	222
	Summary	224

CHAPTER 8	The Ins and Outs of Versioning	225
	.NET Framework Versioning Basics	225
	A Short Introduction to Strong Naming	225
	Versioning in .NET Remoting—Fundamentals	233
	Versioning of Server-Activated Objects	233
	Versioning of Client-Activated Objects	240
	Versioning of [Serializable] Objects	242
	Advanced Versioning Concepts	246
	Versioning with Interfaces	246
	Versioning Concepts for Serialized Types	256
	Summary	273
CHAPTER 9	.NET Remoting Tips and Best Practices	275
	.NET Remoting Use Cases	275
	Cross-AppDomain Remoting	276
	Cross-Process on a Single Machine	276
	Cross-Process on Multiple Machines in a LAN	276
	Cross-Process via WAN/Internet	278
	Nonusage Scenarios	279
	The Nine Rules of Scalable Remoting	280
	Using Events and Sponsors	281
	How to Notify Nevertheless	282
	Message Queuing to the Rescue	283
	Other Approaches	286
	SoapSuds vs. Interfaces in .NET Remoting	286
	Custom Exceptions	288
	Scaling Out Remoting Solutions	290
	Load Balancing Basics	290
	Taking Nodes Online/Offline	299
	Designing Applications for Static Scalability	299
	Summary	301
CHAPTER 10	Troubleshooting .NET Remoting	303
	Debugging Hints	303
	Manual Breakpoints	304
	Configuration File Settings	305
	Local or Remote?	307
	Checking Types on Your Server	308

BinaryFormatter Version Incompatibility	309
Troubleshooting with a Custom Sink	310
Changing Security Restrictions with TypeFilterLevel	311
Using Custom Exceptions	313
Multihomed Machines and Firewalls	315
Client-Activated Objects Behind Firewalls	317
Summary	318

PART 2 ■ ■ ■ Extending

■ CHAPTER 11 Inside the Framework	321
Looking at the Five Elements of Remoting	321
A Bit About Proxies	322
Understanding the Role of Messages	326
Examining Message Sinks	328
Serialization Through Formatters	329
Moving Messages Through Transport Channels	330
Client-Side Messaging	331
ClientContextTerminatorSink and Dynamic Sinks	332
SoapClientFormatterSink	333
HttpClientChannel	333
Server-Side Messaging	333
HttpServerChannel and HttpServerTransportSink	335
SDLChannelSink	335
SoapServerFormatterSink and BinaryServerFormatterSink	336
DispatchChannelSink	336
CrossContextChannel	336
ServerContextTerminatorSink	337
LeaseSink	337
ServerObjectTerminatorSink and StackbuilderSink	337
All About Asynchronous Messaging	338
Asynchronous IMessageSink Processing	338
Asynchronous IClientChannelSink Processing	340
Generating the Request	342
Handling the Response	345
Server-Side Asynchronous Processing	347
Summary	348

CHAPTER 12	Creation of Sinks	349
	Understanding Sink Providers	349
	Creating Client-Side Sinks	350
	Creating Server-Side Sinks	354
	Using Dynamic Sinks	356
	Summary	357
CHAPTER 13	Extending .NET Remoting	359
	Creating a Compression Sink	359
	Implementing the Client-Side Sink	361
	Implementing the Server-Side Sink	364
	Creating the Sink Providers	367
	Using the Sinks	369
	Extending the Compression Sink	371
	Encrypting the Transfer	375
	Essential Symmetric Encryption	376
	Creating the Sinks	380
	Creating the Providers	386
	Passing Runtime Information	390
	Changing the Programming Model	402
	Using This Sink	408
	Avoiding the BinaryFormatter Version Mismatch	409
	Using a Custom Proxy	413
	Some Final Words of Caution	419
	Summary	419
CHAPTER 14	Developing a Transport Channel	421
	Protocol Considerations	421
	The Shortcut Route to SMTP	422
	... And Round-Trip to POP3	423
	Character Encoding Essentials	424
	Creating E-Mail Headers	425
	Encapsulating the Protocols	426
	Checking for New Mail	433
	Registering a POP3 Server	435
	Connecting to .NET Remoting	437
	Implementing the Client Channel	445
	Creating the Client's Sink and Provider	449

Implementing the Server Channel	453
Creating the Server's Sink	458
Wrapping the Channel	462
Using the SmtChannel	465
Preparing Your Machine	467
Some Final Words of Caution	468
Summary	468

CHAPTER 15 Context Matters	469
Working at the MetaData Level	471
Creating a Context	472
Checking Parameters in an IMessageSink	480
Summary	483
Conclusion	484

PART 3 ■ ■ ■ Reference

APPENDIX A .NET Remoting Usage Reference	487
System Types	487
System.Activator Class	488
System.MarshalByRefObject Class	488
System.SerializableAttribute Class	489
System.Delegate Class	490
System.IAsyncResult Interface	491
System.Runtime.Remoting	491
Basic Infrastructure Classes	491
Configuration Classes	493
Exception Classes	497
General Interfaces	498
System.Runtime.Remoting.Channels	499
General Interfaces and Classes	499
System.Runtime.Remoting.Channels.Http	504
HttpChannel Class	504
HttpClientChannel Class	505
HttpServerChannel Class	506
System.Runtime.Remoting.Channels.Tcp	506
TcpChannel Class	506
TcpClientChannel Class	507
TcpServerChannel Class	508

System.Runtime.Remoting.Lifetime	508
ILease Interface	508
ISponsor Interface	509
ClientSponsor Class	510
LifetimeServices Class	511
LeaseState Enumeration	511
System.Runtime.Remoting.Messaging	512
AsyncResult Class	512
CallContext Class	512
LogicalCallContext Class	514
OneWayAttribute Class	514
System.Runtime.Remoting.Metadata	514
SoapAttribute Class	515
SoapTypeAttribute Class	515
SoapFieldAttribute Class	515
SoapMethodAttribute Class	516
SoapParameterAttribute Class	516
SoapOption Enumeration	516
System.Runtime.Remoting.Services	516
EnterpriseServicesHelper Class	516
RemotingClientProxy Class	517
ITrackingHandler Interface	517
TrackingServices Class	517
System.Runtime.Serialization	518
ISerializable Interface	519
SerializationInfo Class	520
StreamingContext Structure	520
SerializationException Class	521
System.Runtime.Serialization.Formatter	521
SoapFault Class	521
SoapMessage Class	521
TypeFilterLevel Enumeration	521
Summary	523
APPENDIX B .NET Remoting Extensibility Reference	525
System.Runtime.Remoting.Messaging	525
IMessage Interface	525
IMessageSink Interface	526
IMethodMessage Interface	527
IMethodCallMessage Interface	528

IMethodReturnMessage Interface	528
MethodCall Class	529
MethodResponse Class	529
System.Runtime.Remoting.Activation	529
IConstructionCallMessage Interface	530
IConstructionReturnMessage Interface	530
System.Runtime.Remoting.Proxies	530
RealProxy Class	531
ProxyAttribute Class	531
System.Runtime.Remoting.Channels	531
IChannelSinkBase Interface	532
IClientChannelSink Interface	532
IClientChannelSinkProvider Interface	533
IClientFormatterSink Interface	534
IClientFormatterSinkProvider Interface	534
IServerChannelSink Interface	534
IServerChannelSinkProvider Interface	535
ITransportHeaders Interface	536
IChannel Interface	537
IChannelReceiver Interface	538
IChannelSender Interface	539
BaseChannelObjectWithProperties Class	539
BaseChannelWithProperties Class	540
BaseChannelSinkWithProperties Class	540
Summary	540

■ APPENDIX C .NET Remoting Links 541

Ingo's .NET Remoting FAQ Corner	541
MSDN and MSDN Magazine Articles	541
"Improving Remoting Performance"	541
".NET Remoting Security"	541
"Boundaries: Processes and Application Domains"	542
".NET Remoting Architectural Assessment"	542
".NET Remoting Overview"	542
"Building Secure ASP.NET Applications: Authentication, Authorization, and Secure Communication"	542
"NET Remoting Authentication and Authorization Sample"	542
"Managed Extensions for C++ and .NET Remoting Tutorial"	543
".NET Remoting Use-Cases and Best Practices" and "ASP.NET Web Services or .NET Remoting: How to Choose"	543

“Remoting Examples”	543
“Secure Your .NET Remoting Traffic by Writing an Asymmetric Encryption Channel”	543
“Create a Custom Marshaling Implementation Using .NET Remoting and COM Interop”	543
.NET Remoting Interoperability	544
.NET Remoting: CORBA Interoperability	544
.NET Remoting: Java RMI Bridges	544
XML-RPC with .NET Remoting	544
Custom .NET Remoting Channels	544
Named Pipes Channel for .NET Remoting	545
TcpEx Channel for .NET Remoting	545
Jabber Channel	545
Remoting Channel Framework Extension	545
“Using MSMQ for Custom Remoting Channel”	545
“Using WSE-DIME for Remoting over Internet”	546
Interesting Technical Articles	546
C# Corner: Remoting Section	546
“Share the Clipboard Using .NET Remoting”	546
“Chaining Channels in .NET Remoting”	546
“Applying Observer Pattern in .NET Remoting”	546
“Aspect-Oriented Programming Enables Better Code Encapsulation and Reuse” and “.NET Remoting Spied On”	547
“Persistent Events in Stateless Remoting Server”	547
“Intrinsyc’s Ja.NET—Extending the Reach of .NET Remoting”	547
“Implementing Object Pooling with .NET Remoting—Part I”	547
“.NET Remoting Versus Web Services”	547
“.NET Remoting Central”	547
“Output Caching for .NET Remoting”	548
“Abstract Client Formatter Sink”	548
Remoting Tools	548
Remoting Management Console	548
Remoting Probe	548
INDEX	549

About the Authors

■ **INGO RAMMER** is cofounder of thinktecture, a company supporting software architects and developers with architecture and design of .NET and Web Services applications. He is a regular speaker about these topics at conferences around the world, author of numerous online and print articles, and winner of the *.NET Developer's Journal's* Readers' Choice Award for Best .NET Book of 2003. You can reach him at <http://www.thinktecture.com/staff/ingo>.

■ **MARIO SZPUSZTA** is working in the Developer and Platform Group of Microsoft Austria. Before he started working for Microsoft, Mario was involved in several projects based on COM+ and DCOM with Visual Basic and Visual C++ as well as projects based on Java and J2SE. With Beta 2 of the .NET Framework, he started developing Web applications with ASP.NET. Right now, as developer evangelist for Microsoft Austria, he is doing workshops, trainings, and proof-of-concept projects together with independent software vendors in Austria based on .NET, Web Services, and Office 2003 technologies.

About the Technical Reviewer

■ **KENT SHARKEY** is currently the content strategist for ASP.NET and Visual Studio content for MSDN. When not answering e-mail, he occasionally writes, codes, and sleeps. He lives in green, green Redmond with his wife, and two furry “children,” Squirrel and Cica.

Acknowledgments

First and foremost, I want to thank Mario for taking the challenge to write the second edition of this book together with me. Without him, this book could not exist.

I especially want to thank those people whom I've met in the previous years and whose insight constantly influenced the way I think about software. A big “thank you” therefore needs to go to Beat Schwegler, Clemens Vasters, Don Box, and Steve Swartz.

But the biggest “thank you” needs to go to the thousands of software developers and architects who contacted me by e-mail, who chatted with me at various conferences, and who wrote thought-provoking Weblog and newsgroup posts. You are the reason why I wrote this book.

Of course, writing a book would not be possible without the support of many people. Even though only Mario's and my name appear on the cover, this book would not have been possible without my fellows at thinkecture: Christian Weyer, Christian Nagel, and Ralf Westphal; my friend and the technical reviewer for this book, Kent Sharkey; and the fine editorial staff at Apress: Laura Brown, Ami Knox, and Ellie Fountain. Thank you for making this book a reality!

—Ingo Rammer

The first large project I started working on was one of the most interesting projects I have ever been part of. The enthusiasm and creativity of the two masterminds in this team, Harald Leitenmüller and Benedikt Redl, have inspired me. These two persons have shown me what software development in large projects really means, and they have shown me how interesting software architecture is! Without them, I would not have progressed even half as far as I have today. Therefore, my biggest thanks go to Harald and Benedikt.

Without Beat Schwegler, I would not have had the chance to get involved in writing this book. Thank you, Beat, for this great opportunity and much more for the things I have learned in the past two years from you. These two years have been a really great time!

Last but not least, I want to thank you, Ingo, for giving me the chance to write this book with you. It was really great. I learned many things, and right now I recognize that writing concepts and thoughts down is something that can be really funny and, even more so, interesting.

—Mario Szpuszta

Introduction

In the time since the first edition of this book has been published, quite a bit has changed in the world of software development on Microsoft's platforms. The .NET Framework has become a tried-and-true development platform, and service orientation gained a larger-than-expected momentum. The latter especially presents a very challenging task for the developer using .NET Remoting; the need to avoid possible incompatibilities with future paradigms. If service orientation will, in the next few years, gain the success it deserves, it might be important for your application to be developed in a way to easily adopt these new ideas.

In this book, I have therefore followed a slightly different approach from the one I did in the previous edition. While the first book focused only on covering all the features of the .NET Remoting framework, Mario and I tried to extend this second edition with the best practices for using this technology. While it still covers nearly each and every feature of the .NET Remoting framework, the largest part of the new chapters of this book—especially Chapters 5, 8, 9, and 10—deals with security, best practices, and the general avoidance of problems.

What Is Covered in This Book

This book covers the means of cross-process and cross-machine interaction of applications developed with the .NET Framework. It will provide you with an in-depth understanding of the remoting capabilities that are built into the .NET Framework.

.NET Remoting is different from most other means of remote object access because it can be as easy as writing COM components in Visual Basic 6, yet also gives you the option to extend remoting to include virtually any protocol on any transportation layer you will come across.

Part 1 of the book gives you a thorough introduction to .NET Remoting basics and how you can use .NET Remoting “out of the box.” This gives you a whole range of possibilities, from fast binary transfer protocol to a cross-platform SOAP protocol, with the potential to switch between both without changing a single line in your code. At the end of this part, you will be able to design and develop remoteable components and know just what you have to do to achieve your goals. This part also deals with objects' lifetimes, security, versioning, marshalling, and deployment.

Part 2 covers the advanced features of .NET Remoting and its extensibility model. At the end of the second part, you will have an in-depth understanding of the inner workings of remoting and will know how to extend the framework to meet your requirements. You should not be afraid, especially as you go through the sample code in the second part of the book, to either hit F1 or to insert a breakpoint and examine the Locals window in your custom channel sink to see the exact contents of the objects that get passed as parameters to your methods.

What This Book Doesn't Cover

This book is in no way a rehash of the supplied documentation, but is meant to be used in conjunction with it. You will only find a small percentage of the information that is covered in the online documentation in this book and vice versa, so it is very important for you to use the .NET Framework SDK documentation as well.

I chose this approach to writing a book for one simple reason: I assume that, as an advanced developer, you don't have much time to waste going through a 1,000-page book of which 600 pages are a reproduction of the online documentation. Instead, you want to read the information that has not been covered before. If you think so as well, this book is right for you.

Who This Book Is For

This book is for the intermediate-to-advanced programmer who wants a hands-on guide to .NET Remoting. Although this book is not an introduction to .NET, the CLR, or any .NET language, you nevertheless will be able to use the knowledge and insight you'll get from this book with any of these programming languages. All the samples printed in this book are written in Visual Basic .NET, but you can download each and every sample in both *C#* and Visual Basic .NET.

If you are a “use-it” developer, Part 1 (Chapters 1 through 10) of this book will serve you well by providing a general introduction to the possibilities of remoting and giving you in-depth information on how to use the capabilities that come with .NET Remoting “out of the box.” This part also includes guidance on security, best practices, and troubleshooting.

If you are more of an “understand-it-and-extend-it” developer, Part 2 of this book is for you. Chapters 11 through 15 were written for those who want to understand what's going on behind the scenes of .NET Remoting and how the framework can be customized using proxies, messages, channel sinks, and providers. It also demonstrates how a complete transport channel is implemented from scratch.

At the end of the book, you'll find a collection of appendixes that provide a reference of the namespaces, classes, and interfaces that comprise the .NET Remoting framework.

How This Book Is Structured

Advanced .NET Remoting is divided into two parts. Part 1 (Chapters 1 through 10) covers everything you need to know for developing distributed applications within the .NET Framework. Part 2 (Chapters 11 through 15) gives you a thorough technical insight that will allow you to really understand what's happening behind the scenes and how you can tap into customizing the framework to suit your exact needs. Following is a brief chapter-by-chapter summary of the topics covered in this book.

Chapter 1: Introduction to Remoting

This chapter gives you a short introduction to the world of distributed application development and the respective technologies. It presents some scenarios in which .NET Remoting can be employed and includes historical background on the progress and development of various remoting frameworks during the last ten years.

Chapter 2: .NET Remoting Basics

This chapter gets you started with your first remoting application. Before going directly into the code, I present the distinctions between .NET Remoting and other distributed application frameworks. I then introduce you to the basic types of remote objects, which are server-activated objects and client-activated objects, and show you how to pass data by value. I also give you some basic information about lifetime management issues and the generation of metadata, which is needed for the client to know about the interfaces of the server-side objects.

Chapter 3: .NET Remoting in Action

In this chapter, I demonstrate the key techniques you'll need to know to use .NET Remoting in your real-world applications. I show you the differences between Singleton and SingleCall objects and untangle the mysteries of client-activated objects. I also introduce you to SoapSuds, which can be used to generate proxy objects containing only methods' stubs.

Chapter 4: Configuration and Deployment

This chapter introduces you to the aspects of configuration and deployment of .NET Remoting applications. It shows you how to use configuration files to avoid the hard coding of URLs or channel information for your remote object. You also learn about hosting your server-side components in Windows Services and IIS.

Chapter 5: Securing .NET Remoting

This chapter shows you how to leverage IIS's features when it comes to hosting your components in a secured environment. In this chapter, you learn how to enable basic HTTP sign-on and the more secure Windows-integrated authentication scheme, which is based on a challenge/response protocol. You also see how to enable encrypted access by using standard SSL certificates at the server side.

You will also read about ways to use .NET Remoting in a secure way when not relying on IIS.

Chapter 6: Creating Remoting Clients

Whenever I explain a new feature of the .NET Remoting framework, I tend to present it in an easily digestible console application to avoid having to show you numerous lines of boilerplate .NET code.

Of course, most of your real-world applications will either be Windows Forms or ASP.NET Web applications or Web Services. In this chapter, you therefore learn how to create remoting clients either as desktop or Web applications.

Chapter 7: In-Depth .NET Remoting

As a developer of distributed applications using .NET Remoting, you have to consider several fundamental differences from other remoting techniques and, of course, from the development of local applications. These differences, including lifetime management, versioning, and the handling of asynchronous calls and events, are covered in this chapter.

Chapter 8: The Ins and Outs of Versioning

Here you learn how to create .NET Remoting applications that are version resilient in a way that allows you to support different versions of clients with the same server.

Chapter 9: .NET Remoting Tips and Best Practices

In this chapter, I introduce you to a number of best practices that I've learned in more than three years of using .NET Remoting in numerous projects. This chapter will help you to increase scalability, performance, and stability of your distributed applications.

Chapter 10: Troubleshooting .NET Remoting

Unfortunately, things can and will go wrong at some point in time. That's why this chapter gives you a number of techniques and tools that help you to troubleshoot various issues you might encounter when using .NET Remoting. But don't be afraid: most of these can be remedied in a very brief amount of time.

Chapter 11: Inside the Framework

.NET provides an unprecedented extensibility for the remoting framework. The layered architecture of the .NET Remoting framework can be customized by either completely replacing the existing functionality of a given tier or chaining new implementation with the baseline .NET features.

Before working on the framework and its extensibility, I really encourage you to get a thorough understanding of the existing layers and their inner workings in this architecture. This chapter gives you that information.

Chapter 12: Creation of Sinks

This chapter covers the instantiation of message and channel sinks and sink chains. It shows you the foundation on which to build your own sinks—something you need to know before tackling the implementation of custom sinks.

Chapter 13: Extending .NET Remoting

This chapter builds on the information from Chapters 7 and 8 and shows you how to implement custom remoting sinks. This includes channel sinks that compress or encrypt the transported information, and message sinks to pass additional runtime information from a client to the server or to change the .NET Remoting programming model. This chapter concludes with showing you how to implement custom remoting proxies that forward method calls to remote objects.

Chapter 14: Developing a Transport Channel

This chapter builds on the information you gained in Chapters 7, 8, and 9 and presents the development of a custom .NET Remoting channel that transfers messages via standard Internet e-mail by using SMTP and POP3. It shows not only the implementation of this channel, but also the necessary phase of analyzing the underlying protocol to combine it with the features and requirements of .NET Remoting.

Chapter 15: Context Matters

This last chapter is about message-based processing in local applications. Here you learn how you can intercept calls to objects to route them through `IMessageSinks`. This routing allows you to create and maintain parts of your application's business logic at the metadata level by using custom attributes. You also discover why it might or might not be a good idea to do so.

Appendix A: .NET Remoting Usage Reference

This first appendix includes reference information you'll need when using .NET Remoting in your application. You'll learn about all the namespaces involved when creating clients and servers, and configuring and troubleshooting your application.

Appendix B: .NET Remoting Extensibility Reference

This second appendix covers the namespaces, classes, and interfaces that allow you to extend the .NET Remoting framework.

Appendix C: .NET Remoting Links

At the end of this book are collected a number of links to additional .NET Remoting-specific content on the Web. This includes everything from Microsoft-provided additional articles to custom channels and remoting extensions.

Source Code Download

You can find all source code presented in this book at the Apress download page at <http://www.apress.com>. If you have further suggestions or comments or want to access even more sample code on .NET Remoting, you are invited to visit thinktecture's .NET Remoting FAQ, which is hosted at <http://www.thinktecture.com/Resources/RemotingFAQ>.

We hope that you will benefit from the techniques and information we provide in this book when building your distributed applications based on the .NET Framework.

Ingo Rammer and Mario Szpuszta
Vienna, Austria