

Join the discussion @ p2p.wrox.com



Wrox Programmer to Programmer™



Professional C# 4 and .NET 4

Christian Nagel, Bill Evjen, Jay Glynn, Karli Watson, Morgan Skinner

Related Wrox Books

Beginning ASP.NET 4: in C# and VB

ISBN: 9780470502211

This introductory book offers helpful examples and step-by-step format and has code examples written in both C# and Visual Basic. With this book you will gradually build a Web site example that takes you through the processes of building basic ASP.NET Web pages, adding features with pre-built server controls, designing consistent pages, displaying data, and more.

Beginning Microsoft Visual Basic 2010

ISBN: 9780470502228

This book not only shows you how to write Windows applications, Web applications with ASP.NET, and Windows mobile and embedded CE apps with Visual Basic 2010, but you'll also get a thorough grounding in the basic nuts-and-bolts of writing good code. You'll be exposed to the very latest VB tools and techniques with coverage of both the Visual Studio 2010 and .NET 4 releases.

Beginning Microsoft Visual C# 2010

ISBN: 9780470502266

Using this book, You will first cover the fundamentals such as variables, flow control, and object-oriented programming and gradually build your skills for Web and Windows programming, Windows forms, and data access. Step-by-step directions walk you through processes and invite you to "Try it Out," at every stage. By the end, you'll be able to write useful programming code following the steps you've learned in this thorough, practical book. If you've always wanted to master Visual C# programming, this book is the perfect one-stop resource.

Professional ASP.NET 4: in C# and VB

ISBN: 9780470502204

Written by three highly recognized and regarded ASP.NET experts, this book provides all-encompassing coverage on ASP.NET 4 and offers a unique approach of featuring examples in both C# and VB, as is the incomparable coverage of core ASP.NET. After a fast-paced refresher on essentials such as server controls, the book delves into expert coverage of all the latest capabilities of ASP.NET 4. You'll learn site navigation, personalization, membership, role management, security, and more.

Professional Visual Basic 2010 and .NET 4

ISBN: 9780470502242

If you've already covered the basics and want to dive deep into VB and .NET topics that professional programmers use most, this is your guide. You'll explore all the new features of Visual Basic 2010 as well as all the essential functions that you need, including .NET features such as LINQ to SQL, LINQ to XML, WCF, and more. Plus, you'll examine exception handling and debugging, Visual Studio features, and ASP.NET web programming.

Visual Basic 2010 Programmer's Reference

ISBN: 9780470499832

Visual Basic 2010's Programmer's Reference is a language tutorial and a reference guide to the 2010 release of Visual Basic. The tutorial provides basic material suitable for beginners but also includes in-depth content for more advanced developers.

Professional Visual Studio 2010

ISBN: 9780470548653

Written by an author team of veteran programmers and developers, this book gets you quickly up to speed on what you can expect from Visual Studio 2010. Packed with helpful examples, this comprehensive guide examines the features of Visual Studio 2010, and walks you through every facet of the Integrated Development Environment (IDE), from common tasks and functions to its powerful tools.

WPF Programmer's Reference: Windows Presentation Foundation with C# 2010 and .NET 4

ISBN: 9780470477229

Written by a leading expert on Microsoft graphics programming, this richly illustrated book provides an introduction to WPF development and explains fundamental WPF concepts.

PROFESSIONAL C# 4 AND .NET 4

| | |
|--|-----|
| INTRODUCTION..... | LI |
| ► PART I: THE C# LANGUAGE | |
| CHAPTER 1: .NET Architecture..... | 3 |
| CHAPTER 2: Core C# | 23 |
| CHAPTER 3: Objects and Types..... | 65 |
| CHAPTER 4: Inheritance..... | 89 |
| CHAPTER 5: Generics..... | 107 |
| CHAPTER 6: Arrays and Tuples | 129 |
| CHAPTER 7: Operators and Casts..... | 151 |
| CHAPTER 8: Delegates, Lambdas, and Events | 183 |
| CHAPTER 9: Strings and Regular Expressions | 207 |
| CHAPTER 10: Collections | 225 |
| CHAPTER 11: Language Integrated Query..... | 267 |
| CHAPTER 12: Dynamic Language Extensions..... | 295 |
| CHAPTER 13: Memory Management and Pointers..... | 307 |
| CHAPTER 14: Reflection..... | 333 |
| CHAPTER 15: Errors and Exceptions..... | 351 |
| ► PART II: VISUAL STUDIO | |
| CHAPTER 16: Visual Studio 2010 | 373 |
| CHAPTER 17: Deployment..... | 407 |
| ► PART III: FOUNDATION | |
| CHAPTER 18: Assemblies | 431 |
| CHAPTER 19: Instrumentation | 461 |
| CHAPTER 20: Threads, Tasks, and Synchronization..... | 491 |
| CHAPTER 21: Security | 545 |

Continues

| | | |
|--------------------|---|-----|
| CHAPTER 22: | Localization | 569 |
| CHAPTER 23: | System.Transactions | 605 |
| CHAPTER 24: | Networking | 637 |
| CHAPTER 25: | Windows Services | 667 |
| CHAPTER 26: | Interop | 695 |
| CHAPTER 27: | Core XAML | 727 |
| CHAPTER 28: | Managed Extensibility Framework | 747 |
| CHAPTER 29: | Manipulating Files and the Registry | 771 |

► PART IV: DATA

| | | |
|--------------------|--|-----|
| CHAPTER 30: | Core ADO.NET | 817 |
| CHAPTER 31: | ADO.NET Entity Framework | 861 |
| CHAPTER 32: | Data Services | 885 |
| CHAPTER 33: | Manipulating XML | 903 |
| CHAPTER 34: | .NET Programming with SQL Server | 955 |

► PART V: PRESENTATION

| | | |
|--------------------|--------------------------------------|------|
| CHAPTER 35: | Core WPF | 983 |
| CHAPTER 36: | Business Applications with WPF | 1035 |
| CHAPTER 37: | Creating Documents with WPF | 1075 |
| CHAPTER 38: | Silverlight | 1095 |
| CHAPTER 39: | Windows Forms | 1117 |
| CHAPTER 40: | Core ASP.NET | 1149 |
| CHAPTER 41: | ASP.NET Features | 1185 |
| CHAPTER 42: | ASP.NET Dynamic Data and MVC | 1243 |

► PART VI: COMMUNICATION

| | | |
|--------------------|--|------|
| CHAPTER 43: | Windows Communication Foundation | 1279 |
| CHAPTER 44: | Windows Workflow Foundation 4 | 1309 |
| CHAPTER 45: | Peer-to-Peer Networking | 1339 |
| CHAPTER 46: | Message Queuing | 1357 |
| CHAPTER 47: | Syndication | 1387 |

APPENDIX: Guidelines for Windows 7 and Windows Server 2008 R2 1397

INDEX.1417

► ONLINE CHAPTERS

CHAPTER 48: Graphics with GDI+OC1

CHAPTER 49: Visual Studio Tools for Office.....OC49

CHAPTER 50: Managed Add-In Framework..... OC77

CHAPTER 51: Enterprise Services OC97

CHAPTER 52: Directory Services OC123

CHAPTER 53: C#, Visual Basic, C++/CLI, and F# OC157

CHAPTER 54: .NET RemotingOC191

CHAPTER 55: Web Services with ASP.NETOC233

CHAPTER 56: LINQ to SQLOC255

CHAPTER 57: Windows Workflow Foundation 3.0OC279

PROFESSIONAL

C# 4 and .NET 4

Christian Nagel
Bill Evjen
Jay Glynn
Karli Watson
Morgan Skinner



WILEY

Wiley Publishing, Inc.

Professional C# 4 and .NET 4

Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Copyright © 2010 by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-0-470-50225-9

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Library of Congress Control Number: 2009942439

Trademarks: Wiley, the Wiley logo, Wrox, the Wrox logo, Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book.

To my two girls, Angela and Stephanie

The first half of the year 2009 was the hardest time in my life — a trip through hell and back. I cannot thank Angela enough for all her love and support she gave to me. Without her I couldn't have made it through that.

Stephanie was born shortly after medical treatment and was the biggest motivation during that time.

I love you both!

—CHRISTIAN NAGEL

To Tuija, always.

—BILL EVJEN

For donna

—KARLI WATSON

Dedicated to my parents, Joan & Donald Skinner.

There are many things that I'd like to say to my Mum and Dad, but I'm in the privileged position of not needing to say anything to them. They were both wonderful people, are greatly missed, and the world was made a sweeter place by their being in it. Thanks Mum, thanks Dad, you were brilliant.

—MORGAN SKINNER

ABOUT THE AUTHORS



CHRISTIAN NAGEL is a Microsoft Regional Director and Microsoft MVP, an associate of thinkecture, and owner of CN innovation. He is a software architect and developer who offers training and consulting on how to develop Microsoft .NET solutions. He looks back on more than 25 years of software development experience. Christian started his computing career with PDP 11 and VAX/VMS systems, covering a variety of languages and platforms. Since 2000, when .NET was just a technology preview, he has been working with various .NET technologies to build numerous .NET solutions. With his profound knowledge of Microsoft technologies, he has written numerous .NET books, and is certified as a Microsoft Certified Trainer and Professional Developer. Christian speaks at

international conferences such as TechEd and Tech Days, and started INETA Europe to support .NET user groups. You can contact Christian via his web sites, www.cninnovation.com and www.thinkecture.com, and follow his tweets on www.twitter.com/christiannagel.



BILL EVJEN is an active proponent of .NET technologies and community-based learning initiatives for .NET. He has been actively involved with .NET since the first bits were released in 2000. In the same year, Bill founded the St. Louis .NET User Group (www.stlnet.org), one of the world's first such groups. Bill is also the founder and former executive director of the International .NET Association (www.ineta.org), which represents more than 500,000 members worldwide.

Based in St. Louis, Missouri, Bill is an acclaimed author and speaker on ASP.NET and Web Services. He has authored or coauthored more than 20 books including *Professional ASP.NET 4*, *Professional VB 2008*, *ASP.NET Professional Secrets*,

XML Web Services for ASP.NET, and *Web Services Enhancements: Understanding the WSE for Enterprise Applications* (all published by Wiley). In addition to writing, Bill is a speaker at numerous conferences, including DevConnections, VSLive, and TechEd. Along with these activities, Bill works closely with Microsoft as a Microsoft Regional Director and an MVP.

Bill is the Global Head of Platform Architecture for Thomson Reuters, Lipper, the international news and financial services company (www.thomsonreuters.com). He graduated from Western Washington University in Bellingham, Washington, with a Russian language degree. When he isn't tinkering on the computer, he can usually be found at his summer house in Toivakka, Finland. You can reach Bill on Twitter at @billevjen.



JAY GLYNN is the Principle Architect at PureSafety, a leading provider of results-driven software and information solutions for workforce safety and health. Jay has been developing software for over 25 years and has worked with a variety of languages and technologies including PICK Basic, C, C++, Visual Basic, C# and Java. Jay currently lives in Franklin, Tennessee with his wife and son.



KARLI WATSON is consultant at Infusion Development (www.infusion.com), a technology architect at Boost.net (www.boost.net), and a freelance IT specialist, author, and developer. For the most part, he immerses himself in .NET (in particular C# and lately WPF) and has written numerous books in the field. He specializes in communicating complex ideas in a way that is accessible to anyone with a passion to learn, and spends much of his time playing with new technology to find new things to teach people about.

During those (seemingly few) times where he isn't doing the above, Karli will probably be wishing he was hurtling down a mountain on a snowboard. Or possibly trying to get his novel published. Either way, you'll know him by his brightly colored clothes. You can also find him tweeting online at www.twitter.com/karlequin, and maybe one day he'll get round to making himself a web site.



MORGAN SKINNER began his computing career at a young age on the Sinclair ZX80 at school, where he was underwhelmed by some code a teacher had written and so began programming in assembly language. Since then he's used all sorts of languages and platforms, including VAX Macro Assembler, Pascal, Modula2, Smalltalk, X86 assembly language, PowerBuilder, C/C++, VB, and currently C# (of course). He's been programming in .NET since the PDC release in 2000, and liked it so much he joined Microsoft in 2001. He now works in premier support for developers and spends most of his time assisting customers with C#. You can reach Morgan at www.morganskinner.com.

ABOUT THE TECHNICAL EDITORS

ALEXEI GORKOV is the chief software engineer at EPAM Systems (www.epam.com), a leading software development outsourcing company in Central and Eastern Europe. He has worked with .NET since 2004 and as a front-end developer involved in development of web portals and line-of-business web applications using Microsoft technologies. Over the past four years, he has edited more than a dozen programming books from Wiley Publishing on ASP.NET, Ajax, JavaScript, CSS and XML. He lives in Saratov, Russia.

MITCHEL SELLERS is the CEO of IowaComputerGurus Inc. His extensive experience in software development has lead to a focus on proper architecture, performance, stability, and overall cost effectiveness of delivered solutions. He is the author of *Professional DotNetNuke Module Programming* and many technical articles, and is currently working on his next book, *Visual Studio 2010 and .NET 4 Six-in-One*, scheduled for publication by Wiley in early 2010. He is a regular speaker at user groups and conferences.

Mitchel is a Microsoft C# MVP and a Microsoft Certified Professional, an active participant in the .NET and DotNetNuke development communities, and a DotNetNuke Core Team member. For more information, see his resume at MitchelSellers.com.

CREDITS

ACQUISITIONS EDITOR

Paul Reese

PROJECT EDITOR

Sara Shlaer

DEVELOPMENT EDITOR

Susan Cohen

TECHNICAL EDITORS

Alexei Gorkov

Mitchel Sellers

PRODUCTION EDITOR

Kathleen Wisor

COPY EDITORS

Tricia Liebig

Foxxe Editorial Services

EDITORIAL DIRECTOR

Robyn B. Siesky

EDITORIAL MANAGER

Mary Beth Wakefield

MARKETING MANAGER

David Mayhew

PRODUCTION MANAGER

Tim Tate

VICE PRESIDENT AND EXECUTIVE**GROUP PUBLISHER**

Richard Swadley

VICE PRESIDENT AND EXECUTIVE**PUBLISHER**

Barry Pruett

ASSOCIATE PUBLISHER

Jim Minatel

PROJECT COORDINATOR, COVER

Lynsey Stanford

PROOFREADER

Word One New York

INDEXER

Robert Swanson

COVER DESIGNER

Michael E. Trent

COVER IMAGE

© Punchstock/Corbis

ACKNOWLEDGMENTS

THANKS TO ALL AT WILEY for your patience when I started working later on the book than I had expected and planned. Special thanks to Sara Shlaer who was of great support with the time pressure and the change of the Wrox style. Similar to the authors, she worked many weekends to get the book out on time. Many thanks!

—CHRISTIAN NAGEL

THANKS TO SARA SHLAER, Paul Reese, and Jim Minatel for the opportunity to work on such a great book, and to the technical editors, Alexei Gorkov and Mitchel Sellers, for their excellent work. In addition to my co-authors, I would like to thank my family for putting up with all the writing. Thank you Tuija, Sofia, Henri, and Kalle!

—BILL EVJEN

I WOULD LIKE TO THANK Sara Shlaer, Paul Reese, and the rest of the team at Wrox for all the help they provided. I would also like to thank my wife and son for putting up with the nights and weekends lost to the computer. They are my inspiration.

—JAY GLYNN

THANKS TO ALL AT WILEY for their support and assistance on this project, as well as their understanding and flexibility in dealing with an author who never seems to have enough time to write. Special thanks to Sara Shlaer — one of the best and friendliest editors I've had the pleasure to work with. Also, thanks to friends, family, and work colleagues for understanding why I haven't had time for much socializing lately, and to donna, as always, for all her support and for putting up with all the late nights.

—KARLI WATSON

CONTENTS

INTRODUCTION

LI

PART I: THE C# LANGUAGE

CHAPTER 1: .NET ARCHITECTURE

3

| | |
|---|-----------|
| The Relationship of C# to .NET | 3 |
| The Common Language Runtime | 4 |
| Platform Independence | 4 |
| Performance Improvement | 4 |
| Language Interoperability | 5 |
| A Closer Look at Intermediate Language | 6 |
| Support for Object Orientation and Interfaces | 7 |
| Distinct Value and Reference Types | 8 |
| Strong Data Typing | 8 |
| Error Handling with Exceptions | 13 |
| Use of Attributes | 13 |
| Assemblies | 14 |
| Private Assemblies | 14 |
| Shared Assemblies | 15 |
| Reflection | 15 |
| Parallel Programming | 15 |
| .NET Framework Classes | 16 |
| Namespaces | 17 |
| Creating .NET Applications Using C# | 17 |
| Creating ASP.NET Applications | 17 |
| Creating Windows Forms | 19 |
| Using the Windows Presentation Foundation (WPF) | 19 |
| Windows Controls | 19 |
| Windows Services | 20 |
| Windows Communication Foundation | 20 |
| Windows Workflow Foundation | 20 |
| The Role of C# in the .NET Enterprise Architecture | 20 |
| Summary | 21 |

| | |
|---|-----------|
| CHAPTER 2: CORE C# | 23 |
| Your First C# Program | 23 |
| The Code | 24 |
| Compiling and Running the Program | 24 |
| A Closer Look | 25 |
| Variables | 26 |
| Initialization of Variables | 27 |
| Type Inference | 28 |
| Variable Scope | 29 |
| Constants | 31 |
| Predefined Data Types | 31 |
| Value Types and Reference Types | 31 |
| CTS Types | 33 |
| Predefined Value Types | 33 |
| Predefined Reference Types | 35 |
| Flow Control | 37 |
| Conditional Statements | 37 |
| Loops | 40 |
| Jump Statements | 43 |
| Enumerations | 44 |
| Namespaces | 45 |
| The using Directive | 46 |
| Namespace Aliases | 47 |
| The Main() Method | 48 |
| Multiple Main() Methods | 48 |
| Passing Arguments to Main() | 49 |
| More on Compiling C# Files | 50 |
| Console I/O | 51 |
| Using Comments | 53 |
| Internal Comments within the Source Files | 53 |
| XML Documentation | 53 |
| The C# Preprocessor Directives | 55 |
| #define and #undef | 55 |
| #if, #elif, #else, and #endif | 56 |
| #warning and #error | 57 |
| #region and #endregion | 57 |
| #line | 57 |
| #pragma | 58 |
| C# Programming Guidelines | 58 |
| Rules for Identifiers | 58 |
| Usage Conventions | 59 |
| Summary | 64 |

| | |
|---|------------|
| CHAPTER 3: OBJECTS AND TYPES | 65 |
| Classes and Structs | 65 |
| Classes | 66 |
| Data Members | 66 |
| Function Members | 67 |
| readonly Fields | 78 |
| Anonymous Types | 79 |
| Structs | 80 |
| Structs Are Value Types | 81 |
| Structs and Inheritance | 82 |
| Constructors for Structs | 82 |
| Partial Classes | 82 |
| Static Classes | 84 |
| The Object Class | 84 |
| System.Object Methods | 84 |
| The ToString() Method | 85 |
| Extension Methods | 86 |
| Summary | 87 |
| CHAPTER 4: INHERITANCE | 89 |
| Types of Inheritance | 89 |
| Implementation Versus Interface Inheritance | 89 |
| Multiple Inheritance | 90 |
| Structs and Classes | 90 |
| Implementation Inheritance | 90 |
| Virtual Methods | 91 |
| Hiding Methods | 92 |
| Calling Base Versions of Functions | 93 |
| Abstract Classes and Functions | 93 |
| Sealed Classes and Methods | 94 |
| Constructors of Derived Classes | 94 |
| Modifiers | 99 |
| Visibility Modifiers | 99 |
| Other Modifiers | 99 |
| Interfaces | 100 |
| Defining and Implementing Interfaces | 101 |
| Derived Interfaces | 104 |
| Summary | 105 |

| | |
|---|------------|
| CHAPTER 5: GENERICS | 107 |
| Generics Overview | 107 |
| Performance | 108 |
| Type Safety | 109 |
| Binary Code Reuse | 109 |
| Code Bloat | 109 |
| Naming Guidelines | 110 |
| Creating Generic Classes | 110 |
| Generics Features | 113 |
| Default Values | 114 |
| Constraints | 114 |
| Inheritance | 117 |
| Static Members | 117 |
| Generic Interfaces | 118 |
| Covariance and Contra-variance | 118 |
| Covariance with Generic Interfaces | 119 |
| Contra-Variance with Generic Interfaces | 120 |
| Generic Structs | 121 |
| Generic Methods | 123 |
| Generic Methods Example | 124 |
| Generic Methods with Constraints | 125 |
| Generic Methods with Delegates | 126 |
| Generic Methods Specialization | 126 |
| Summary | 128 |
| CHAPTER 6: ARRAYS AND TUPLES | 129 |
| Simple Arrays | 129 |
| Array Declaration | 129 |
| Array Initialization | 130 |
| Accessing Array Elements | 130 |
| Using Reference Types | 131 |
| Multidimensional Arrays | 132 |
| Jagged Arrays | 133 |
| Array Class | 134 |
| Creating Arrays | 134 |
| Copying Arrays | 135 |
| Sorting | 136 |
| Arrays as Parameters | 139 |
| Array Covariance | 139 |
| ArraySegment<T> | 140 |
| Enumerations | 140 |

| | |
|---|------------|
| IEnumerator Interface | 141 |
| foreach Statement | 141 |
| yield Statement | 142 |
| Tuples | 146 |
| Structural Comparison | 147 |
| Summary | 149 |
| CHAPTER 7: OPERATORS AND CASTS | 151 |
| Operators | 151 |
| Operator Shortcuts | 153 |
| Operator Precedence | 157 |
| Type Safety | 157 |
| Type Conversions | 158 |
| Boxing and Unboxing | 161 |
| Comparing Objects for Equality | 162 |
| Comparing Reference Types for Equality | 162 |
| Comparing Value Types for Equality | 163 |
| Operator Overloading | 163 |
| How Operators Work | 164 |
| Operator Overloading Example: The Vector Struct | 165 |
| Which Operators Can You Overload? | 171 |
| User-Defined Casts | 172 |
| Implementing User-Defined Casts | 173 |
| Multiple Casting | 179 |
| Summary | 182 |
| CHAPTER 8: DELEGATES, LAMBDA, AND EVENTS | 183 |
| Delegates | 183 |
| Declaring Delegates | 184 |
| Using Delegates | 185 |
| Simple Delegate Example | 188 |
| Action<T> and Func<T> Delegates | 190 |
| BubbleSorter Example | 190 |
| Multicast Delegates | 193 |
| Anonymous Methods | 196 |
| Lambda Expressions | 197 |
| Parameters | 198 |
| Multiple Code Lines | 198 |
| Variables Outside of the Lambda Expression | 199 |
| Events | 200 |

| | |
|---|------------|
| Event Publisher | 200 |
| Event Listener | 202 |
| Weak Events | 203 |
| Summary | 205 |
| CHAPTER 9: STRINGS AND REGULAR EXPRESSIONS | 207 |
| <hr/> | |
| Examining System.String | 208 |
| Building Strings | 209 |
| StringBuilder Members | 211 |
| Format Strings | 212 |
| Regular Expressions | 217 |
| Introduction to Regular Expressions | 218 |
| The RegularExpressionsPlayaround Example | 219 |
| Displaying Results | 221 |
| Matches, Groups, and Captures | 223 |
| Summary | 224 |
| CHAPTER 10: COLLECTIONS | 225 |
| <hr/> | |
| Collection Interfaces and Types | 225 |
| Lists | 226 |
| Creating Lists | 228 |
| Read-Only Collections | 236 |
| Queue | 236 |
| Stack | 240 |
| Linked List | 241 |
| Sorted List | 246 |
| Dictionaries | 248 |
| Key Type | 248 |
| Dictionary Example | 250 |
| Lookup | 253 |
| Sorted Dictionary | 254 |
| Sets | 255 |
| Observable Collection | 256 |
| Bit Arrays | 258 |
| BitArray | 258 |
| BitVector32 | 260 |
| Concurrent Collections | 262 |
| Performance | 264 |
| Summary | 266 |

| | |
|--|------------|
| CHAPTER 11: LANGUAGE INTEGRATED QUERY | 267 |
| LINQ Overview | 267 |
| Lists and Entities | 267 |
| LINQ Query | 271 |
| Extension Methods | 272 |
| Deferred Query Execution | 273 |
| Standard Query Operators | 275 |
| Filtering | 277 |
| Filtering with Index | 277 |
| Type Filtering | 278 |
| Compound from | 278 |
| Sorting | 279 |
| Grouping | 280 |
| Grouping with Nested Objects | 281 |
| Join | 282 |
| Set Operations | 283 |
| Zip | 284 |
| Partitioning | 285 |
| Aggregate Operators | 286 |
| Conversion | 287 |
| Generation Operators | 288 |
| Parallel LINQ | 289 |
| Parallel Queries | 289 |
| Partitioners | 290 |
| Cancellation | 290 |
| Expression Trees | 291 |
| LINQ Providers | 293 |
| Summary | 294 |
| CHAPTER 12: DYNAMIC LANGUAGE EXTENSIONS | 295 |
| Dynamic Language Runtime | 295 |
| The Dynamic Type | 296 |
| Dynamic Behind the Scenes | 297 |
| Hosting the DLR ScriptRuntime | 300 |
| DynamicObject and ExpandoObject | 302 |
| DynamicObject | 302 |
| ExpandoObject | 304 |
| Summary | 305 |

| | |
|---|------------|
| CHAPTER 13: MEMORY MANAGEMENT AND POINTERS | 307 |
| Memory Management Under the Hood | 307 |
| Value Data Types | 308 |
| Reference Data Types | 309 |
| Garbage Collection | 311 |
| Freeing Unmanaged Resources | 312 |
| Destructors | 313 |
| The IDisposable Interface | 314 |
| Implementing IDisposable and a Destructor | 315 |
| Unsafe Code | 317 |
| Accessing Memory Directly with Pointers | 317 |
| Pointer Example: PointerPlayground | 325 |
| Using Pointers to Optimize Performance | 329 |
| Summary | 332 |
| CHAPTER 14: REFLECTION | 333 |
| Custom Attributes | 334 |
| Writing Custom Attributes | 334 |
| Custom Attribute Example: WhatsNewAttributes | 337 |
| Using Reflection | 340 |
| The System.Type Class | 340 |
| The TypeView Example | 342 |
| The Assembly Class | 344 |
| Completing the WhatsNewAttributes Example | 345 |
| Summary | 349 |
| CHAPTER 15: ERRORS AND EXCEPTIONS | 351 |
| Exception Classes | 352 |
| Catching Exceptions | 353 |
| Implementing Multiple Catch Blocks | 355 |
| Catching Exceptions from Other Code | 359 |
| System.Exception Properties | 359 |
| What Happens If an Exception Isn't Handled? | 360 |
| Nested try Blocks | 360 |
| User-Defined Exception Classes | 362 |
| Catching the User-Defined Exceptions | 363 |
| Throwing the User-Defined Exceptions | 364 |
| Defining the User-Defined Exception Classes | 367 |
| Summary | 369 |

PART II: VISUAL STUDIO

| | |
|---|------------|
| CHAPTER 16: VISUAL STUDIO 2010 | 373 |
| Working with Visual Studio 2010 | 373 |
| Creating a Project | 378 |
| Distinguishing Projects from Solutions | 383 |
| Windows Application Code | 386 |
| Exploring and Coding a Project | 386 |
| Building a Project | 394 |
| Debugging Your Code | 398 |
| Refactoring Tools | 401 |
| Multi-Targeting the .NET Framework | 403 |
| WPF, WCF, WF, and More | 404 |
| Building WPF Applications in Visual Studio 2010 | 404 |
| Building WF Applications in Visual Studio 2010 | 405 |
| Summary | 406 |
| CHAPTER 17: DEPLOYMENT | 407 |
| Planning for Deployment | 407 |
| Deployment Options | 408 |
| Deployment Requirements | 408 |
| Deploying the .NET Runtime | 409 |
| Simple Deployment Options | 409 |
| Xcopy Deployment | 410 |
| Xcopy and Web Applications | 411 |
| Publishing a Web Site | 411 |
| Visual Studio 2010 Setup and Deployment Projects | 412 |
| What is Windows Installer? | 412 |
| Creating Installers | 413 |
| ClickOnce | 419 |
| ClickOnce Operation | 419 |
| Publishing a ClickOnce Application | 420 |
| ClickOnce Settings | 420 |
| Application Cache for ClickOnce Files | 421 |
| Application Security | 421 |
| Visual Studio 2010 Editors | 422 |
| File System Editor | 422 |
| Registry Editor | 422 |
| File Types Editor | 423 |
| User Interface Editor | 423 |

| | |
|--------------------------|------------|
| Custom Actions Editor | 424 |
| Launch Conditions Editor | 426 |
| Summary | 427 |

PART III: FOUNDATION

CHAPTER 18: ASSEMBLIES **431**

| | |
|---|------------|
| What Are Assemblies? | 431 |
| Assembly Features | 432 |
| Assembly Structure | 432 |
| Assembly Manifests | 433 |
| Namespaces, Assemblies, and Components | 434 |
| Private and Shared Assemblies | 434 |
| Satellite Assemblies | 434 |
| Viewing Assemblies | 434 |
| Creating Assemblies | 435 |
| Creating Modules and Assemblies | 435 |
| Assembly Attributes | 436 |
| Creating and Loading Assemblies Dynamically | 438 |
| Application Domains | 441 |
| Shared Assemblies | 445 |
| Strong Names | 445 |
| Integrity Using Strong Names | 446 |
| Global Assembly Cache | 446 |
| Creating a Shared Assembly | 447 |
| Create a Strong Name | 447 |
| Installing the Shared Assembly | 448 |
| Using the Shared Assembly | 448 |
| Delayed Signing of Assemblies | 449 |
| References | 450 |
| Native Image Generator | 451 |
| Configuring .NET Applications | 452 |
| Configuration Categories | 452 |
| Binding to Assemblies | 453 |
| Versioning | 454 |
| Version Numbers | 455 |
| Getting the Version Programmatically | 455 |
| Binding to Assembly Versions | 456 |
| Publisher Policy Files | 457 |
| Runtime Version | 458 |
| Summary | 458 |

| | |
|--|------------|
| CHAPTER 19: INSTRUMENTATION | 461 |
| Code Contracts | 461 |
| Preconditions | 463 |
| Postconditions | 464 |
| Invariants | 465 |
| Contracts for Interfaces | 465 |
| Tracing | 467 |
| Trace Sources | 468 |
| Trace Switches | 469 |
| Trace Listeners | 470 |
| Filters | 471 |
| Correlation | 472 |
| Event Logging | 475 |
| Event-Logging Architecture | 476 |
| Event-Logging Classes | 477 |
| Creating an Event Source | 478 |
| Writing Event Logs | 479 |
| Resource Files | 479 |
| Performance Monitoring | 483 |
| Performance-Monitoring Classes | 483 |
| Performance Counter Builder | 484 |
| Adding PerformanceCounter Components | 486 |
| perfmon.exe | 488 |
| Summary | 489 |
| CHAPTER 20: THREADS, TASKS, AND SYNCHRONIZATION | 491 |
| Overview | 492 |
| Asynchronous Delegates | 492 |
| Polling | 493 |
| Wait Handle | 493 |
| Asynchronous Callback | 494 |
| The Thread Class | 495 |
| Passing Data to Threads | 496 |
| Background Threads | 497 |
| Thread Priority | 498 |
| Controlling Threads | 499 |
| Thread Pools | 499 |
| Tasks | 501 |
| Starting Tasks | 501 |
| Continuation Tasks | 502 |

| | |
|---|------------|
| Task Hierarchies | 502 |
| Results from Tasks | 503 |
| Parallel Class | 504 |
| Looping with the Parallel.For Method | 504 |
| Looping with the Parallel.ForEach Method | 506 |
| Invoking Multiple Methods with the Parallel.Invoke Method | 507 |
| Cancellation Framework | 507 |
| Cancellation of Parallel.For | 507 |
| Cancellation of Tasks | 509 |
| Threading Issues | 510 |
| Race Condition | 510 |
| Deadlock | 513 |
| Synchronization | 514 |
| lock Statement and Thread Safety | 515 |
| Interlocked | 520 |
| Monitor | 521 |
| SpinLock | 522 |
| WaitHandle | 522 |
| Mutex | 523 |
| Semaphore | 524 |
| Events | 526 |
| Barrier | 529 |
| ReaderWriterLockSlim | 530 |
| Timers | 533 |
| Event-Based Asynchronous Pattern | 534 |
| BackgroundWorker | 535 |
| Creating an Event-Based Asynchronous Component | 539 |
| Summary | 543 |
| CHAPTER 21: SECURITY | 545 |
| Authentication and Authorization | 545 |
| Identity and Principal | 545 |
| Roles | 547 |
| Declarative Role-Based Security | 547 |
| Client Application Services | 548 |
| Encryption | 552 |
| Signature | 554 |
| Key Exchange and Secure Transfer | 556 |
| Access Control to Resources | 558 |
| Code Access Security | 561 |
| Security Transparency Level 2 | 562 |

| | |
|---|------------|
| Permissions | 562 |
| Distributing Code Using Certificates | 567 |
| Summary | 568 |
| CHAPTER 22: LOCALIZATION | 569 |
| Namespace System.Globalization | 569 |
| Unicode Issues | 570 |
| Cultures and Regions | 570 |
| Cultures in Action | 574 |
| Sorting | 578 |
| Resources | 579 |
| Creating Resource Files | 579 |
| Resource File Generator | 579 |
| ResourceWriter | 580 |
| Using Resource Files | 581 |
| The System.Resources Namespace | 584 |
| Windows Forms Localization Using Visual Studio | 584 |
| Changing the Culture Programmatically | 588 |
| Using Custom Resource Messages | 590 |
| Automatic Fallback for Resources | 590 |
| Outsourcing Translations | 591 |
| Localization with ASP.NET | 591 |
| Localization with WPF | 593 |
| .NET Resources with WPF | 594 |
| XAML Resource Dictionaries | 594 |
| A Custom Resource Reader | 598 |
| Creating a DatabaseResourceReader | 598 |
| Creating a DatabaseResourceSet | 600 |
| Creating a DatabaseResourceManager | 600 |
| Client Application for DatabaseResourceReader | 601 |
| Creating Custom Cultures | 601 |
| Summary | 603 |
| CHAPTER 23: SYSTEM.TRANSACTIONS | 605 |
| Overview | 605 |
| Transaction Phases | 606 |
| ACID Properties | 607 |
| Database and Entity Classes | 607 |
| Traditional Transactions | 609 |
| ADO.NET Transactions | 609 |

| | |
|--|----------------|
| System.EnterpriseServices | 610 |
| System.Transactions | 611 |
| Committable Transactions | 612 |
| Transaction Promotion | 614 |
| Dependent Transactions | 616 |
| Ambient Transactions | 618 |
| Isolation Level | 624 |
| Custom Resource Managers | 626 |
| Transactional Resources | 627 |
| Transactions with Windows 7 and Windows Server 2008 | 632 |
| Summary | 635 |
| CHAPTER 24: NETWORKING | 637 |
| The WebClient Class | 638 |
| Downloading Files | 638 |
| Basic WebClient Example | 638 |
| Uploading Files | 639 |
| WebRequest and WebResponse Classes | 640 |
| Authentication | 642 |
| Working with Proxies | 642 |
| Asynchronous Page Requests | 642 |
| Displaying Output as an HTML Page | 643 |
| Allowing Simple Web Browsing from Your Applications | 643 |
| Launching Internet Explorer Instances | 645 |
| Giving Your Application More IE-Type Features | 645 |
| Printing Using the WebBrowser Control | 651 |
| Displaying the Code of a Requested Page | 651 |
| The WebRequest and WebResponse Classes Hierarchy | 652 |
| Utility Classes | 653 |
| URIs | 653 |
| IP Addresses and DNS Names | 654 |
| Lower-Level Protocols | 656 |
| Using SmtpClient | 657 |
| Using the TCP Classes | 658 |
| The TcpSend and TcpReceive Examples | 658 |
| TCP Versus UDP | 660 |
| The UDP Class | 661 |
| The Socket Class | 661 |
| Summary | 665 |