Preface

Welcome to Visual Basic .NET and the world of Windows, Internet and World-Wide-Web programming with Visual Studio and the .NET platform! This book, the first in our new *Simply* series, was a joy to create. Our goal was to write a book that focuses on core concepts and features of Visual Basic .NET while keeping the discussion of this highly technical subject as simple as possible.

To achieve these goals, we implemented an innovative teaching methodology. We present the core concepts of leading-edge computing technologies using the tutorial-based, Application-Driven™ approach, combined with the Deitel™ signature Live-Code™ approach of teaching programming using complete, working, real-world applications. We merged the notion of a lab manual with that of a conventional textbook, creating a book that is best used by students sitting at computers and building each example application as they read the tutorials.

As students work through the tutorials, they learn about Visual Basic .NET and its fundamental features, such as visual programming concepts, graphical-user-interface (GUI) components, multimedia (audio, images, animation and video), file processing, database processing and Internet and World-Wide-Web-based client/server networking. At the end of most sections, we provide self-review questions with answers so that students receive immediate feedback on their understanding of the material. Hundreds of additional self-review questions with answers are available on this book's Companion Web Site.

Features in Simply Visual Basic .NET

This book is loaded with pedagogic features, including:

- ***Application-Driven™ Tutorial Approach.*** Each tutorial uses a contemporary, real-world application to teach programming concepts. The examples and exercises are up-to-the-minute with Internet/Web-related examples and with popular applications, such as ATMs, game playing, graphics, multimedia and even a 3-tier Web-based bookstore. Most examples have a business focus. At the beginning of each tutorial, students "test-drive" the completed application so they can see how it works. Then they build the application by following our step-by-step instructions. The book concentrates on the principles of good software engineering and stresses program clarity.
- ***Live-Code™ Approach.*** This book is loaded with Live-Code™ examples. Each tutorial ends with the complete, working program code and the students can run the application that they just created. We call this method of teaching and writing the ***Live-Code™ Approach***. We feel that this approach is more effective than presenting only snippets of code out of the context of a complete program.
- ***Real-World Technologies.*** This text incorporates today's technologies to develop useful applications. For example, we use the Unified Modeling Language™ (UML) to replace flowcharts—an older standard. The UML has become the preferred graphical modeling language for designing object-ori-

ented applications. In *Simply Visual Basic .NET*, we use UML to show the flow of control for several applications, so students gain practice reading the type of diagrams that are used in industry.

- ***Visual Programming and Graphical User Interface (GUI).*** From the first tutorial, we immerse students in visual programming techniques and modifying Visual Basic .NET GUIs. Students who learn these techniques can create graphical programs more quickly and easily. The early tutorials provide students with a foundation for designing GUIs—concepts that they will apply throughout the book as we teach core programming concepts. Many tutorials contain GUI Design Tips that are summarized at the end of the tutorials for easy reference. Additionally, Appendix C compiles all the GUI Design Tips to help students as they prepare for exams.

- ***Full-Color Presentation.*** This book is in full color so that students can see sample outputs as they would appear on a monitor. Also, we syntax color the Visual Basic .NET code, similar to the way Visual Studio .NET colors the code in its editor window. This way, students can match what they see in the book with what they see on their own screens. Our syntax-coloring conventions are as follows:

```
comments appear in green
keywords appear in dark blue
literal values appear in light blue
text, class, method and variable names appear in black
errors appear in red
```

- ***Graphics and Multimedia.*** Graphics make applications fun to create and use. In our introduction to graphics, Tutorial 26, we discuss Graphical Device Interface (GDI+)—the Windows service that provides the graphical features used by .NET—to teach students to personalize a bank check. In Tutorial 27, we use a fun technology called Microsoft Agent to add interactive, animated characters to a phone book application. With Microsoft Agent, your applications can speak to users and even respond to their voice commands!

- ***Databases.*** Databases are crucial to businesses today, and we use real-world applications to teach the fundamentals of database programming. Tutorials 25 and 30 familiarize students with databases, presented in the context of two applications—an ATM and a Web-based bookstore.

- ***Case Study.*** This book concludes with a sequence of four tutorials in which the student builds a Web-based, bookstore application. Tutorial 28 familiarizes readers with Microsoft's Internet Information Services (which enables Web publishing), multi-tier architecture and simple Web transactions. Tutorials 29–31 use ASP .NET and ADO .NET to build an application that retrieves information from a database and displays the information in a Web page.

- ***Object-Oriented Programming.*** Object-oriented programming is the most widely employed technique for developing robust, reusable software, and Visual Basic .NET offers advanced object-oriented programming features. This book introduces students to defining classes and using objects, laying a solid foundation for future programming courses.

- ***Visual Studio .NET Debugger.*** Debuggers are programs that help programmers find and correct logic errors in program code. Visual Studio .NET contains a powerful debugging tool that allows programmers to analyze their programs line-by-line as the programs execute. Throughout the book, we teach the Visual Studio .NET Debugger; we explain how to use its key features and offer many debugging exercises.

To the Instructor

Focus of the Book

Our goal was clear: Produce a Visual Basic .NET textbook for introductory-level courses in computer programming aimed at students with little or no programming experience. This book teaches computer programming principles and the Visual Basic .NET language, including data types, control structures, object-oriented programming, Visual Basic .NET classes, GUI concepts, event-driven programming and more. After mastering the material in this book, students will be able to program in Visual Basic .NET and to employ many key capabilities of the .NET platform.

Lab Setup

To install some of the required software for this book, students and instructors will need Administrator-level access to the computer. For university computer labs where students do not have Administrator-level access, instructors and system administrators must ensure that the proper software is installed on the lab computers. In Tutorial 27, certain Microsoft Agent software components must be installed to execute and develop the **Phone Book** application. If students are not allowed to install software on lab computers, the Microsoft Agent components discussed in Tutorial 27 must be installed in advance. To configure and execute some of the examples and exercises, such as the **Bookstore** case study in Tutorials 28–31, students will need to have Administrator-level access. In addition, the examples in the book require that students have Debugger or Administrator access to the computer, which is typically required to develop applications in Visual Studio .NET.

A Note Regarding Terminology Used in the Book

In Tutorial 13, we discuss methods as `Sub` procedures (sometimes called subroutines) and `Function` procedures (sometimes called functions). We use this terminology for two reasons. First, the keywords `Sub` and `Function` are used in procedure definitions, so this naming is logical for students. Second, Visual Basic professionals have used this terminology for years and will continue to do so in the future. We also use the term “function” at certain points in this text to refer to Visual Basic 6 `Function` procedures that remain in Visual Basic .NET (such as `Val` and `Pmt`). When we introduce object-oriented programming concepts in Tutorial 19, we discuss the difference between procedures and methods and indicate that the procedures defined throughout the text are, in fact, methods. We hope our use of terminology helps you present the material in a simple and understandable manner.

Exception Handling: Bonus Tutorial Available Online

Exception Handling is one of the most important topics in Visual Basic .NET for building mission-critical and business-critical applications. Programmers need to know how to recognize the exceptions (errors) that could occur in software components and handle those exceptions effectively, allowing programs to deal with problems and continue executing instead of “crashing.” This tutorial overviews the proper use of exception handling, including the termination model of exception handling, throwing and catching exceptions and the library class `Exception`. To download this tutorial, please go to `www.deitel.com/vbnetSIMPLY1/index.html`.

Objectives

Each tutorial begins with objectives that inform students of what to expect and give them an opportunity, after reading the tutorial, to determine whether they have met the intended goals.

Outline

The tutorial outline enables students to approach the material in top-down fashion. Along with the tutorial objectives, the outline helps students anticipate future topics and set a comfortable and effective learning pace.

Example Programs (with Program Outputs)

We present Visual Basic .NET features in the context of complete, working Visual Basic .NET programs. We call this our LIVE-CODE™ approach. All examples are available on the CD that accompanies the book or as downloads from our Web site, `www.deitel.com/vbnetSIMPLY1/index.html`.

Illustrations/Figures

An abundance of charts, line drawings and program outputs are included. The discussion of control structures, for example, features carefully drawn UML activity diagrams. [*Note:* We do not teach UML diagramming as a program-development tool, but we do use UML diagrams to explain the precise operation of many of Visual Basic .NET's control structures.]

Programming Tips

Hundreds of programming tips to help students focus on important aspects of program development. These tips and practices represent the best the authors have gleaned from a combined seven decades of programming and teaching experience.

Good Programming Practices

Good Programming Practices highlight techniques that help students write programs that are clearer, more understandable and more maintainable.

Common Programming Errors

Students learning a language—especially in their first programming course—frequently make errors. Pointing out these *Common Programming Errors* in the text reduces the likelihood that students will make the same mistakes.

Error Prevention Tips

These tips describe aspects of Visual Basic .NET that prevent errors from getting into programs in the first place, which simplifies the testing and debugging process.

Performance Tips

Teaching students to write clear and understandable programs is the most important goal for a first programming course. But students want to write programs that run the fastest, use the least memory, require the smallest number of keystrokes, etc. *Performance Tips* highlight opportunities for improving program performance.

Portability Tips

The *Portability Tips* provide insights on how Visual Basic .NET achieves its high degree of portability among .NET platforms.

Software Design Tips

The *Software Design Tips* highlight architectural and design issues that affect the construction of object-oriented software systems.

GUI Design Tips

The *GUI Design Tips* highlight graphical-user-interface conventions to help students design attractive, user-friendly GUIs and use GUI features.

Skills Summary

Each tutorial includes a bullet-list-style summary of the new programming concepts presented. This reinforces key actions taken to build the application in each tutorial.

Key Terms

Each tutorial includes a list of important terms defined in the tutorial. These terms also appear in the index and in a book-wide glossary, so the student can locate terms and their definitions quickly.

230 Self-Review Questions and Answers

Self-review multiple-choice questions and answers are included after most sections to build students' confidence with the material and prepare them for the regular exercises. Students should be encouraged to attempt all the self-review exercises and check their answers.

834 Exercises (Solutions in Instructor's Manual)

Each tutorial concludes with exercises. Typical exercises include 10 multiple-choice questions, a "What does this code do?" exercise, a "What's wrong with this code?" exercise and a programming challenge. The questions involve simple recall of important terminology and concepts, writing individual Visual Basic .NET statements, writing small portions of Visual Basic .NET applications and writing complete Visual Basic .NET methods, classes and applications. Every programming exercise uses a step-by-step methodology to suggest how to solve the problems. The solutions for the exercises are *available only to instructors* through their Prentice-Hall representatives. **[*NOTE:* Please do not write to us requesting the instructor's manual. Distribution of this publication is strictly limited to instructors teaching from the book. Instructors may obtain the solutions manual only from their regular Prentice Hall representatives. We regret that we cannot provide the solutions to professionals.]**

GUI Design Guidelines

Consistent and proper graphical user interface design is crucial to visual programming. In each tutorial, we summarize the GUI design guidelines that were introduced. Appendix C presents a cumulative list of these GUI design guidelines for easy reference.

Controls, Events, Properties & Methods Summaries

Each tutorial includes a summary of the controls, events, properties and methods covered in the tutorial. The summary includes a picture of each control, shows the control "in action" and lists the control's properties, events and methods that were discussed up to and including that tutorial. In addition, Appendix E groups the controls by tutorial for easy reference.

Index

The extensive index includes important terms both under main headings and as separate entries so that students can search for any term or concept by keyword. The code examples and the exercises also are included in the index. For every Visual Basic .NET source-code program in the book, we indexed it both under the appropriate application and as a subindex item under "code examples." We have also double-indexed features such as controls and properties. This makes it easier to find examples using particular features.

Simply Visual Basic .NET Ancillary Package

Simply Visual Basic .NET is accompanied by extensive ancillary materials for instructors, including the following:

- *Instructor's Resource CD (IRCD)* which contains the
    - *Instructor's Manual* with solutions to the end-of-tutorial exercises and
    - *Test-Item File* of multiple-choice questions (approximately two per tutorial section).
- *Customizable PowerPoint® Slides* containing all the code and figures in the text, and bulleted items that summarize the key points in the text. The slides are downloadable from `www.deitel.com/vbnetSIMPLY1/index.html` and are available as part of Prentice Hall's *Companion Web Site* (`www.prenhall.com/deitel`) for *Simply Visual Basic .NET*, which offers resources for both instructors and students.

Companion Web Site

For instructors, the *Companion Web Site* offers a *Syllabus Manager*, which helps instructors plan courses interactively and create online syllabi. Students also benefit from the functionality of the *Companion Web Site*. Book-specific resources for students include:

- PowerPoint® slides
- Example source code
- Reference materials from the book appendices
- Tutorial objectives
- Tutorial summaries
- Tutorial outlines
- Programming tips from each tutorial
- Online Study Guide—contains additional short-answer self-review exercises with answers
- Students can track their results and course performance on quizzes using the *Student Profile* feature, which records and manages all feedback and results from tests taken on the *Companion Web Site*. To access the *Companion Web Site* for *Simply Visual Basic .NET*, visit `www.prenhall.com/deitel`.

Simply Visual Basic .NET Multimedia Cyber Classroom

For the second edition of *Simply Visual Basic .NET*, we are planning to include the *Simply Visual Basic .NET: An* A*PPLICATION*-D*RIVEN*™*, Tutorial Approach Multimedia Cyber Classroom* interactive CD. Our *Cyber Classrooms* are loaded with e-Learning features that are ideal for both learning and reference.

Each *Cyber Classroom* CD provides an introduction in which the authors overview the *Cyber Classroom*'s features. The textbook's LIVE-CODE™ examples truly "come alive" in the *Cyber Classrooms*. If you are viewing a program and want to execute it, you just click the lightning-bolt icon, and the program will run. You will see—and hear, when working with audio-based multimedia programs—the program's output immediately. If you want to modify a program and see the effects of your changes, click the floppy-disk icon that causes the source code to be "lifted off" the CD and "dropped into" one of your own directories so you can edit the text, recompile the program and try out your new version. Click the audio icon, and one of the authors will discuss the program and "walk you through" the code.

The *Cyber Classrooms* also provide navigational aids, including extensive hyperlinking. The *Cyber Classrooms* are browser based, so they remember sections that you have visited recently and allows you to move forward or backward among them. The thousands of index entries are hyperlinked to their text occurrences. Furthermore, when you key in a term using the "find" feature, the *Cyber Classrooms* locate occurrences of that term throughout the text. The Table of Contents entries are "hot," so clicking a tutorial name takes you immediately to that tutorial.

Course Management Systems

Selected content from *Simply Visual Basic .NET* and other Deitel texts, is available to integrate into various Course Management Systems, including CourseCompass, Blackboard and WebCT.[1] Course Management Systems help faculty create, manage and use sophisticated Web-based educational tools and programs. Blackboard, CourseCompass and WebCT offer:

- Features to create and customize an online course
- Communication tools
- Flexible testing tools
- Support materials

In addition to the tools found in Blackboard and WebCT, CourseCompass from Prentice Hall includes:

- **CourseCompass course home page**, which makes the course as easy to navigate as a book.
- **Hosting on Prentice Hall's centralized servers**, which allows course administrators to avoid separate licensing fees or server-space issues.
- **"How Do I" online-support sections** are available for users who need help personalizing course sites.
- **Instructor Quick Start Guide**

Premium Course Management Systems

For the second edition of *Simply Visual Basic .NET*, we are planning support for Premium Course Management Systems that integrate content from a rich variety of sources, including Deitel books, *Cyber Classrooms* and *Companion Web Sites* with Course Management courseware—providing enhanced content to users. Premium Course Management Systems include:

- **Pre-Loaded DEITEL™ Content in a Customizable Interface**. An instructor can aggregate and customize all course materials. This feature includes the e-Book, a searchable digital version of the book, and downloadable PowerPoint® slides.
- **All the Interactivity of the *Cyber Classroom*.** Students can work with code and receive the added benefit of several hours of detailed audio descriptions of thousands of lines of code to help reinforce concepts. Every code example from the books is included.
- **Abundant Self-Assessment and Complete *Test-Item File*.** Use or edit hundreds of pre-loaded assessments, or upload your own. Assessments include self-review exercises, programming exercises (half with answers included) and test questions. Instructors choose which questions to assign, and students receive immediate feedback. Instructors can collect students' work and track their progress in an online gradebook.

To view free online demonstrations and learn more about Course Management Systems that support Deitel content, visit the following Web sites:

- Blackboard: `www.blackboard.com` and `www.prenhall.com/blackboard`.
- WebCT: `www.webct.com` and `www.prenhall.com/webct`.
- CourseCompass: `www.coursecompass.com` and `www.prenhall.com/coursecompass`.

[1] The entire text of *Simply Visual Basic .NET* will be provided in the e-Book included with Premium CourseCompass for the second edition of the book.

For Students and Instructors: Important Information Before You Begin

Please follow the instructions in this section to ensure that you have the proper setup before you begin this book.

Font and Naming Conventions

We use fonts to distinguish between IDE features (such as menu names and menu items) and other elements that appear in the IDE. Our convention is to emphasize IDE features in a sans-serif bold **Helvetica** font (for example, **Properties** window) and to emphasize program text in a serif `Lucida` font (for example, `Dim x As Boolean`). We use *italics* to emphasize special terms that are associated with Visual Studio .NET features, such as *Intellisense* and *Parameter Info*.

Software Bundled with *Simply Visual Basic .NET*

In college bookstores, this book is shrink-wrapped with a trial edition of the Microsoft® Visual Studio® .NET integrated development environment. This is a full and complete version of Visual Studio .NET which may be used without charge for 60 days, at which time it becomes inoperable. No technical support accompanies this edition.

Students enrolled in a course at an accredited school who have purchased this book for use in conjunction with that course may purchase Microsoft Visual Basic .NET Standard Edition from Microsoft for $15.00 (US) plus shipping and handling. The standard edition is fully functional and is shipped on 5 CDs. There is no time limit and technical support is available upon registration of the software after installation.

To purchase this software, you must obtain a purchase code from your instructor. This code is supplied to your instructor by Prentice Hall. To order Visual Basic .NET Standard Edition, visit

`www.gotdotnet.com/community/student/reference/purchase.aspx`

and follow the instructions. A credit card is required to complete the purchase.

Professionals can obtain the 60-day trial edition of Visual Studio .NET from `www.msdn.microsoft.com/vstudio/productinfo/trial/default.asp`.

Hardware and Software Requirements to Run Visual Basic .NET

To install and run Visual Basic .NET, Microsoft recommends that PCs have these minimum requirements:

- Pentium II 450 MHz processor
- Windows NT 4.0 with Service Pack 6 or later operating system
- 64 megabytes for a Windows NT 4.0 computer; 96 MB for Windows 2000 Professional and 160 MB for Windows XP Professional
- 2.5 gigabytes of available hard drive space
- CD-ROM (DVD for the trial version)
- Super VGA monitor with 256 colors
- Mouse or other Microsoft-compatible pointing device
- **You must install Microsoft's Internet Information Services (IIS) before installing Visual Studio .NET.** Otherwise, the Web-based bookstore application in the case study cannot be created or executed. See Appendix G for detailed instructions on installing IIS.

This book assumes that you are using Windows 2000 or Windows XP, plus Microsoft's Internet Information Services (IIS). Additional setup instructions for

Web servers and other software is available on our Web site along with the examples. [*Note:* This is copyrighted material. Feel free to use it as you study, but you may not republish any portion of it in any form without explicit permission from Prentice Hall and the authors.]

Monitor Display Settings

Simply Visual Basic .NET includes hundreds of screenshots of applications. Your monitor-display settings may need to be adjusted so that the screenshots in the book will match what you see on your computer screen as you develop each application. [*Note:* We refer to single-clicking with the left mouse button as **selecting**, or **clicking.** We refer to double-clicking with the left mouse button simply as **double clicking**.] Follow these steps to set your monitor display correctly:

1. Open the **Control Panel** and double click **Display**.
2. Click the **Settings** tab.
3. Click the **Advanced...** button.
4. In the **General** tab, make sure **Small Fonts** is selected; this should indicate that **96 dpi** is now the setting (if you already have this setting, you do not need to do anything else).
5. Click **Apply**.

If you choose to use different settings, the `Size` and `Location` values we provide for different GUI elements (such as `Button`s and `Label`s) in each application might not appear correctly on your screen. If so, simply adjust `Size` and `Location` values so the GUI elements in your application appear similar to those in the screenshots in the book.

Theme Settings for Windows XP Users

If you are using Windows XP, we assume that your theme is set to Windows Classic Style. Follow these steps to set Windows XP to display the Windows Classic theme:

1. Open the **Control Panel** and double click **Display**.
2. Click the **Themes** tab. Select **Windows Classic** from the **Theme:** drop-down list.
3. Click **OK** to save the settings.

Copying and Organizing Files

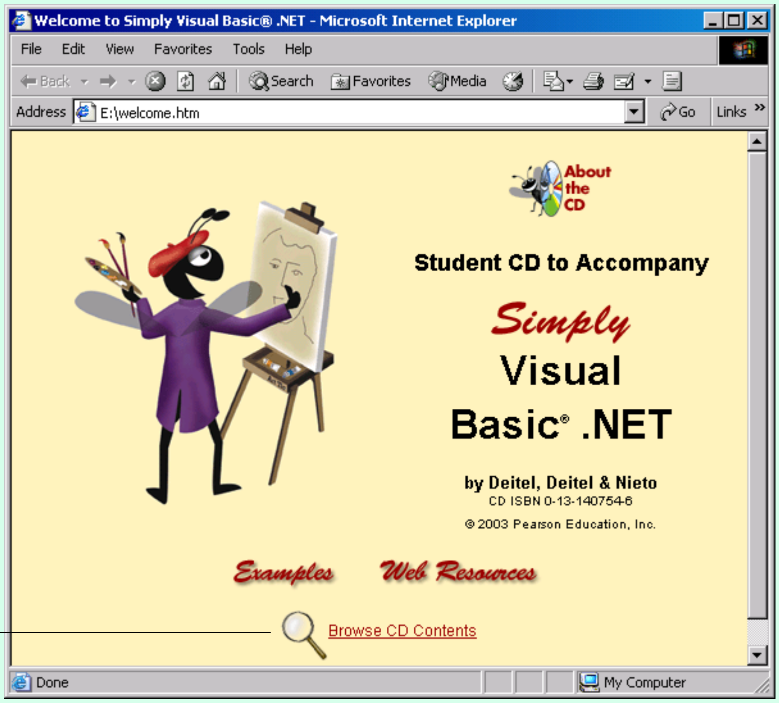
All of the examples for *Simply Visual Basic .NET* are included on the CD-ROM that accompanies this textbook. Follow the steps in the box, *Copying the Book Examples from the CD-ROM*, to copy the examples folder from the CD-ROM onto your hard drive. We suggest that you work from your hard drive rather than your CD drive for two reasons: you cannot save your programs to the book's CD (the CD is read-only), and files can be accessed faster from a hard drive than from a CD. The examples from the book (and our other publications) are also available on the Internet as downloads from the following Web sites:

```
www.deitel.com
www.prenhall.com/deitel
```

Screen shots in the box, *Copying the Book Examples from the CD-ROM*, might differ slightly from what you see on your computer, depending on whether you are using Windows 2000 or Windows XP. We used Windows 2000 to prepare the screenshots for this book.

Copying the Book Examples from the CD-ROM

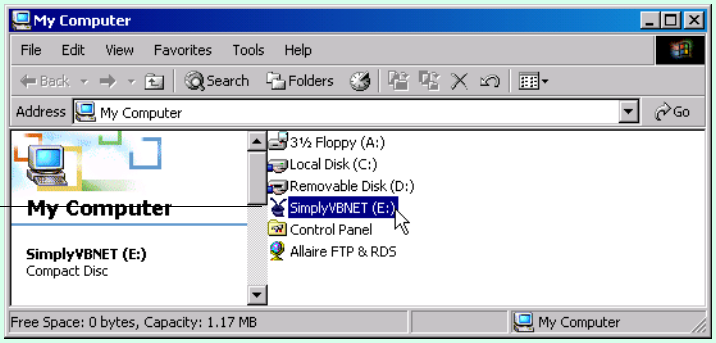
1. ***Locating the CD-ROM drive.*** Insert the CD that accompanies *Simply Visual Basic .NET* into your computer's CD-ROM drive. The window displayed in Fig. 1 should appear. If the page appears, proceed to *Step 3* of this box. If the page does not appear, proceed to *Step 2*.


Welcome to Simply Visual Basic® .NET - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address E:\welcome.htm
Go Links
About the CD
Student CD to Accompany
Simply
Visual
Basic® .NET
by Deitel, Deitel & Nieto
CD ISBN 0-13-140754-6
© 2003 Pearson Education, Inc.
Examples
Web Resources
Browse CD Contents
Done
My Computer

Click the **Browse CD Contents** link to access the CD's contents

Figure 1 Welcome page for Simply Visual Basic .NET CD.

2. ***Opening the CD-ROM directory using My Computer.*** If the page shown in Fig. 1 does not appear, double click the **My Computer** icon on your desktop. In the **My Computer** window, double click your CD-ROM drive (Fig. 2) to access the CD's contents. Proceed to *Step 4*.



Selected CD-ROM drive

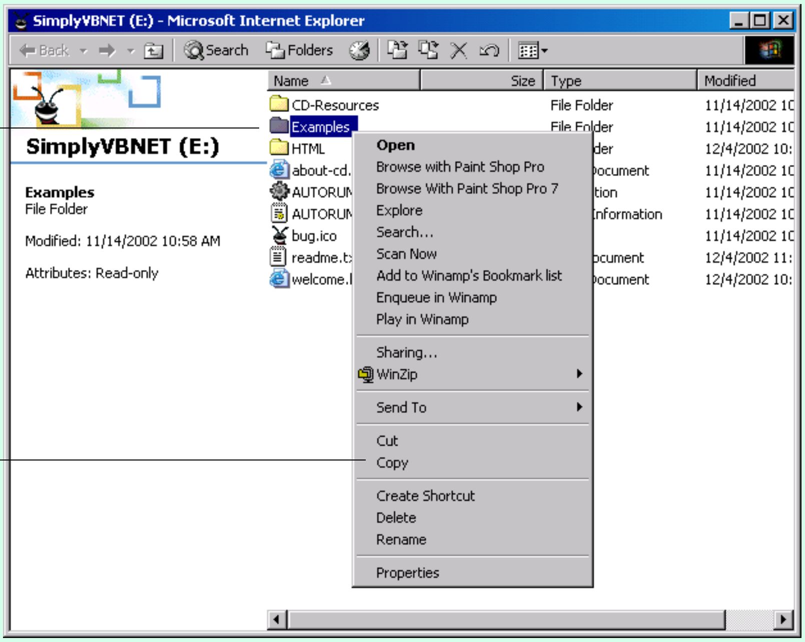
Figure 2 Locating the CD-ROM drive.

3. ***Opening the CD-ROM directory.*** If the page in Fig. 1 does appear, click the **Browse CD Contents** link to access the CD's contents.

4. ***Copying the `Examples` folder.*** Right click the `Examples` folder (Fig. 3), then select **Copy**. Next, go to **My Computer** and double click the `C:` drive. Select the **Edit** menu and select **Paste** to copy the folder and its contents from the CD to your `C:` drive.

(cont.)

[*Note*: We save the examples to the `C:` drive and refer to this drive throughout the text. You may choose to save your files to a different drive based on your lab setup or personal preferences. Please see your instructor for more information if you are working in a computer lab to confirm where the examples should be saved.]



Examples
File Folder

Modified: 11/14/2002 10:58 AM

Attributes: Read-only

Right clicking the `Examples` folder

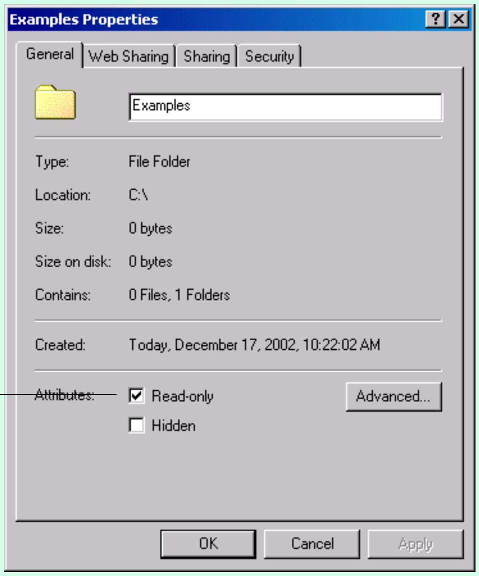
Select **Copy**

Figure 3 Copying the `Examples` folder.

The book example files you copied onto your computer from the CD are read-only. To access and modify these files, you must change this property. In the box, *Changing the Read-Only Property of Files*, you change the read-only property so you can run and modify the examples.

Changing the Read-Only Property of Files

1. ***Opening the Properties dialog.*** Right click the `Examples` folder and select **Properties** from the menu. The **Examples Properties** dialog appears (Fig. 4).



Examples

Type: File Folder

Location: C:\

Size: 0 bytes

Size on disk: 0 bytes

Contains: 0 Files, 1 Folders

Created: Today, December 17, 2002, 10:22:02 AM

Attributes: ☑ Read-only

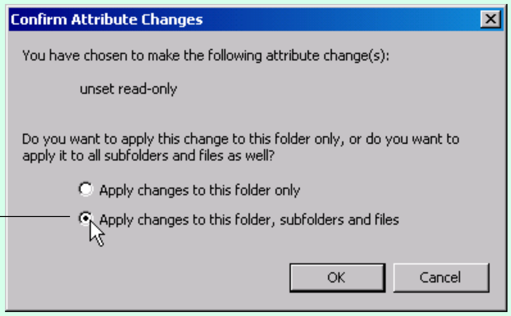
□ Hidden

Uncheck the Read-only attribute

Figure 4 Removing the check in the **Read-only** check box.

(cont.)

2. ***Changing the read-only property.*** In the **Attributes** section of this dialog, click the box next to **Read-only** to remove the check mark. Click **Apply** to apply the changes.
3. ***Changing the property for all files.*** Clicking **Apply** will display the **Confirm Attribute Changes** window (Fig. 5). In this window, click the radio button next to **Apply changes to this folder, subfolders and files** and click **OK** to remove the read-only property for all of the files and folders in the **Examples** folder.



Confirm Attribute Changes

You have chosen to make the following attribute change(s):

unset read-only

Do you want to apply this change to this folder only, or do you want to apply it to all subfolders and files as well?

○ Apply changes to this folder only

● Apply changes to this folder, subfolders and files

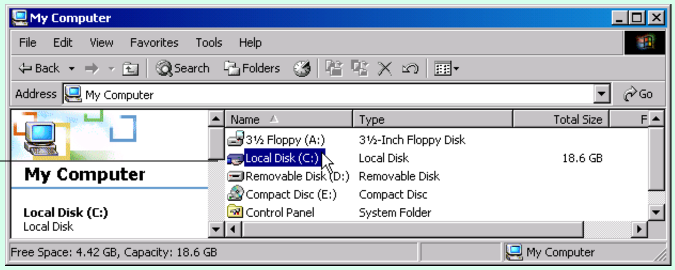
Click this radio button to remove the read-only property for all the files

Figure 5 Removing read-only for all the files in the **Examples** folder.

As you work through this book, you will be developing your own applications. In the box, *Creating a Work Folder*, you create a folder on your `C:` drive in which you will save all of your applications.

Creating a Work Folder

1. ***Selecting the drive.*** Double click the **My Computer** icon on your desktop to access a list of your computer drives (Fig. 6). Double click the `C:` drive. The contents of the `C:` drive are displayed.



Address: My Computer

My Computer

Local Disk (C:)
Local Disk

Name	Type	Total Size	F
3½ Floppy (A:)	3½-Inch Floppy Disk		
Local Disk (C:)	Local Disk	18.6 GB	
Removable Disk (D:)	Removable Disk		
Compact Disc (E:)	Compact Disc		
Control Panel	System Folder		

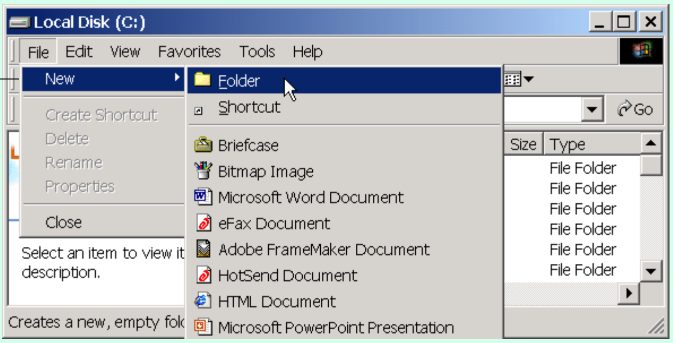
Free Space: 4.42 GB, Capacity: 18.6 GB   My Computer

Local disk

Figure 6 Computer drives listed under **My Computer**.

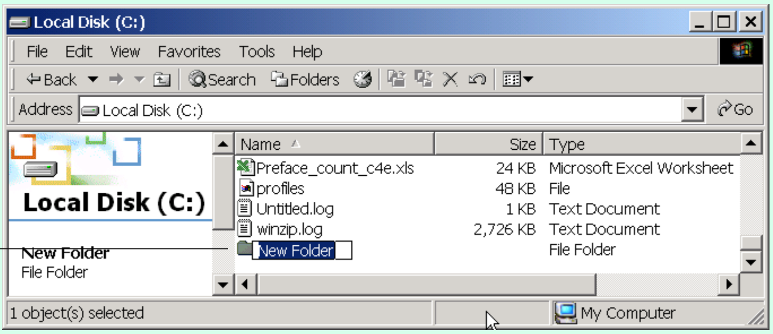
2. ***Creating a new folder.*** Select the **File** menu and under the **New** submenu, select **Folder** (Fig. 7). A new, empty folder appears in your `C:` directory (Fig. 8). [*Note:* From this point onward, we use the **>** character to indicate the selection of a menu command. For example, we use the notation **File > Open** to indicate the selection of the **Open** command from the **File** menu.]
3. ***Naming the folder.*** Enter a name for the folder. We suggest that you choose a name that you recognize and remember. We chose `SimplyVB` (Fig. 9). You can use this folder to save the examples from this book, your applications and your exercise solutions.

(cont.)



New folder option (selected)

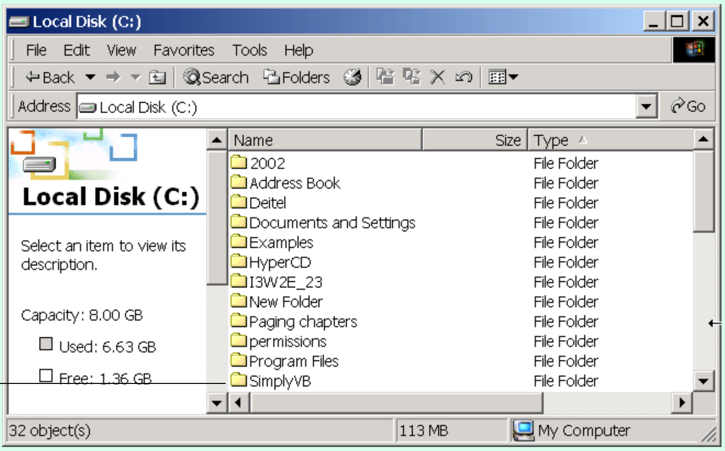
Figure 7 Creating a new folder.



Name	Size	Type
Preface_count_c4e.xls	24 KB	Microsoft Excel Worksheet
profiles	48 KB	File
Untitled.log	1 KB	Text Document
winzip.log	2,726 KB	Text Document
New Folder		File Folder

New folder

Figure 8 New folder appears in the `C:` directory.



Name	Size	Type
2002		File Folder
Address Book		File Folder
Deitel		File Folder
Documents and Settings		File Folder
Examples		File Folder
HyperCD		File Folder
I3W2E_23		File Folder
New Folder		File Folder
Paging chapters		File Folder
permissions		File Folder
Program Files		File Folder
SimplyVB		File Folder

Newly created work folder

Figure 9 New work folder in `C:` drive.

Acknowledgments

One of the great pleasures of writing a textbook is acknowledging the efforts of many people whose names may not appear on the cover, but whose hard work, cooperation, friendship and understanding were crucial to the production of the book. Many people at Deitel & Associates, Inc., devoted long hours to this project.

- David Choffnes, Developer
- Christina J. Courtemarche, Developer
- Abbey Deitel, President
- Barbara Deitel, Chief Financial Officer
- Betsy DuWaldt, Editorial Director
- Christi Kelsey, Director of Business Development

- Jonathan Liperi, Developer
- Laura Treibick, Director of Multimedia
- Cheryl Yaeger, Director of .NET Development

We would also like to thank the participants in the Deitel & Associates, Inc., College Internship Program: Emanuel Achildiev, Nicholas Cassie, Thiago Lucas da Silva, Mike Dos'Santos, Jimmy Nguyen and Ngale Truong.[2]

We are fortunate to have been able to work on this project with the talented and dedicated team of publishing professionals at Prentice Hall. We especially appreciate the extraordinary efforts of our Computer Science editor, Petra Recter and her boss—our mentor in publishing—Marcia Horton, Editorial Director of Prentice Hall's Engineering and Computer Science Team. Tom Manshreck and Vince O'Brien did a marvelous job managing the production of the book. Chirag Thakkar and John Lovell served as production editors and Sarah Parker handled editorial responsibilities on the book's extensive ancillary package. We would like to thank the design team that created a completely new look and feel for the *Simply* series—Carole Anson, Paul Belfanti, Tom Manshreck, Jonathan Boylan, John Root and Gary Gray. We would like to thank Carol Trueheart, Editor-in-Chief of Development for Prentice Hall's Engineering, Science and Math Division, and her team of developmental editors—Laura Cheu, Deena Cloud and Susan Cohen.

We owe special thanks to the creativity of Tamara Newnam (`smart_art@earthlink.net`) and illustrator Steve Lefkowitz. They produced the art work for the cover, and Tammy created the delightful bug creature who shares with you the book's programming tips.

We wish to acknowledge the efforts of our reviewers and to thank Jennifer Cappello of Prentice Hall, who managed the review process. The 42 reviewers from colleges, industry and Microsoft helped us to get this book "right." Adhering to a tight time schedule, these reviewers scrutinized the text and the programs, providing countless suggestions for improving the accuracy and completeness of the presentation. It is a privilege to have the guidance of such talented and busy professionals.

Simply Visual Basic .NET reviewers:
Cameron McColl (Microsoft)
Colin Merry (Microsoft)
Jeffrey Welton (Microsoft)
Judith Ashworth (Orillion USA, Inc.)
James Ball (Indiana State University)
Robert Benavides (Collin County Community College)
Chadi Boudiab (Georgia Perimeter College)
Charles Cadenhead (Brookhaven College)
Kunal Cheda (`DotNetExtreme.com`)
Mave Coxon (Lansing Community College)
Chris Crane (Independent Consultant)
Sergio Davalos (University of Washington-Tacoma)
David Fullerton (Yeshiva University)
George Gintowt (William Rainey Harper College)

[2] The Deitel & Associates, Inc. College Internship Program offers a limited number of salaried positions to college students majoring in Computer Science, Information Technology, Marketing and English. Students work at our corporate headquarters in Maynard, Massachusetts full-time in the summers and (for those attending college in the Boston area) part-time during the academic year. We also offer full-time internship positions for students interested in taking a semester off from school to gain industry experience. Regular full-time positions are available to college graduates. For more information, please contact Abbey Deitel at `deitel@deitel.com`, visit our Web site, `www.deitel.com`, and subscribe to our free e-mail newsletter at `www.deitel.com/newsletter/subscribe.html`.

James Gips (Boston College)
Manu Gupta (Patni Computer Systems)
Richard Hewer (Ferris State University)
James Huddleston (Independent Consultant)
Terrell Hull (Sun Certified Java Architect, Rational Qualified Practitioner)
Jeff Jones (A.D.A.M. Inc.)
Faisal Kaleem (Florida International University)
Yashavant Kanetkar (KICIT Pvt. Ltd.)
Dhananjay Katre (Patni Computer Systems, Ltd.)
Kurt Kominek (Northeast State Technical CC)
Stan Kurkovsky (Columbus State University)
Brian Larson (Modesto Junior College)
Sukan Makmuri (DeVry-Fremont)
Ken McLean (Northern Virginia CC)
Gordon McNorton (Collin County Community College)
Manish Mehta (Independent Consultant)
Marilyn Meyer (Fresno City College)
John Mueller (DataCon Services)
Narayana Rao Surapaneni (Patni Computer Systems)
Michael Rudisill (Northern Michigan University)
Sara Rushinek (University of Miami)
Praveen Sadhu (Infodat Solutions, Inc.)
Kenneth Schoonover (Chubb Technical Institute)
Andrea Shelly (Florida International University)
Robert Taylor (Lansing Community College)
Yateen Thakkar (Syntel India Ltd.)
Catherine Wyman (DeVry-Phoenix)
David Zeng (DeVry-Calgary)

We would sincerely appreciate your comments, criticisms, corrections and suggestions for improving the text. Please address all correspondence to:

`deitel@deitel.com`

We will respond promptly.

Well, that's it for now. Welcome to the exciting world of Visual Basic .NET programming. We hope you enjoy this look at leading-edge computer applications development. Good luck!

Dr. Harvey M. Deitel
Paul J. Deitel
Tem R. Nieto

About the Authors

Dr. Harvey M. Deitel, Chairman of Deitel & Associates, Inc., has 41 years experience in the computing field, including extensive industry and academic experience. Dr. Deitel earned B.S. and M.S. degrees from the Massachusetts Institute of Technology and a Ph.D. from Boston University. He worked on the pioneering virtual-memory operating-systems projects at IBM and MIT that developed techniques now widely implemented in systems such as UNIX, Linux and Windows NT. He has 20 years of college teaching experience and served as the Chairman of the Computer Science Department at Boston College before founding Deitel & Associates, Inc., with his son, Paul J. Deitel. He is the author or co-author of several dozen books and multimedia packages. With translations published in numerous foreign languages, Dr. Deitel's texts have earned international recognition. Dr. Deitel has delivered professional seminars to major corporations, government organizations and various branches of the military.

Paul J. Deitel, CEO and Chief Technical Officer of Deitel & Associates, Inc., is a graduate of the Massachusetts Institute of Technology's Sloan School of Management, where he studied information technology. Through Deitel & Associates, Inc., he has delivered professional seminars to numerous industry and government clients and has lectured on C++ and Java for the Boston Chapter of the Association for Computing Machinery. He and his father, Dr. Harvey M. Deitel, are the world's best-selling Computer Science textbook authors.

Tem R. Nieto, Director of Product Development of Deitel & Associates, Inc., is a graduate of the Massachusetts Institute of Technology, where he studied engineering and computing. Through Deitel & Associates, Inc., he has delivered courses for numerous industry and government clients. He has co-authored many books and multimedia packages with the Deitels and has contributed to virtually every Deitel & Associates, Inc. publication.

About Deitel & Associates, Inc.

Deitel & Associates, Inc., is an internationally recognized corporate-training and content-creation organization specializing in computer programming languages education, object technology and Internet/World Wide Web software technology. Through its 27-year publishing partnership with Prentice Hall, Deitel & Associates, Inc. publishes leading-edge programming textbooks, professional books, interactive CD-ROM-based multimedia *Cyber Classrooms*, *Complete Training Courses* and course management systems e-content. To learn more about Deitel & Associates, Inc., its publications and its worldwide corporate on-site curriculum, visit:

```
www.deitel.com
```

Individuals wishing to purchase Deitel books, *Cyber Classrooms* and *Complete Training Courses* can do so through bookstores or online booksellers through:

```
www.deitel.com
www.prenhall.com/deitel
www.InformIT.com/deitel
```

Bulk orders by corporations and academic institutions should be placed directly with Prentice Hall. For ordering information, please visit:

```
www.prenhall.com/deitel
```

The Deitel Buzz Online Newsletter

Our free e-mail newsletter includes commentary on industry trends and developments, links to articles and resources from our published books and upcoming publications, information on future publications, product-release schedules and more. For opt-in registration, visit `www.deitel.com/newsletter/subscribe.html`.