

Introduction

Finding a reference book on Microsoft Visual Basic 2005 is easy. By the time this book reaches the bookstores, you'll find plenty of Visual Basic books on the shelves. Why should you buy this one? What makes this book different? When I began to write this book, I asked myself similar questions. How can a book compete with Microsoft Visual Studio manuals and all the samples and tips you can find on the Internet? To answer this question I need to take a short historical detour.

Where Visual Basic is coming from and heading to

I have been teaching Visual Basic since the early 1990s, well before it became Visual Basic .NET, and I taught (and wrote about) Microsoft QuickBasic before then. I have seen this language evolve from the time that you were practically compelled to use `GoTo` statements to make things work up to today's phenomenal object-oriented features. Everything has changed in these 20 years, except one thing: developers have always underutilized—or even misused—this language.

For instance, Microsoft Visual Basic 4 was the first version to offer the ability to define classes, yet very few developers actually used classes in their applications. The few who did, however, were able to catch the Microsoft .NET Framework wave easier than their colleagues were and could deliver more powerful Visual Basic .NET applications in less time. Another example: Microsoft Visual Basic 6 developers were able to access a database through ActiveX Data Objects (ADO) using client-side recordsets in disconnected mode, but many preferred to ignore this feature and continued to work with easier-to-use but less scalable server-side cursors. (And they had serious problems when writing large client/server applications.)

Versions 2002 and 2003 of Visual Basic .NET are *very* powerful development platforms, yet I see that many developers are missing their full potential. For example, features such as threading, reflection, and custom attributes can really revolutionize the way you write applications, but only a minority of programmers leverage them. The gap between what the language offers and what most developers actually use has always been large, but it is going to become larger with Visual Basic 2005, which offers great new features such as generics, custom events, operator overloading, and many other object-oriented enhancements.

Becoming a better developer

The bottom line is: developers don't need yet another reference manual. Instead—better, in addition to a reference—they need to understand how the old and the new features can be used to create more efficient, robust, reusable, and secure code. In the programming world, you can often achieve the same result with two or more equivalent techniques, but each one has its specific pros and cons, and often selecting the right approach can have far-reaching consequences on the end result. You need more than a mere reference book to gain the knowledge needed to detect these subtle differences.

xxii Introduction

A common misunderstanding in the programming community is that all you need to write great applications is familiarity with the .NET Framework and related technologies, such as Windows Forms and ASP.NET. Granted, you do need to learn these technologies to create real-world programs, but that familiarity isn't a surrogate for in-depth knowledge of low-level mechanisms that enable you to reduce memory and resource consumption, adopt effective optimization techniques, or leverage inheritance to write more concise and reusable code. I have seen too many applications that have a great user interface, yet perform very slowly and aren't structured in an orderly manner. Maintaining and evolving these applications are nightmares and cost much more in time and money than if they had been written with a solid understanding of the .NET Framework basics in mind.

Another facet of programming that many developers tend to overlook is the quest for thorough knowledge of the tool you spend most of your time with: Visual Studio. I find it quite ironic that most developers can argue for hours about which language can be more efficient or productive, yet they fail to leverage Visual Studio to its full potential, for example, by learning how to write macros, templates, and code snippets, or how to customize the IDE to fit their needs or programming style.

Not the usual programming language reference

For all these reasons, I decided that I wanted to write something different from the typical language reference, something that would cover all the language features *and* show real-world cases when these features can be used profitably. The problem with this approach is that it tends to take a lot of space. Clearly, a book on this premise would be remarkably thicker than a standard reference book, and it would take me much longer to write.

If this book would cover the entire Visual Basic potential—including both the language features and higher-level technologies such as Windows Forms and ASP.NET—it would have exceeded the number of pages that Microsoft Press can bind in a book. And it would have hit the streets too many months after the Visual Basic 2005 release.

In the end, I saw that the only realistic solution to this issue was focusing on the language and most of the .NET Framework foundation classes—memory management, serialization, threading, reflection, PInvoke, and COM Interop—and leaving out important topics such as Windows Forms, ASP.NET Web Forms and Web Services, and ADO.NET. It was a painful decision, but now that the book is completed, I am very glad I took this route. This is a book I have had in mind for years, and I finally had the opportunity to write it.



Note As of this writing, I am planning to write at least one other book in the *Programming Microsoft Visual Basic 2005* series, but I haven't finalized the agreement with the publisher, and thus I can't be more precise about the topics I'll cover in a forthcoming book(s). If you want to learn more, read my Weblog or subscribe to my Web site's newsletter. (Information on how to do this is provided later in this introduction.)

Live updates and feedback

Even though I have been working with Visual Basic, C#, and the .NET Framework for so many years, I do continue to learn something new almost every day. You can learn more about my discoveries by visiting my Web site, where I maintain the home page for this and all my other books, at this URL:

<http://www.dotnet2themax.com>

You can also subscribe to the site's newsletter and receive information about new articles and code snippets available online. Or you can read my English Weblog (see Figure I-1) where I post updates about this book, comments from readers, plans for future Microsoft Press books, and so forth:

<http://www.dotnet2themax.com/blogs/fbalena>



Tip Select the Books category to read all posts related to this and other forthcoming books. Writing this book has been a challenge. I think (and hope) I did a good job, but I surely look forward to hearing your comments, reactions, and suggestions for improvements. You can leave a comment at my Weblog or write me at fbalena@dotnet2themax.com or fbalena@codearchitects.com.

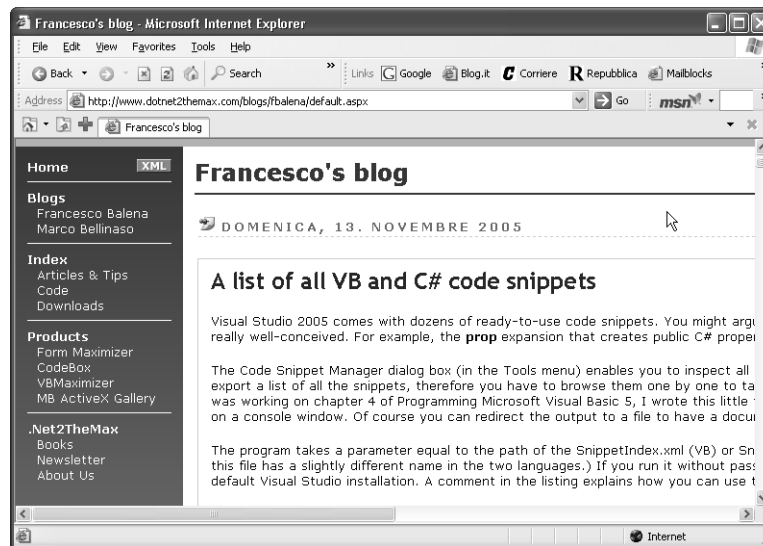


Figure I-1 My Weblog

Who Is This Book For?

The short answer is that this book is for all Visual Basic 2005 developers.

A more articulated answer is that this book is addressed to the following people:

- Developers who have been writing applications with versions 2002 and 2003 of Visual Basic .NET and want to learn all the new features in Visual Basic 2005 as quickly as possible
- Visual Basic 6 programmers who are facing the daunting task of converting their skills and their applications to Visual Basic 2005 and the .NET Framework
- Programmers who are already familiar with another programming language—for example, C, C#, C++, Java, or Borland Delphi—and who want to learn quickly how to write Visual Basic 2005 applications
- Expert Visual Basic developers who want to learn more about advanced .NET Framework programming techniques, such as memory optimization, object serialization, and threading
- Programmers of any expertise level who want to write robust and maintainable applications by leveraging object-oriented features of Visual Basic and other .NET Framework techniques, such as reflection and custom attributes

Of course, not all the chapters in this book will require the same degree of attention from each of the preceding groups. For example, Visual Basic novices will spend most of their time digesting the first half of the book, whereas expert programmers will find the second half more intriguing. Developers coming from edition 2003 of Visual Basic .NET will probably focus on chapters that are interspersed here and there in the book, for example, Chapter 4 (“Using Visual Studio 2005”), Chapter 5 (“Debugging Visual Basic Applications”), Chapter 11 (“Generics”), and Chapter 16 (“The My Namespace”).



Visual Basic 6 developers switching to Visual Studio 2005 should carefully read the sections marked with this icon. In these sections, I focus on the important differences between these two languages as well as subtle issues you might face when migrating a Visual Basic 6 application to the .NET Framework.



Sections marked with this icon describe features that have been added in version 2005 of the Visual Basic language or in version 2.0 of the .NET Framework. Notice that some features are so important that I devote an entire chapter to them, in which case this icon appears only at the top of the chapter. Otherwise, some really minor improvements are mentioned in text without being described in a section of their own.

Organization of This Book

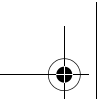
Programming Microsoft Visual Basic 2005: The Language is broadly organized in four parts.

The first three chapters of Part I cover the language basics. If you aren't new to Visual Basic, you might skip them, even though you might find some interesting tips here and there. If you are switching from Visual Basic 6 or Visual Basic .NET 2003, you can simply stop at the Visual Basic 6 and New icons, as described in the previous section. Regardless of your familiarity with Visual Basic, however, I recommend that you read Chapters 4 and 5 carefully because they explain the many new features of the Visual Studio IDE and illustrate concepts that are used in subsequent chapters.

Part II is devoted to object-oriented features of the Visual Basic language. Again, if you are already familiar with Visual Basic .NET, you might want to spend more time on the sections marked with the New icon, but I suggest you read Chapter 7, "Delegates and Events," and Chapter 9, "Object Lifetime," because they illustrate advanced techniques that can improve your skills remarkably. Chapter 11 is a must-read for learning more about the most intriguing and important new features of version 2.0 of the .NET Framework.

Part III is about basic types in the .NET Framework. Chapter 16, "The My Namespace," contains an in-depth description of this new Visual Basic feature, but you'll surely find a lot of useful information in all the chapters in this part. For example, Chapter 13, "Arrays and Collections," shows you how to work wonders with .NET Framework complex data structures (including generics collection). My favorite chapter is Chapter 14, "Regular Expressions," where I describe all I've learned about this exciting (and very useful) .NET Framework feature.

Part IV covers advanced programming topics, such as threading, serialization, PInvoke, and COM Interop. These features can make your applications more powerful, but failing to use them properly can introduce many hard-to-find bugs; thus, read these chapters carefully. Chapter 18, "Reflection," and Chapter 19, "Custom Attributes," are actually one very long chapter split into two: in the former, I offer a very complete reference on reflection, whereas in the latter I offer a few real-world (and quite complex) examples of the wonders custom attributes can do for you.



System Requirements

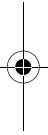
You need the following hardware and software to build and run the code samples for this book:

- Microsoft Windows XP with Service Pack 2, Microsoft Windows Server 2003 with Service Pack 1, or Microsoft Windows 2000 with Service Pack 4
- Microsoft Visual Studio 2005 Standard Edition or Microsoft Visual Studio 2005 Professional Edition. (A few sections in Chapters 4 and 5 assume that you have installed the Developer Edition of Visual Studio Team System.)
- 1-GHz Pentium or compatible processor
- 384 MB RAM (512 MB or more recommended)
- Video (800 × 600 or higher resolution) monitor with at least 256 colors (1,024 × 768 High Color 16-bit recommended)
- Microsoft Mouse or compatible pointing device

Technology Updates

As technologies related to this book are updated, links to additional information will be added to the Microsoft Press Technology Updates Web page. Visit this page periodically for updates on Visual Studio 2005 and other technologies:

<http://www.microsoft.com/mspress/updates/>



Code Samples

All of the code samples discussed in this book can be downloaded from the book's companion content page at the following address:

<http://www.microsoft.com/mspress/companion/0-7356-2183-7/>

Support for This Book

Every effort has been made to ensure the accuracy of the information in this book and the companion content. Microsoft Press provides support for books and companion content at the following Web site:

<http://www.microsoft.com/learning/support/books/>

I provide support for this book, including an errata page and updated code samples, at my Web site:

<http://www.dotnet2themax.com/>

and through my Weblog:

<http://www.dotnet2themax.com/blogs/fbalena>

Questions and Comments

If you have comments, questions, or ideas regarding the book or the companion content, or if you have questions that are not answered by visiting the preceding sites, please send them to Microsoft Press by e-mail:

mspinput@microsoft.com

Or by postal mail:

Microsoft Press
Attn: *Programming Microsoft Visual Basic 2005* Editor
One Microsoft Way
Redmond, WA 98052-6399

Please note that Microsoft software product support is not offered through the preceding addresses.

