



## Foreword by Francesco Balena

---

If Ted Pattison were a product, it would be a dream for any advertising agency. I mean, one of those products that sell by themselves: you don't really need to *promote* them, just let folks out there know that they're available. From this perspective, if the main goal for a foreword was to convince potential buyers to become actual readers, I'd say that my job is quite limited and that can be summarized in a sentence: *Want to learn Visual Basic .NET? Buy this book. Period.*

Still reading? Not convinced yet? Good, I have an excuse to talk a bit more about the book and its author, and throw in some personal memories.

I first met Ted at VBITS conferences, near the end of last millennium (not many years ago, but it sounds more impressive...), where both of us were delivering sessions and full-day seminars. As I've told him countless times, having to cope with his machine-gun-speed speech was just a bit too hard for my non-U.S. ears. However, I made a point not to miss his sessions, even if he talked on topics I was already familiar with. The reason was simple: Ted is a natural born teacher and he complements his natural talent with a rigorous approach to the material, right-on-focus code samples, and thorough explanations. Thus I always find it interesting *how* he delivers his material as much as *what* his talks are all about.

When I read *Programming Distributed Applications with COM and Visual Basic 6.0*, and its second edition that covers COM+ as well, I wasn't surprised to see that his teaching talent can fit the written page so nicely. That book found a place on the desk of many VB6 developers willing to leave the

**xviii ■ FOREWORD BY FRANCESCO BALENA**

safe harbors of RAD programming to face the challenges of enterprise-level applications.

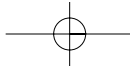
All Ted's talks and books, including the one you're reading now, have one thing in common: he isn't really interested in covering every single keyword or secondary detail about a language, a product, or a technology. Rather, he is more concerned in providing solid foundations about the basic concepts. This attitude is even more valuable now that Visual Basic has so many new features—inheritance, constructors, attributes, delegates, just to name a few—and most programmers are puzzled about when each one should be used, and how.

You didn't have to be familiar with object-oriented programming under previous versions of the language. In fact, you could create moderately large business applications without even coding a single class. In practice, VB6 classes were only a means to create COM components, but the language didn't really promote object-oriented design or programming. The rules have now changed, and you simply can't ignore the ins and outs of objects if you want to build robust, efficient, and scalable applications.

I am sure that this book will work wonderfully in helping developers in moving to the new world of Visual Basic .NET and the Microsoft .NET Framework. VB has become more powerful than ever, and you need the experience of a guy such as Ted Pattison to tame the new power and find your way in this object-oriented maze.

—Francesco Balena

Microsoft MSDN Regional Director for Italy  
Author of *Programming Microsoft Visual Basic .NET*  
Founder of VB-2-The-Max, [www.vb2themax.com](http://www.vb2themax.com)



## Foreword by Rocky Lhotka

---

The computer industry is cyclical. We oscillate between loving thin client terminal or browser-based interfaces and thick client interfaces. We've gone from procedural or modular designs to object-oriented designs and now we're returning to procedural designs in the form of Web services.

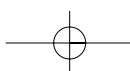
In the background, however, there's been a trend that doesn't seem so cyclical: component-based design. Whether we're doing a thin or intelligent client, we build our applications with components. Components are used when building procedural, object-oriented or Web service based applications.

Components are the common coin of the industry. Visual Basic and the venerable VBX components popularized the concept in the early '90s. The subsequent switch to COM and ActiveX components broadened the use of components from UI widgets to a wide variety of pre-built libraries.

Perhaps more importantly, Visual Basic 4 enabled us to build our own components. That was seven years ago, and since that time components have become pervasive. Not only in Visual Basic programming, but also in C++ and Java.

Today, virtually all applications are a composition of various components, working in concert to provide the desired functionality. We use components to create both Windows and web UI's, to manage middle-tier code and to access data. How do you create code to run in MTS or COM+? You create a component. How do you create a Web service? You create a component.

Microsoft's Component Object Model (COM) technology has been the most popular component-based technology over the past several years. COM, especially when programmed from Visual Basic, made component-



**xx ■ FOREWORD BY ROCKY LHOTKA**

based programming relatively easy. Not only could most developers use components, most developers could create them.

More recently, Microsoft has developed the .NET Framework. The .NET Framework is a powerful development platform. It incorporates the component-based features of COM, while overcoming COM's limitations, especially around deployment and versioning.

Perhaps more importantly, the .NET Framework merges component-based and object-oriented concepts into a seamless whole. In COM it was not possible to use inheritance between components, but in .NET it is common practice for us to inherit from a class in a different component. This is true even if that other component was written in a different programming language. So now, not only is Visual Basic .NET fully object-oriented, but the underlying development platform itself is object-oriented.

Of course with great power comes some complexity. That's where this book comes into play.

The key to successfully using the new capabilities of the .NET Framework and of Visual Basic .NET is in understanding what the capabilities are and how to implement them. In this book Ted has done an excellent job of walking through the features of Visual Basic .NET in a clear and methodical manner.

In going through the book, I learned a number of things, and had other key features reinforced. This book provides the foundational information that is required to successfully use Visual Basic .NET and the .NET Framework for application and component development.

As you move forward into the .NET Framework with Visual Basic .NET, you'll find the material in this book to be valuable. This is true for both Windows and Web developers, and for both data-centric and object-oriented developers. Whether you are using or creating components, you will find Visual Basic .NET to be a powerful and fun tool, and I think that this book will help you make the most of it.

—Rocky Lhotka  
Magenic Technologies