

Excel[®] 2007 Power Programming with VBA

by John Walkenbach



Wiley Publishing, Inc.

Excel[®] 2007 Power Programming with VBA

Excel[®] 2007 Power Programming with VBA

by John Walkenbach



Wiley Publishing, Inc.

Wiley Publishing, Inc.

Excel® 2007 Power Programming with VBA

Published by
Wiley Publishing, Inc.
111 River Street
Hoboken, NJ 07030-5774
www.wiley.com

Copyright © 2007 by Wiley Publishing, Inc., Indianapolis, Indiana

Published by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

Library of Congress Control Number: 2006939606

ISBN: 978-0-470-04401-8

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Legal Department, Wiley Publishing, Inc., 10475 Crosspoint Blvd., Indianapolis, IN 46256, (317) 572-3447, fax (317) 572-4355, or online at <http://www.wiley.com/go/permissions>.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ. FULFILLMENT OF EACH COUPON OFFER IS THE SOLE RESPONSIBILITY OF THE OFFEROR.

For general information on our other products and services, please contact our Customer Care Department within the U.S. at 800-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002.

For technical support, please visit www.wiley.com/techsupport.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Trademarks: Wiley, the Wiley Publishing logo, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates in the United States and other countries, and may not be used without written permission. Microsoft and Excel are registered trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book.



About the Author

John Walkenbach is author of about 50 spreadsheet books and lives in southern Arizona.
Visit his Web site: <http://j-walk.com>.

***Acquisitions, Editorial, and
Media Development***

Senior Project Editor: Christopher Morris
(Previous Edition: Linda Morris)

Senior Acquisitions Editor: Bob Woerner

Copy Editor: Andy Hollandbeck

Technical Editor: Niek Otten

Editorial Manager: Kevin Kirschner

Media Development Specialists:

Angela Denny, Kate Jenkins,
Steven Kudirka, Kit Malone

Media Development Coordinator:

Laura Atkinson

Media Project Supervisor: Laura Moss

Media Development Manager:

Laura VanWinkle

Editorial Assistant: Amanda Foxworth

Sr. Editorial Assistant: Cherie Case

Composition Services

Project Coordinator: Kristie Rees

Layout and Graphics: Denny Hager,
Joyce Haughey, Jennifer Mayberry,
Heather Ryan

Proofreader: John Greenough

Indexer: Johnna VanHoose

Anniversary Logo Design: Richard Pacifico

Publishing and Editorial for Technology Dummies

Richard Swadley, Vice President and Executive Group Publisher

Andy Cummings, Vice President and Publisher

Mary Bednarek, Executive Acquisitions Director

Mary C. Corder, Editorial Director

Publishing for Consumer Dummies

Diane Graves Steele, Vice President and Publisher

Joyce Pepple, Acquisitions Director

Composition Services

Gerry Fahey, Vice President of Production Services

Debbie Stailey, Director of Composition Services

Preface

Welcome to *Excel 2007 Power Programming with VBA*. If your job involves developing spreadsheets that others will use — or if you simply want to get the most out of Excel — you’ve come to the right place.

Topics Covered

This book focuses on Visual Basic for Applications (VBA), the programming language built into Excel (and other applications that make up Microsoft Office). More specifically, it will show you how to write programs that automate various tasks in Excel. This book covers everything from recording simple macros through creating sophisticated user-oriented applications and utilities.

This book does *not* cover Microsoft Visual Studio Tools for Office (VSTO). VSTO is a relatively new technology that uses Visual Basic .NET and Microsoft Visual C#. VSTO can also be used to control Excel and other Microsoft Office applications.

What You Need to Know

This is not a book for beginning Excel users. If you have no experience with Excel, a better choice might be my *Excel 2007 Bible*, which provides comprehensive coverage of all the features of Excel. That book is meant for users of all levels.

To get the most out of this book, you should be a relatively experienced Excel user. I didn’t spend much time writing basic how-to information. In fact, I assume that you know the following:

- How to create workbooks, insert sheets, save files, and so on
- How to navigate through a workbook
- How to use the Excel 2007 Ribbon
- How to enter formulas
- How to use Excel’s worksheet functions
- How to name cells and ranges
- How to use basic Windows features, such as file management techniques and the Clipboard

If you don't know how to perform the preceding tasks, you could find some of this material over your head, so consider yourself warned. If you're an experienced spreadsheet user who hasn't used Excel 2007, Chapter 2 presents a brief overview of what this product offers.

What You Need to Have

To make the best use of this book, you need a copy of Excel 2007. Although most of the material also applies to Excel 2000 and later versions, I assume that you are using Excel 2007. Although Excel 2007 is radically different from its predecessors, the VBA environment has not changed at all. If you plan to develop applications that will be used in earlier versions of Excel, I strongly suggest that you *don't* use Excel 2007 for your development work.

Most of the material in this book also applies to Excel for Macintosh. However, I did no compatibility testing with the Mac version, so you're on your own.

Any computer system that can run Windows will suffice, but you'll be much better off with a fast machine with plenty of memory. Excel is a large program, and using it on a slower system or a system with minimal memory can be extremely frustrating.

I recommend using a high-resolution video driver (1024 × 768 is adequate, and 1600 × 1200 is even better). For optimal results, try a dual-monitor system and place Excel on one screen and the Visual Basic Editor on the other. You'll soon become spoiled.

To use the examples on the companion CD, you also need a CD-ROM drive.

Conventions in This Book

Take a minute to skim this section and learn some of the typographic conventions used throughout this book.

Excel commands

Excel 2007 features a brand new “menu-less” user interface. In place of a menu system, Excel uses a context-sensitive Ribbon system. The words along the top (such as Insert, View, and so on) are known as *tabs*. Click a tab, and the Ribbon of icons displays the commands that are most suited to the task at hand. Each icon has a name that is (usually) displayed next to or below the icon. The icons are arranged in groups, and the group name appears below the icons.

The convention I use in this book is to indicate the tab name, followed by the group name, followed by the icon name. So, for example, the command used to toggle word wrap within a cell is indicated as:

Home↔Alignment↔Wrap Text

The large round icon in the upper left corner of Excel 2007's window is known as the Office Button. When I refer to commands that use the Office Button, I abbreviate it as Office. For example, the following command displays the Excel Options dialog box:

Office ⇨ Excel Options

VBA editor commands

The VBA editor is the window in which you work with your VBA code. The VBA editor uses the traditional menu-and-toolbar interface. A command like the following means to click the Tools menu and select the References menu item:

Tools ⇨ References

Keyboard conventions

You need to use the keyboard to enter data. In addition, you can work with menus and dialog boxes directly from the keyboard — a method that you might find easier if your hands are already positioned over the keys.

INPUT

Input that you type from the keyboard appears in boldface — for example, enter **=SUM(B2: B50)** into cell B51.

More lengthy input usually appears on a separate line in a monospace font. For example, I might instruct you to enter the following formula:

```
=VLOOKUP (StockNumber, PriceList, 2)
```

VBA CODE

This book contains many snippets of VBA code as well as complete procedure listings. Each listing appears in a monospace font; each line of code occupies a separate line. (I copied these listings directly from the VBA module and pasted them into my word processor.) To make the code easier to read, I often use one or more tabs to create indentations. Indentation is optional, but it does help to delineate statements that go together.

If a line of code doesn't fit on a single line in this book, I use the standard VBA line continuation sequence: At the end of a line, a space followed by an underscore character indicates that the line of code extends to the next line. For example, the following two lines are a single code statement:

```
If Right (ActiveCell, 1) = "!" Then ActiveCell _  
    = Left (ActiveCell, Len (ActiveCell) - 1)
```

You can enter this code either on two lines, exactly as shown, or on a single line without the underscore character.

FUNCTIONS, FILENAMES, AND NAMED RANGES

Excel's worksheet functions appear in uppercase font, like so: "Enter a SUM formula in cell C20." VBA procedure names, properties, methods, and objects appear in monospace font: "Execute the `GetTotals` procedure." I often use mixed upper- and lowercase to make these names easier to read.

I also use the monospace font for filenames and named ranges in a worksheet — for example: Open `myfile.xls`m and select the range named `data`.

Mouse conventions

If you're reading this book, you're well versed in mouse usage. The mouse terminology I use is all standard fare: pointing, clicking, right-clicking, dragging, and so on.

What the Icons Mean

Throughout the book, I use icons to call your attention to points that are particularly important:



NEW

I use this icon to indicate that the material discussed is new to Excel 2007.



NOTE

I use Note icons to tell you that something is important — perhaps a concept that could help you master the task at hand or something fundamental for understanding subsequent material.



TIP

Tip icons indicate a more efficient way of doing something or a technique that might not be obvious.



CD-ROM

These icons indicate that an example file is on the companion CD-ROM. (See "About the Companion CD-ROM," later in this Preface.) This CD holds many of the examples that I show in the book.



CAUTION

I use Caution icons when the operation that I'm describing can cause problems if you're not careful.

**CROSS-REFERENCE**

I use the Cross Reference icon to refer you to other chapters that have more to say on a subject.

How This Book Is Organized

The chapters of this book are grouped into eight main parts.

Part I: Some Essential Background

In this part, I set the stage for the rest of the book. Chapter 1 presents a brief history of spreadsheets so that you can see how Excel fits into the big picture. In Chapter 2, I offer a conceptual overview of Excel 2007 — quite useful for experienced spreadsheet users who are switching to Excel. In Chapter 3, I cover the essentials of formulas, including some clever techniques that might be new to you. Chapter 4 covers the ins and outs of the various files used and generated by Excel.

Part II: Excel Application Development

This part consists of just two chapters. In Chapter 5, I broadly discuss the concept of a spreadsheet application. Chapter 6 goes into more detail and covers the steps typically involved in a spreadsheet application development project.

Part III: Understanding Visual Basic for Applications

Chapters 7 through 11 make up Part III, and these chapters include everything that you need to know to learn VBA. In this part, I introduce you to VBA, provide programming fundamentals, and detail how to develop VBA subroutines and functions. Chapter 11 contains many useful VBA examples.

Part IV: Working with UserForms

The four chapters in this part cover custom dialog boxes (also known as *UserForms*). Chapter 12 presents some built-in alternatives to creating custom UserForms. Chapter 13 provides an introduction to UserForms and the various controls that you can use. Chapters 14 and 15 present many examples of custom dialog boxes, ranging from basic to advanced.

Part V: Advanced Programming Techniques

Part V covers additional techniques that are often considered advanced. The first three chapters discuss how to develop utilities and how to use VBA to work with pivot tables and charts. Chapter 19 covers *event handling*, which enables you to execute procedures automatically when certain events occur. Chapter 20 discusses various techniques that you can

use to interact with other applications (such as Word). Chapter 21 concludes Part V with an in-depth discussion of creating add-ins.

Part VI: Developing Applications

The chapters in Part VI deal with important elements of creating user-oriented applications. Chapter 22 discusses how to modify the new Ribbon interface. Chapter 23 describes how to modify Excel's shortcut menus. Chapter 24 presents several different ways to provide online help for your applications. In Chapter 25, I present some basic information about developing user-oriented applications, and I describe such an application in detail.

Part VII: Other Topics

The six chapters in Part VII cover additional topics. Chapter 26 presents information regarding compatibility. In Chapter 27, I discuss various ways to use VBA to work with files. In Chapter 28, I explain how to use VBA to manipulate Visual Basic components such as UserForms and modules. Chapter 29 covers the topic of class modules. Chapter 30 explains how to work with color in Excel. I finish the part with a useful chapter that answers many common questions about Excel programming.

Part VIII: Appendixes

Four appendixes round out the book. Appendix A contains useful information about Excel resources online. Appendix B is a reference guide to all VBA's keywords (statements and functions). I explain VBA error codes in Appendix C, and Appendix D describes the files available on the companion CD-ROM.

About the Companion CD-ROM

The inside back cover of this book contains a CD-ROM that holds many useful examples that I discuss in the text. When I write about computer-related material, I emphasize learning by example. I know that I learn more from a well-thought-out example than from reading a dozen pages in a book. I assume that this is true for many other people. Consequently, I spent more time developing the examples on the CD-ROM than I did writing chapters.

The files on the companion CD-ROM are not compressed, so you can access them directly from the CD.



CROSS-REFERENCE

Refer to Appendix D for a description of each file on the CD-ROM.

About the Power Utility Pak Offer

Toward the back of the book, you'll find a coupon that you can redeem for a discounted copy of my popular Power Utility Pak software. PUP is an award-winning collection of useful Excel utilities and many new worksheet functions. I developed this package exclusively with VBA.

I think you'll find this product useful in your day-to-day work with Excel. You can also purchase the complete VBA source code for a nominal fee. Studying the code is an excellent way to pick up some useful programming techniques.

You can take Power Utility Pak for a test drive by installing the 30-day trial version available at my Web site:

<http://j-walk.com/ss>

How to Use This Book

You can use this book any way that you please. If you choose to read it from cover to cover, be my guest. But because I'm dealing with intermediate-to-advanced subject matter, the chapter order is often immaterial. I suspect that most readers will skip around, picking up useful tidbits here and there. If you're faced with a challenging task, you might try the index first to see whether the book specifically addresses your problem.

Reach Out

The publisher and I want your feedback. After you've had a chance to use this book, please take a moment to visit the Wiley Publishing Web site to give us your comments. (Go to www.wiley.com and then click the Contact Us link.) Please be honest in your evaluation. If you thought a particular chapter didn't tell you enough, let us know. Of course, I would prefer to receive comments like, "This is the best book I've ever read," or "Thanks to this book, I was promoted and now make \$112,000 a year."

I get at least a half dozen questions every day, via e-mail, from people who have read my books. I appreciate the feedback. Unfortunately, I simply don't have the time to reply to questions. Appendix A provides a good list of sources that *can* answer your questions.

I also invite you to visit my Web site, which contains lots of Excel-related material. Despite the massive attempts to make this book completely accurate, a few errors have probably crept into its pages. My Web site includes a list of any such errors. The URL is

<http://j-walk.com/ss/w>

Contents at a Glance

Preface

vii

Part I: Some Essential Background

Chapter 1: Excel 2007: Where It Came From	3
Chapter 2: Excel in a Nutshell	15
Chapter 3: Formula Tricks and Techniques	47
Chapter 4: Understanding Excel's Files	73

Part II: Excel Application Development

Chapter 5: What Is a Spreadsheet Application?	97
Chapter 6: Essentials of Spreadsheet Application Development	109

Part III: Understanding Visual Basic for Applications

Chapter 7: Introducing Visual Basic for Applications	133
Chapter 8: VBA Programming Fundamentals	189
Chapter 9: Working with VBA Sub Procedures	237
Chapter 10: Creating Function Procedures	275
Chapter 11: VBA Programming Examples and Techniques	315

Part IV: Working with UserForms

Chapter 12: Custom Dialog Box Alternatives	389
Chapter 13: Introducing UserForms	413
Chapter 14: UserForm Examples	449
Chapter 15: Advanced UserForm Techniques	487

Part V: Advanced Programming Techniques

Chapter 16: Developing Excel Utilities with VBA	533
Chapter 17: Working with Pivot Tables	555
Chapter 18: Working with Charts	571
Chapter 19: Understanding Excel's Events	629
Chapter 20: Interacting with Other Applications	669
Chapter 21: Creating and Using Add-Ins	699

Part VI: Developing Applications

Chapter 22: Working with the Ribbon	727
Chapter 23: Working with Shortcut Menus	761

Chapter 24: Providing Help for Your Applications	781
Chapter 25: Developing User-Oriented Applications	801

Part VII: Other Topics

Chapter 26: Compatibility Issues	817
Chapter 27: Manipulating Files with VBA	829
Chapter 28: Manipulating Visual Basic Components	859
Chapter 29: Understanding Class Modules	885
Chapter 30: Working with Colors	901
Chapter 31: Frequently Asked Questions about Excel Programming	929

Part VIII: Appendixes

Appendix A: Excel Resources Online	971
Appendix B: VBA Statements and Functions Reference	977
Appendix C: VBA Error Codes	989
Appendix D: What's on the CD-ROM	993
Index	1011

Table of Contents

Preface

vii

Part I: Some Essential Background

Chapter 1: Excel 2007: Where It Came From	3
A Brief History of Spreadsheets	3
It all started with VisiCalc	4
Lotus 1-2-3	5
Quattro Pro	7
Microsoft Excel	7
Why Excel Is Great for Developers	13
Excel's Role in Microsoft's Strategy	14
Chapter 2: Excel in a Nutshell	15
Thinking in Terms of Objects	15
Workbooks	16
Worksheets	17
Chart sheets	18
XLM macro sheets	18
Excel 5/95 dialog sheets	20
Excel's User Interface	20
Introducing the Ribbon	20
Shortcut menus	26
Dialog boxes	27
Keyboard shortcuts	28
Smart Tags	28
Task pane	29
Customizing the Display	30
Data Entry	30
Formulas, Functions, and Names	30
Selecting Objects	32
Formatting	33
Numeric formatting	33
Stylistic formatting	34
Protection Options	34
Protecting formulas from being overwritten	35
Protecting a workbook's structure	36
Applying password protection to a workbook	36
Protecting VBA code with a password	36
Charts	37
Shapes and SmartArt	38
Database Access	39
Worksheet databases	39
External databases	40

Internet Features	41
Analysis Tools	41
Outlines	42
Analysis ToolPak	42
Pivot tables	42
Solver	42
XML features	42
Add-Ins	43
Macros and Programming	44
File Format	44
Excel's Help System	44
Chapter 3: Formula Tricks and Techniques	47
About Formulas	48
Calculating Formulas	48
Cell and Range References	49
Why use references that aren't relative?	50
About R1C1 notation	50
Referencing other sheets or workbooks	51
Using Names	53
Naming cells and ranges	53
Applying names to existing references	54
Intersecting names	55
Naming columns and rows	55
Scoping names	56
Naming constants	56
Naming formulas	57
Naming objects	59
Formula Errors	59
Array Formulas	60
An array formula example	61
An array formula calendar	62
Array formula pros and cons	63
Counting and Summing Techniques	63
Counting formula examples	64
Summing formula examples	66
Other counting tools	67
Working with Dates and Times	67
Entering dates and times	67
Using pre-1900 dates	68
Creating Megaformulas	69
Chapter 4: Understanding Excel's Files	73
Starting Excel	73
File Types	76
Excel file formats	76
Text file formats	78
Database file formats	79
Other file formats	79

Working with Template Files	80
Viewing templates	81
Creating templates	82
Creating workbook templates	84
Inside an Excel File	84
Dissecting a file	85
Why is the file format important?	88
The QAT File	89
The XLB File	90
Add-In Files	90
Excel Settings in the Registry	91
About the Registry	91
Excel's settings	92

Part II: Excel Application Development

Chapter 5: What Is a Spreadsheet Application?	97
Spreadsheet Applications	98
The Developer and the End User	99
Who are developers? What do they do?	99
Classifying spreadsheet users	100
The audience for spreadsheet applications	101
Solving Problems with Excel	102
Basic Spreadsheet Types	103
Quick-and-dirty spreadsheets	103
For-your-eyes-only spreadsheets	104
Single-user applications	104
Spaghetti applications	104
Utility applications	105
Add-ins that contain worksheet functions	105
Single-block budgets	106
What-if models	106
Data storage and access spreadsheets	106
Database front ends	107
Turnkey applications	107
Chapter 6: Essentials of Spreadsheet Application Development	109
Determining User Needs	110
Planning an Application That Meets User Needs	111
Determining the Most Appropriate User Interface	114
Customizing the Ribbon	114
Customizing shortcut menus	116
Creating shortcut keys	116
Creating custom dialog boxes	117
Using ActiveX controls on a worksheet	118
Executing the development effort	120
Concerning Yourself with the End User	120
Testing the application	121
Making the application bulletproof	122

Making the application aesthetically appealing and intuitive	124
Creating a user Help system	125
Documenting the development effort	126
Distributing the application to the user	127
Updating the application when necessary	127
Other Development Issues	128
The user's installed version of Excel	128
Language issues	128
System speed	129
Video modes	129

Part III: Understanding Visual Basic for Applications

Chapter 7: Introducing Visual Basic for Applications	133
Some BASIC Background	134
About VBA	134
Object models	134
VBA versus XLM	135
The Basics of VBA	135
Introducing the Visual Basic Editor	139
Displaying Excel's Developer tab	139
Activating the VBE	139
The VBE windows	140
Working with the Project Explorer	142
Adding a new VBA module	143
Removing a VBA module	143
Exporting and importing objects	143
Working with Code Windows	144
Minimizing and maximizing windows	144
Storing VBA code	145
Entering VBA code	145
Customizing the VBE Environment	152
Using the Editor tab	152
Using the Editor Format tab	155
Using the General tab	156
Using the Docking tab	157
The Macro Recorder	158
What the macro recorder actually records	158
Relative or absolute?	159
Recording options	162
Cleaning up recorded macros	163
About Objects and Collections	165
The object hierarchy	165
About collections	166
Referring to objects	167
Properties and Methods	167
Object properties	168
Object methods	169

The Comment Object: A Case Study	171
Viewing Help for the Comment object	171
Properties of a Comment object	171
Methods of a Comment object	172
The Comments collection	173
About the Comment property	174
Objects within a Comment object	175
Determining whether a cell has a comment	176
Adding a new Comment object	177
Some Useful Application Properties	177
Working with Range Objects	179
The Range property	179
The Cells property	182
The Offset property	184
Things to Know about Objects	185
Essential concepts to remember	185
Learning more about objects and properties	186
Chapter 8: VBA Programming Fundamentals	189
VBA Language Elements: An Overview	189
Comments	192
Variables, Data Types, and Constants	193
Defining data types	194
Declaring variables	197
Scoping variables	199
Working with constants	203
Working with strings	205
Working with dates	206
Assignment Statements	207
Arrays	209
Declaring arrays	210
Declaring multidimensional arrays	210
Declaring dynamic arrays	211
Object Variables	211
User-Defined Data Types	212
Built-in Functions	213
Manipulating Objects and Collections	217
With-End With constructs	217
For Each-Next constructs	218
Controlling Code Execution	220
GoTo statements	220
If-Then constructs	221
Select Case constructs	225
Looping blocks of instructions	228
Chapter 9: Working with VBA Sub Procedures	237
About Procedures	237
Declaring a Sub procedure	238
Scoping a procedure	239

Executing Sub Procedures	240
Executing a procedure with the Run Sub/UserForm command	241
Executing a procedure from the Macro dialog box	241
Executing a procedure with a Ctrl+shortcut key combination	242
Executing a procedure from the Ribbon	243
Executing a procedure from a customized shortcut menu	244
Executing a procedure from another procedure	244
Executing a procedure by clicking an object	249
Executing a procedure when an event occurs	250
Executing a procedure from the Immediate window	250
Passing Arguments to Procedures	251
Error-Handling Techniques	255
Trapping errors	256
Error-handling examples	257
A Realistic Example That Uses Sub Procedures	260
The goal	260
Project requirements	260
What you know	261
The approach	261
What you need to know	262
Some preliminary recording	262
Initial setup	264
Code writing	265
Writing the Sort procedure	266
More testing	270
Fixing the problems	270
Utility availability	274
Evaluating the project	274
Chapter 10: Creating Function Procedures	275
Sub Procedures versus Function Procedures	276
Why Create Custom Functions?	276
An Introductory Function Example	277
A custom function	277
Using the function in a worksheet	278
Using the function in a VBA procedure	278
Analyzing the custom function	279
Function Procedures	281
Declaring a function	281
A function's scope	283
Executing function procedures	283
Function Arguments	285
Function Examples	286
Functions with no argument	286
A function with one argument	289
A function with two arguments	292
A function with an array argument	293
A function with optional arguments	294
A function that returns a VBA array	296

A function that returns an error value	299
A function with an indefinite number of arguments	300
Emulating Excel's SUM function	301
Debugging Functions	305
Dealing with the Insert Function Dialog Box	306
Specifying a function category	307
Adding a function description	308
Using Add-ins to Store Custom Functions	310
Using the Windows API	311
Windows API examples	311
Determining the Windows directory	311
Detecting the Shift key	313
Learning more about API functions	314
Chapter 11: VBA Programming Examples and Techniques	315
Working with Ranges	316
Copying a range	316
Moving a range	318
Copying a variably sized range	318
Selecting or otherwise identifying various types of ranges	319
Prompting for a cell value	321
Entering a value in the next empty cell	323
Pausing a macro to get a user-selected range	324
Counting selected cells	326
Determining the type of selected range	327
Looping through a selected range efficiently	329
Deleting all empty rows	332
Duplicating rows a variable number of times	333
Determining whether a range is contained in another range	334
Determining a cell's data type	335
Reading and writing ranges	336
A better way to write to a range	337
Transferring one-dimensional arrays	339
Transferring a range to a variant array	339
Selecting cells by value	340
Copying a noncontiguous range	342
Working with Workbooks and Sheets	344
Saving all workbooks	344
Saving and closing all workbooks	344
Hiding all but the selection	345
Synchronizing worksheets	346
VBA Techniques	347
Toggling a Boolean property	347
Determining the number of printed pages	348
Displaying the date and time	349
Getting a list of fonts	351
Sorting an array	352
Processing a series of files	354

Some Useful Functions for Use in Your Code	356
The FileExists function	356
The FileNameOnly function	356
The PathExists function	357
The RangeNameExists function	357
The SheetExists function	358
The WorkbookIsOpen function	358
Retrieving a value from a closed workbook	358
Some Useful Worksheet Functions	361
Returning cell formatting information	361
A talking worksheet	363
Displaying the date when a file was saved or printed	363
Understanding object parents	364
Counting cells between two values	365
Counting visible cells in a range	366
Determining the last non-empty cell in a column or row	366
Does a string match a pattern?	368
Extracting the nth element from a string	369
A multifunctional function	371
The SheetOffset function	372
Returning the maximum value across all worksheets	373
Returning an array of nonduplicated random integers	374
Randomizing a range	375
Windows API Calls	377
Determining file associations	377
Determining disk drive information	378
Determining default printer information	379
Determining video display information	380
Adding sound to your applications	382
Reading from and writing to the Registry	384

Part IV: Working with UserForms

Chapter 12: Custom Dialog Box Alternatives	389
Before You Create That UserForm . . .	390
Using an Input Box	390
The VBA InputBox function	390
The Excel InputBox method	392
The VBA MsgBox Function	394
The Excel GetOpenFilename Method	399
The Excel GetSaveAsFilename Method	403
Prompting for a Directory	403
Using a Windows API function to select a directory	404
Using the FileDialog object to select a directory	406
Displaying Excel's Built-In Dialog Boxes	407
About the Dialogs collection	407
Executing Ribbon commands	408

Displaying a Data Form	410
Making the data form accessible	410
Displaying a data form by using VBA	411
Chapter 13: Introducing UserForms	413
How Excel Handles Custom Dialog Boxes	414
Inserting a New UserForm	414
Adding Controls to a UserForm	415
Toolbox Controls	416
CheckBox	416
ComboBox	416
CommandButton	417
Frame	417
Image	417
Label	418
ListBox	418
MultiPage	418
OptionButton	418
RefEdit	418
ScrollBar	418
SpinButton	419
TabStrip	419
TextBox	419
ToggleButton	419
Adjusting UserForm Controls	421
Adjusting a Control's Properties	422
Using the Properties window	422
Common properties	424
Learning more about properties	424
Accommodating keyboard users	424
Displaying and Closing UserForms	427
Displaying a UserForm	427
Closing a UserForm	428
About event handler procedures	429
Creating a UserForm: An Example	430
Creating the UserForm	430
Writing code to display the dialog box	433
Testing the dialog box	434
Adding event handler procedures	435
Validating the data	436
The finished dialog box	437
Understanding UserForm Events	437
Learning about events	437
UserForm events	438
SpinButton events	439
Pairing a SpinButton with a TextBox	441
Referencing UserForm Controls	444

Customizing the Toolbox	445
Changing icons or tip text	445
Adding new pages	445
Customizing or combining controls	445
Adding new ActiveX controls	447
Creating UserForm Templates	447
A UserForm Checklist	448
Chapter 14: UserForm Examples	449
Creating a UserForm “Menu”	450
Using CommandButtons in a UserForm	450
Using a ListBox in a UserForm	450
Selecting Ranges from a UserForm	452
Creating a Splash Screen	454
Disabling a UserForm’s Close Button	456
Changing a UserForm’s Size	457
Zooming and Scrolling a Sheet from a UserForm	458
ListBox Techniques	460
About the ListBox control	461
Adding items to a ListBox control	461
Determining the selected item	466
Determining multiple selections in a ListBox	466
Multiple lists in a single ListBox	467
ListBox item transfer	468
Moving items in a ListBox	470
Working with multicolumn ListBox controls	472
Using a ListBox to select worksheet rows	474
Using a ListBox to activate a sheet	476
Using the MultiPage Control in a UserForm	479
Using an External Control	480
Animating a Label	482
Chapter 15: Advanced UserForm Techniques	487
A Modeless Dialog Box	488
Displaying a Progress Indicator	491
Creating a standalone progress indicator	492
Showing a progress indicator by using a MultiPage control	496
Showing a progress indicator without using a MultiPage control	499
Creating Wizards	500
Setting up the MultiPage control for the wizard	500
Adding the buttons to the wizard UserForm	501
Programming the wizard buttons	502
Programming dependencies in a wizard	503
Performing the task with the wizard	505
Emulating the MsgBox Function	506
MsgBox emulation: MyMsgBox code	507
How the MyMsgBox function works	508
Using the MyMsgBox function in the MsgBox emulation	510
A UserForm with Movable Controls	510

A UserForm with No Title Bar	511
Simulating a Toolbar with a UserForm	513
A Resizable UserForm	515
Handling Multiple UserForm Controls with One Event Handler	520
Selecting a Color in a UserForm	523
Displaying a Chart in a UserForm	524
General steps to display a chart in a userform	525
Saving a chart as a GIF file	525
Changing the Image control Picture property	526
An Enhanced Data Form	526
About the Enhanced Data Form	527
Installing the Enhanced Data Form add-in	528
A Puzzle on a UserForm	529

Part V: Advanced Programming Techniques

Chapter 16: Developing Excel Utilities with VBA	533
About Excel Utilities	533
Using VBA to Develop Utilities	534
What Makes a Good Utility?	535
Text Tools: The Anatomy of a Utility	535
Background for Text Tools	536
Project goals for Text Tools	537
The Text Tools workbook	537
How the Text Tools utility works	538
The UserForm for the Text Tools utility	538
The Module1 VBA module	540
The UserForm1 code module	542
Making the Text Tools utility efficient	544
Saving the Text Tools utility settings	545
Implementing Undo	547
Displaying the Help file	549
Adding the RibbonX code	550
Post-mortem of the project	553
Understand the Text Tools utility	554
More about Excel Utilities	554
Chapter 17: Working with Pivot Tables	555
An Introductory Pivot Table Example	555
Creating a pivot table	556
Examining the recorded code for the pivot table	558
Cleaning up the recorded pivot table code	558
Creating a More Complex Pivot Table	560
Data for a more complex pivot table	560
The code that created the pivot table	562
How the more complex pivot table works	563
Creating Multiple Pivot Tables	565
Creating a Reverse Pivot Table	568

Chapter 18: Working with Charts	571
About Charts	571
Chart locations	572
The macro recorder and charts	572
The Chart object model	574
Common VBA Charting Techniques	575
Creating an embedded chart	575
Creating a chart on a chart sheet	577
Using VBA to activate a chart	578
Moving a chart	578
Using VBA to deactivate a chart	580
Determining whether a chart is activated	580
Deleting from the ChartObjects or Charts collection	581
Looping through all charts	582
Sizing and aligning ChartObjects	584
Exporting a chart	585
Exporting all graphics	586
Using VBA to Apply Chart Formatting	587
Formatting a chart	587
More chart formatting examples	589
Changing the Data Used in a Chart	592
Changing chart data based on the active cell	593
Using VBA to determine the ranges used in a chart	595
Using VBA to Display Arbitrary Data Labels on a Chart	598
Displaying a Chart in a UserForm	600
Understanding Chart Events	603
An example of using Chart events	604
Enabling events for an embedded chart	607
Example: Using Chart events with an embedded chart	608
VBA Charting Tricks	610
Printing embedded charts on a full page	610
Displaying a slide show	611
Hiding series by hiding columns	612
Creating unlinked charts	613
Displaying text with the MouseOver event	615
Animating Charts	617
Scrolling a chart	619
Creating a hypocycloid chart	621
Creating a “clock” chart	622
Creating an Interactive Chart without VBA	624
Getting the data to create an interactive chart	625
Creating the Option Button controls for an interactive chart	625
Creating the city lists for the interactive chart	625
Creating the interactive chart data range	626
Creating the interactive chart	627