

# Introduction

YOU'VE PROBABLY BEEN HEARING ABOUT Microsoft's .NET Framework and the new features of Visual Basic.NET. Perhaps you've read articles about it in magazines. Perhaps you've read promotional material from Microsoft. Perhaps you've even started experimenting with it and creating your own applications.

Regardless of how you've learned about it, you're probably feeling a bit overwhelmed. It's such a massive change both in language and approach that it's difficult to sort out the reality from the marketing and difficult to decide where one should actually start when approaching this new technology.

That's what this book is about.

- It's about the priorities you should use in learning .NET and the strategies you should use in deciding how and when to deploy .NET.
- It's about the concepts you need to know in order to understand the new features of Visual Basic.NET and how they will influence the way you write code under this new framework.
- And it's about the changes in the Visual Basic language itself.

This book is not intended to be a general survey of the framework or even a comprehensive reference to Visual Basic.NET—There are plenty of those books available and more on the way. Nor will you find here a rehash of the .NET documentation. The .NET Framework is so large and the changes to Visual Basic.NET so widespread that a comprehensive book would be huge and probably overwhelming (assuming it is even possible to fit one into a single volume). Entire subjects such as the development environment, debugging, and high-level services are deserving of books of their own and will be covered here barely or not at all.

This book is intended to be one of the first books you read on .NET—the one that will prepare you to understand the more comprehensive books, Microsoft's documentation, and the inevitable manual rehashes. It is intended to help you to think about this new technology in the context of your own development plans, and to get you up to speed quickly on concepts that will be new to most Visual Basic programmers. It is intended to be a concise and easy read yet provide enough depth to make you an expert on the VB.NET language and to lay the foundation for you to become an expert in those areas of the .NET Framework that most interest you.

## *Who Are You?*

I am assuming that you are an intermediate to advanced-level Visual Basic programmer. This book is emphatically *not* intended for people who are new to programming in general, though it should prove useful to non-Visual Basic programmers who are coming to VB.NET from other languages. It will also be valuable to team leaders and managers who are making deployment decisions—they’ll find that Part One—Strategies directly addresses management and technology issues.

Unlike many similar books, this one is focused entirely on helping current VB6 programmers come up to speed on .NET in general and VB.NET in particular. Hence, the organization is quite unusual. I waste no time explaining basic concepts such as “What is a class?” or “How does a For...Next loop work?” or “What is a collection?” Believe me, there is plenty to teach without rehashing the things that every VB6 programmer beyond the beginner level knows.

## *Regarding Microsoft*

The world seems to have recently divided itself into two camps: those who love and support Microsoft and those who hate and despise them. Personally, I fall into that somewhat unworldly group who refuses to see Microsoft as a major deity.<sup>1</sup> As a result, I sometimes find myself condemned by the non-Microsoft world as a “Microsoft supporter” and condemned by the Microsoft world as someone who “hates Microsoft.”

The truth is that I have a lot of respect for Microsoft.<sup>2</sup> They have a great deal of good technology, quite a bit of great technology, and a depressing amount of bloated technology. They also have a very effective marketing department whose actions I often find incomprehensible. I know enough about them to know that they are a chaotic organization—and their decisions are sometimes influenced as much by internal politics as technological sophistication (which is true of any company but somehow people think saying this of Microsoft is an insult).

So, I like them. I also like to make fun of them—partly because they make a great target but partly because it’s a natural and human reaction in dealing with an overwhelming force.<sup>3</sup>

For the record, I don’t hate them. I like them.

However, that doesn’t mean that I blindly agree with them or believe that they make the right choices on technology. I certainly don’t believe everything they write about “the best way for you to write software” and neither should you. Why?

- 
1. A minor deity perhaps. My point is that Microsoft has joined the ranks of keyboards, programmer’s editors, and operating systems as a “religious” issue—one on which I am a confirmed atheist.
  2. Some of my best friends work at Microsoft—it’s a cliché but true.
  3. And if you don’t think they’re an overwhelming force, just think for a moment about the impact .NET is going to have on your own career.

Because you know as well as I do that the best way for you to write software changes quickly—often in ways that Microsoft itself doesn't anticipate.

In this book, I will be sharing my opinions on the .NET Framework and the changes that have been made to the Visual Basic language. Some of those opinions will differ from what Microsoft says. Some will be controversial. Some will probably turn out to be wrong. But that's okay. My goal is not that you take everything I say as gospel, but that you think about the new technology in the context of your own situation—that you evaluate and judge it for yourself. I hope my perspective will broaden the way you look at Visual Basic.NET and ultimately help you in the process of adopting the technology to suit your own needs.

## *Regarding Prereleases*

If you've followed my prior books and articles, you probably know that there are two types of books I hate: manual rehashes and books that are rushed and inaccurate because they're based on prerelease code.

While this printing is based on the version 1 release of Visual Studio.NET, this book was originally based on prerelease code, which leads to three possibilities:

1. I have become a complete hypocrite.
2. I sold out for the money because an early book sells well even if it's crap.
3. I had some good reason for doing this book that isn't obvious but you're at least willing to hear my side of the story.

I know what you're thinking. The answer is three, right?

Of course. But it's also two<sup>4</sup>—except that I sincerely hope and believe that this book isn't crap.

I gambled that even though the .NET Framework was not released, the overall architecture and approach of the framework and its namespaces, as well as the Visual Basic .NET language itself, would be stable enough so that there would be few changes between beta 2 and release.

Fortunately, I won the bet. If you look at the revision lists for this book (included with your downloaded source code), you'll see that there have been miraculously few changes.

I took that gamble because I was and still am very concerned that people will approach these technologies the wrong way...

- That they'll condemn the language changes to VB.NET unfairly just because they break compatibility with existing code.

---

4. Sometimes it's painful to be honest.

- That they'll condemn the Common Language Runtime just because it's a huge runtime and VB.NET no longer has native code support.
- That they'll cheerfully start using inheritance and multithreading in ways that will get them into trouble—with code that is unscalable, unsupported, or hard to debug.

With this technology, more than any in recent memory, it is critical that you start out right—that you understand why Microsoft made the choices they did and why they are mostly good choices. You need to understand that inheritance is something you use only after a great deal of thought—almost as a last resort. You need to realize that consequences of a design mistake in a free-threaded environment can be incredibly expensive due to the difficulty of detecting obscure synchronization bugs.

That's the main reason that I wanted to get this book out early—because I want to help VB programmers get started off correctly. And I guess I'm confident enough in my own ability to think that what I have to say will help.

### *Regarding Source Code*

There are quite a few sample programs included with this book. You can download them from the publisher's Web site at <http://www.apress.com>. The source code is organized by chapter so if you see a reference to a project named `TwoInterfaces` in Chapter 5, look under the `Source\CH5\TwoInterfaces` directory.

My assumption is that you will use the source code provided from the Web site and not attempt to type in the code printed in the book. There are two reasons for this. First, in many cases, I have not included all of the code needed to run the project—just enough to explain the concept under discussion. That way you don't have to wade through endless printed listings. Second, this approach allows me to keep the code up to date and updated even as Microsoft continues to enhance Visual Studio .NET. Also, check out the book's Web site at <http://www.desaware.com> for lists of updates and corrections.

The code printed in the book was tested against the release build of Visual Studio .NET.

## *Regarding Desaware*

I always mention Desaware in my books. I do so because I am first and foremost a software developer. I write real code that ships and is currently running on untold thousands of systems. I helped cofound Apress in order to find a good home for my books (and if you are a software developer who is interested in writing, you'll find no better home—send me an email if you're interested).

But Desaware is the home for my software—primarily component software designed to help Visual Basic (and now Visual Studio) developers. At this time, I don't know what kinds of products we'll have for VB.NET—but you can bet we'll have some, so I invite you to visit <http://www.desaware.com> and see what we're up to. More important, you'll also find the updates and corrections page for this book on that site.

With that, I invite you join me in this introduction to Visual Basic.NET and the .NET Framework.

Dan Appleman

January 2002

[dan\\_appleman@apress.com](mailto:dan_appleman@apress.com)