Accelerated VB 2008

Guy Fouché and Trey Nash

Apress®

Accelerated VB 2008

ISBN-13 (pbk): 978-1-59059-874-0

ISBN-10 (pbk): 1-59059-874-1

ISBN-13 (electronic): 978-1-4302-0339-1

ISBN-10 (electronic): 1-4302-0339-0

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

Lead Editor: Dominic Shakeshaft
Technical Reviewers: Tim Patrick, Fabio Claudio Ferracchiati
Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Kevin Goff, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Project Manager: Sofia Marchant
Copy Editor: Kim Benbow
Assistant Production Director: Kari Brooks-Copony
Production Editor: Kelly Gunther
Compositor: Lynn L'Heureux
Proofreader: Elizabeth Welch
Indexer: Present Day Indexing
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski
Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit http://www.springeronline.com.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit http://www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at http://www.apress.com/info/bulksales.

The source code for this book is available to readers at http://www.apress.com.

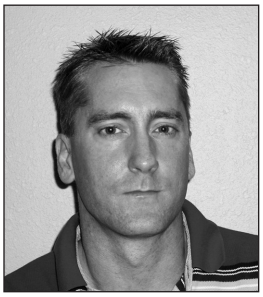
To Jim and Kay Liegl for their friendship and the jaunts in the Jeep
To Charlotte Fouché for her laughter and her compassion toward others
To Gaston Fouché Sr. for his cuisine and sharing the art of the meal
To Frank Reed for the music and trumpet duets after my lessons were long over
To Jodi Fouché for her poetry, being my biggest fan, and unequivocal love
—Guy Fouché

Contents at a Glance

Contents

About the Authors

GUY FOUCHÉ is a business intelligence and decision support system consultant in the Dallas, Texas, area. He has developed a large number of Visual Basic systems in a variety of industries, supporting companies of all shapes and sizes. His VB programming experience dates back to Version 1. Yes, Version 1. Guy spends his evenings playing one of his eight trumpets and expanding his composition skills using the current generation of music technologies. On the weekend, he puts as many miles as he can on his bright yellow Honda F4i sport motorcycle. Guy and Jodi enjoy taking nine-day trips in their Jeep 4 × 4, taking photographs and writing travelogues along the way. You can view their photography at `http://photography.fouche.ws`.



TREY NASH is a principal software engineer working on security solutions at a market-leading security software company. Prior to that, he developed Bluetooth solutions for the release of Microsoft Vista, and he called Macromedia Inc. home for five years before that. At Macromedia, he worked on a cross-product engineering team for several years, designing solutions for a wide range of products throughout the company, including Flash and Fireworks. He specialized in COM/DCOM using C/C++/ATL until the .NET revolution. He's been glued to computers ever since he scored his first, a TI-99/4A, when he was a mere 13 years old. He astounded his parents by turning a childhood obsession into a decently paying career, much to their dismay. Trey received his bachelor of science and his master of engineering degrees in electrical engineering from Texas A&M University. When he's not sitting in front of a computer, you can find him working in his garage, playing his piano, brushing up on a foreign language (Russian and Icelandic are the current favorites), or playing ice hockey.

About the Technical Reviewers

TIM PATRICK has been working professionally as a software architect and developer for nearly 25 years. By day he develops custom business applications in Visual Basic for small to medium-sized organizations. He is a Microsoft Certified Solution Developer (MCSD). In April 2007, Microsoft awarded Tim with its Most Valuable Professional (MVP) award for his work in supporting and promoting Visual Basic and its community of users. Tim received his undergraduate degree in computer science from Seattle Pacific University. You can contact him through his web site, `www.timaki.com`.

FABIO CLAUDIO FERRACCHIATI is a senior developer for Brain Force (`www.brainforce.com`). A prolific writer on leading-edge technologies, he's contributed to more than a dozen books on .NET, C#, Visual Basic, and ASP.NET. He's a .NET MCSD and lives in Milan, Italy.

Acknowledgments

We offer a huge thank you to everyone at Apress who has had input into these pages!

Guy Fouché and Trey Nash

In addition, a note of my appreciation goes to the following people:

Kim Benbow and Kelly Gunther for many a well-chosen edit and making these pages look their best.

Tim Patrick and Fabio Ferracchiati: their insights have added a great deal to this book.

Sofia Marchant for managing countless revisions and keeping us all on track.

Dominic Shakeshaft for his knowledgeable, experienced guidance throughout this edition.

Guy Fouché

Introduction

Visual Basic 2008 (VB 2008) is relatively easy to learn for anyone familiar with another object-oriented language. Even someone familiar with Visual Basic 6.0 who is looking for an object-oriented language will find VB 2008 easy to pick up. However, though VB 2008, coupled with .NET, provides a quick path for creating simple applications, you still must know a wealth of information and understand how to use it correctly in order to produce sophisticated, robust, fault-tolerant applications. We teach you what you need to know and explain how best to use your knowledge so that you can quickly develop true VB 2008 expertise.

Idioms and design patterns are invaluable for developing and applying expertise, and we show you how to use many of them to create applications that are efficient, robust, fault-tolerant, and exception-safe. Although many are familiar to C++ and Java programmers, some are unique to .NET and the Common Language Runtime (CLR). We show you how to apply these indispensable idioms and design techniques to seamlessly integrate your VB 2008 applications with the .NET runtime, focusing on the new capabilities of VB 2008.

Design patterns document best practices in application design that many different programmers have discovered and rediscovered over time. In fact, .NET itself implements many well-known design patterns. You will see these practices detailed throughout this book. Also, it is important to note that the invaluable tool chest of techniques is evolving constantly.

.NET 3.5 provides a unique and stable cross-platform execution environment. VB 2008 is only one of the languages that target this powerful runtime. You will find that many of the techniques explored in this book are also applicable to any language that targets the .NET runtime.

As you'll see, it doesn't take years of trial-and-error experience to become a VB 2008 expert. You simply need to learn about the right tools and the correct ways to use them. That's why we wrote this book for you.

About This Book

We assume that you already have a working knowledge of some object-oriented programming language, such as C++, Java, or Visual Basic. If you already know some VB 2005 or VB 2008, you may find yourself skimming Chapters 1 and 2.

Chapter 1, "VB 2008 Overview," gives a quick glimpse of what a simple VB 2008 application looks like.

Chapter 2, "VB 2008 Syntax," introduces the VB 2008 language syntax. We introduce you to the two fundamental kinds of types within the CLR: value types and reference types. We also describe namespaces and how you can use them to logically partition types and functionality within your applications.

Chapter 3, "Classes and Structures," provides details about defining types in VB 2008. You'll learn more about value types and reference types in the CLR. We also discuss the inefficiencies inherent in boxing and discuss object creation, initialization, and destruction.

Chapter 4, "Methods, Properties, and Fields," discusses using methods to add behavior to your types, using properties to enforce encapsulation, and using fields to represent the state of your object. You'll explore method parameter types, overloading, property modifiers, and field initializers.

Chapter 5, "VB 2008 and the CLR," expands on Chapter 1 and quickly explores the managed environment within which VB 2008 applications run. We introduce you to assemblies, which are the basic building blocks of applications into which VB 2008 code files are compiled. Additionally, you'll see how metadata makes assemblies self-describing.

Chapter 6, "Interfaces," details interfaces and the role they play in the VB 2008 language. Interfaces provide a functionality contract that types may choose to implement. You'll learn the various ways that a type may implement an interface, as well as how the runtime chooses which methods to call when an interface method is called.

Chapter 7, "Operator Overloading," details how you may provide custom functionality for the built-in operators of the VB 2008 language when applied to your own defined types. You'll see how to overload operators responsibly, since not all managed languages that compile code for the CLR are able to use overloaded operators.

Chapter 8, "Exception Handling," shows you the exception-handling capabilities of the VB 2008 language and the CLR. Creating exception-safe and exception-neutral code is tricky in VB 2008, and you'll see that creating fault-tolerant, exception-safe code doesn't require the use of `Try`, `Catch`, or `Finally` constructs at all. We also describe some of the capabilities within the .NET runtime that allow you to create more fault-tolerant code.

Chapter 9, "Working with Strings," describes how strings are a first-class type in the CLR and how to use them effectively in VB 2008. A large portion of the chapter covers the string-formatting capabilities of various types in the .NET Framework and how to make your defined types behave similarly by implementing `IFormattable`. Additionally, we introduce you to the globalization capabilities of the framework and show you how to create custom `CultureInfo` instances for cultures and regions that the .NET Framework doesn't already know about.

Chapter 10, "Arrays and Collections," covers the various array and collection types available in VB 2008. You can create two types of multidimensional arrays, as well as your own collection types, while utilizing collection-utility classes. You'll also learn how to implement `IEnumerable` so that your collection types will work well with `For . . . Each` statements.

Chapter 11, "Delegates and Events," shows you the mechanisms used within VB 2008 to provide callbacks. Historically, all viable frameworks have always provided a mechanism to implement callbacks. VB 2008 goes one step further and encapsulates callbacks into callable objects called *delegates*. Also, you'll see how the .NET Framework builds upon delegates to provide a publish-subscribe event-notification mechanism, allowing your design to decouple the source of the event from the consumer of the event.

Chapter 12, "Generics," introduces you to probably the most exciting feature added to VB 2008 and the CLR. Using generics, you can provide a shell of functionality within which to define more specific types at run time. Generics are most useful with collection types and provide great efficiency compared to the collections of previous .NET versions.

Chapter 13, "Threading," covers the tasks required in creating multithreaded applications in the VB 2008 managed virtual execution environment. You'll see how delegates, through use of the "I owe you" (IOU) pattern, provide an excellent gateway into the process thread pool. Arguably, synchronization is the most important concept when getting multiple threads to run concurrently. This chapter covers the various synchronization facilities available to your applications.

Chapter 14, "VB 2008 Best Practices," is a dissertation on the best design practices for defining new types and how to make them so you can use them naturally and so consumers won't abuse them inadvertently. We touch upon some of these topics in other chapters, but discuss them in detail in this chapter. This chapter concludes with a checklist of items to consider when defining new types using VB 2008.

Chapter 15, "LINQ with VB 2008," explores a new set of technologies built into the .NET 3.5 Framework. LINQ provides a common object model and syntax to consume data with your VB 2008 applications. This chapter covers LINQ to Objects, LINQ to XML, and LINQ to SQL, showing you how to query in-memory objects, XML documents, and relational databases with these technologies. We also discuss several technologies that support LINQ, including type inference, anonymous types, extension methods, and Lambda expressions.