

PREFACE

Welcome to the latest in the series of *Doing Objects* books. This series started in 1995 with Visual Basic 4.0, when VB could first do objects. At that time, little had been discussed about using object-oriented design and development techniques with Visual Basic. To help improve that situation, almost half of each book in the series was dedicated to helping developers think in object-oriented terms and design an object-oriented application. The other half presented a full-featured solution for building an object-oriented application.

As with each of the other books in this series, this book is about designing and developing great applications for Visual Basic using object-oriented principles. However, today's Visual Basic developer is much more versed in design concepts, so only one chapter of this book is dedicated to design. The remaining chapters provide techniques for building an object-oriented application *and* a reusable application framework.

Unlike other books that provide reference-type information and short examples with no context, this book presents each topic in a logical sequence, illustrating the techniques by designing and developing a full-featured application. This allows you to readily see how each technique fits into your development activities.

This book also demonstrates how to build an application framework you can reuse in every application you build. With an application framework you can significantly increase your productivity and that of your team.

This new edition of the book was completely rewritten to use the features of Visual Basic 2005, Visual Studio 2005, and the .NET Framework. It makes no assumptions about your experience with .NET. It provides tips for experienced .NET developers and those new to .NET. There are even some tips specifically for developers moving from VB6 to .NET.

Why Do Objects?

Everything in .NET is basically an object. You cannot really do anything in .NET without doing objects. But Visual Basic 2005 still allows you to create modules, and it provides default form instances. Thus, you can still pretend to code without objects. So why make the move to using object-oriented techniques in your Visual Basic applications?

- Objects help you think about an application in terms of real-world things, aiding in the design process.
- Objects define all their data and processing in one programming unit. All of the code having to do with customers is in one code file, all of the code having to do with products is in another code file, and so on. This makes it much easier to develop, test, and maintain your code and to manage the complexities of software development. It also simplifies multiprogrammer development.
- Objects allow for building an application framework with base form classes and base business object classes. Common code resides in these base classes and not within every form or code file. This can drastically reduce the amount of code you have to write, test, and maintain.
- Objects allow you to work with other high-productivity features in .NET, such as object binding.

This book expands on these topics to help you know when, how, and why to do objects in Visual Basic 2005.

Who Should Read This Book?

The goal of most Visual Basic developers is to get the job done. You have existing code to maintain, enhancements to implement, and new applications to write—all with limited time and other resources. You need to be as productive as possible. Learning a new version of your programming language and development environment takes away from that productivity. So you need a way to learn the new tools and techniques as rapidly and efficiently as possible.

This book is for developers who need to quickly learn the key features of Visual Basic 2005, Visual Studio 2005, and the .NET Framework and

incorporate the new techniques into their daily design and development activities. Specifically, if you are interested in learning how an object-oriented approach and these new tools and techniques can minimize the complexity of software design and development and improve your productivity, this book is for you.

I have many opportunities to talk with software developers at conferences, via newsgroups and e-mail, and through my consulting company. The key concern for many developers is how to come up to speed quickly on the many new features. Bookstores have shelves full of books, but it is hard to know where to begin, how the features interrelate, or even how to find time to do that much reading. The goal of this book is to provide details on the key features in one place and all within the context of designing and building a full-featured application.

This book assumes you are familiar with some version of Visual Basic, even if it is only a classic version like VB3 or VB6. It assumes you know how to write programming logic and put controls on forms. This book is not for people new to programming. Nor is it for developers who have never used a visual programming tool, such as Visual Basic. After you learn the basics, come back to this book to learn best practices for building great Visual Basic 2005 applications.

About This Book

This book begins with an overview of object-oriented terms and techniques. It then covers a pragmatic approach to application design, including the design of an application framework. The majority of the book details how to use the tools and techniques in Visual Basic, Visual Studio, and the .NET Framework to build great object-oriented applications.

This book is organized as follows:

- **Chapter 1, “Introduction to OO in .NET”:** Basic object-oriented terminology and concepts are covered in this first chapter, including specifics on how these concepts are realized in Visual Basic. It also provides an in-depth discussion about the benefits of an object-oriented approach.
- **Chapter 2, “Designing Software”:** When designing an application, you have many design aspects to consider, such as user interface, business objects, and database. Managing all these aspects is

easier if you follow a design methodology. This chapter presents the GUIDS Methodology, a pragmatic approach to object-oriented design.

- **Chapter 3, “Building Projects”:** This chapter covers the basics of building a solution with associated projects for the user interface, business logic, and data access layers. It includes a discussion of productivity enhancers such as templates and using the built-in Windows application framework.
- **Chapter 4, “Building the User Interface Layer”:** The user interface of a .NET application is composed of two parts. The outside is the part of the application that the user sees and interacts with. The inside is the code behind the forms. This chapter shows you how to build both parts with reuse and changeability in mind. It also introduces base form classes and programmatic interfaces.
- **Chapter 5, “Building the Business Logic Layer”:** The business logic layer includes all of the functionality of the application. This chapter details how to build classes with appropriate properties and methods. It also introduces base business object classes and generics.
- **Chapter 6, “Class Tools and Techniques”:** Visual Studio has many fun tools to help you build your application. This chapter covers the Class Designer, Object Test Bench, code snippets, and unit tests. It also covers more advanced techniques such as building master/detail classes.
- **Chapter 7, “Binding the User Interface to the Business Objects”:** Visual Studio provides enhanced features for binding your user interface elements to business object properties. This chapter describes binding and details how to perform object binding. It also covers how to write code to validate user-entered values.
- **Chapter 8, “Building the Data Access Layer”:** Most applications require some type of data. This chapter details how to build a database. It then covers how to build a data access component that uses ADO.NET to retrieve and save data in a database.

“Building Along” Activities

It is often easier to learn by doing. So this book provides “building along” activities in each development chapter. You can build along with these activities to construct the sample application.

By the time you reach the end of the book, you will have a full-featured Windows application that demonstrates many of the techniques presented in this book. You will also have the basics of an application framework that you can reuse in all your applications.

In addition, each chapter includes a “Try It!” section. If you don’t have time to build the entire sample application, you can download the sample code and extend several features using the “Try It!” sections. This provides a facility for you to practice what you are reading without the commitment of building the entire sample application. Note, that many of the “Try It!” exercises build on prior “Try It!” exercises to demonstrate how the techniques build on one another and work together to form a complete application.

What You Need to Use This Book

To work through any of the techniques, “building along” activities, or “Try It!” section exercises, you must have some edition of Visual Basic 2005. You can download the free Visual Basic 2005 Express Edition or use any other edition of these tools. In some cases the Express Edition does not provide a feature discussed in this book. These cases are clearly indicated in the text.

When you launch Visual Studio 2005 for the first time, you are asked to select your default development settings. The settings you select affect the names of many of the menu options, the layout of some of the dialogs, and other features of Visual Studio. This book uses the General Development settings, because they are the most common settings used by .NET developers.

NOTE: If you have your Visual Studio set to use the Visual Basic Development settings instead of the General Development settings, many of your Visual Studio options and some of the dialogs will be different from those in this book. You can continue to use your settings and work with the differences, or you can change your settings.

To set your Visual Studio settings so that your environment matches the one used in this book, select **Tools | Import and Export Settings**. Select **Reset All Settings** and click **Next**. Choose whether to save your current settings, and click **Next**. Select **General Development Settings**, and click **Finish**.

Although you could use any database product, to work through all the techniques in Chapter 8 you should also have access to a version of SQL Server. You can use SQL Server 2000 or any edition of SQL Server 2005. If you don't have SQL Server, you can download the free SQL Server 2005 Express Edition. If you want to use Microsoft Access instead, notes are provided in Chapter 8 about the features of Visual Studio 2005 that do not work with Access.

Downloading the Code

You can build all of the code for the sample application from instructions in this book by following the “building along” activities. However, you may not have time to work through every example. If you want to download the code, you can find it at <http://www.insteptech.com>.

Errata

I have made every effort to describe the concepts presented in this book in a clear and concise fashion. I have tried to ensure that the book is up to date as of this writing. And I have worked with the technical reviewers and technical editors to confirm that there are no errors in the text or code. However, mistakes are possible, and improvements are always welcome.

If you have suggestions for improving the content of the book, or if you find something that is incorrect or unclear, I would like to hear from you. I can then incorporate your comments in future editions of this book. You can reach me via e-mail at deborahk@insteptech.com.