

Contents

Part I: Some Essential Background

1

Chapter 1: Excel 2002: Where It Came From 3

A Brief History of Spreadsheets	3
It all started with VisiCalc	3
Lotus 1-2-3	4
Quattro Pro	7
Microsoft Excel	8
Spreadsheets Today	13
Why Excel Is Great for Developers	14
Excel's Role in Microsoft's Strategy	15

Chapter 2: Excel in a Nutshell 17

Thinking in Terms of Objects	17
Workbooks	18
Worksheets	19
Chart sheets	20
XLM macro sheets	20
Excel 5/95 dialog sheets	20
Excel's User Interface	21
Menus	21
Dialog boxes	21
Toolbars	22
Drag-and-drop	23
Keyboard shortcuts	24
Customizing the Display	24
Data Entry	24
Selecting Objects	26
Formatting	26
Numeric formatting	26
Stylistic formatting	27
Formulas	27
Names	27
Functions	28
Shapes	28
Charts	30
Macros	31
Database Access	31
Worksheet databases	32
External databases	32

Internet Features	33
Analysis Tools	33
Outlines	33
Automatic subtotals	33
Scenario management	34
Analysis ToolPak	34
Pivot tables	34
Auditing	35
Solver	35
Add-Ins	35
Compatibility	36

Chapter 3: Formula Tricks and Techniques 37

About Formulas	37
Calculating Formulas	38
Cell and Range References	39
Why use references that aren't relative?	39
About R1C1 notation	40
Referencing other sheets or workbooks	41
Using Names	42
Naming cells and ranges	42
Applying names to existing references	43
Intersecting names	44
Naming columns and rows	45
Scoping names	45
Naming constants	46
Naming formulas	46
Naming objects	48
Formula Errors	48
Array Formulas	49
An array formula example	50
An array formula calendar	51
Array formula pros and cons	52
Counting and Summing Techniques	52
Using the COUNTIF or SUMIF function	52
Using array formulas to count and sum	53
Other counting tools	54
Working with Dates and Times	55
Entering dates and times	55
Using pre-1900 dates	56
Creating Megaformulas	56

Chapter 4: Understanding Excel's Files 61

Starting Excel	61
Excel's File Extensions	63
Spreadsheet File Formats Supported	64
Lotus 1-2-3 spreadsheet files	64
Quattro Pro spreadsheet files	65

Database file formats	66
Text file formats	66
Other file formats	66
Files Written by Excel	67
XLS files	67
Workspace files	68
Template files	68
Toolbar files	69
Add-in files	69
Excel and HTML	70
So how does it work?	70
Adding some complexity	71
What about interactivity?	72
Excel Settings in the Registry	74
About the Registry	74
Excel's settings	75

Part II: Excel Application Development

79

Chapter 5: What Is a Spreadsheet Application? 81

Spreadsheet Applications	81
The Developer and the End User	83
Who are developers? What do they do?	83
Classifying spreadsheet users	84
The audience for spreadsheet applications	85
Why people use spreadsheets	86
Solving Problems with a Spreadsheet	88
Basic Spreadsheet Types	89
Quick-and-dirty spreadsheets	90
For-your-eyes-only spreadsheets	90
Single-user applications	90
Spaghetti applications	91
Utility applications	91
Add-ins that contain worksheet functions	92
Single-block budgets	92
What-if models	93
Data storage and access spreadsheets	93
Database front ends	94
Turnkey applications	94

Chapter 6: Essentials of Spreadsheet Application Development . . . 95

Determining User Needs	96
Planning an Application That Meets User Needs	97

Determining the Most Appropriate User Interface	100
Creating custom dialog boxes	100
Using ActiveX controls on a worksheet	101
Customizing menus	103
Customizing toolbars	105
Creating shortcut keys	106
Executing the development effort	107
Concerning Yourself with the End User	107
Testing the application	107
Making the application bulletproof	109
Making the application aesthetically appealing and intuitive	110
Documenting the development effort	111
Distributing the application to the user	112
Updating the application when necessary	113
Other Development Issues	113
The user's installed version of Excel	114
Language issues	114
System speed	114
Video modes	115
Directory structure	115

Part III: Understanding Visual Basic for Applications 117

Chapter 7: Introducing Visual Basic for Applications 119

Some BASIC Background	119
About VBA	120
Object models	120
VBA versus XLM	120
VBA versus Lotus macros	121
VBA versus LotusScript	122
The Basics of VBA	122
Introducing the Visual Basic Editor	125
Activating the VBE	125
The VBE windows	126
Working with the Project Explorer	128
Adding a new VBA module	129
Removing a VBA module	129
Exporting and importing objects	129
Working with Code Windows	130
Minimizing and maximizing windows	130
Storing VBA code	131
Entering VBA code	132
Customizing the VBE Environment	138
Using the Editor tab	138
Using the Editor Format tab	141
Using the General tab	142
Using the Docking tab	143

The Macro Recorder	143
What is recorded	144
Relative or absolute?	145
Recording options	149
Cleaning up recorded macros	150
About Objects and Collections	152
The object hierarchy	152
About collections	153
Object referral	153
Properties and Methods	154
Object properties	154
Object methods	155
The Comment Object: A Case Study	157
Online help for the Comment object	157
Properties of a Comment object	158
Methods of a Comment object	158
The Comments collection	159
About the Comment property	160
Objects within a Comment object	161
Determining whether a cell has a comment	163
Adding a new Comment object	163
Some useful application properties	163
Working with Range Objects	165
The Range property	166
The Cells property	167
The Offset property	169
Things to Know about Objects	170
Esoteric but essential concepts to remember	170
Learn more about objects and properties	171

Chapter 8: VBA Programming Fundamentals 177

VBA Language Elements: An Overview	177
Comments	178
Variables, Data Types, and Constants	181
Defining data types	182
Declaring variables	184
Scoping variables	186
Working with constants	190
Working with strings	191
Working with dates	192
Assignment Statements	193
Arrays	195
Declaring arrays	195
Declaring multidimensional arrays	196
Object Variables	196
User-Defined Data Types	198
Built-in Functions	198

Manipulating Objects and Collections	201
With-End With constructs	202
For Each-Next constructs	202
Controlling Execution	204
GoTo statements	205
If-Then constructs	205
Select Case constructs	208
Looping blocks of instructions	212

Chapter 9: Working with VBA Sub Procedures 219

About Procedures	219
Declaring a Sub procedure	220
Scoping a procedure	221
Executing Procedures	222
Executing a procedure with the Run Run Sub/ UserForm command	223
Executing a procedure from the Macro dialog box	223
Executing a procedure using a Ctrl+shortcut key combination	224
Executing a procedure from a custom menu	225
Executing a procedure from another procedure	227
Executing a procedure from a toolbar button	230
Executing a procedure by clicking an object	231
Executing a procedure when an event occurs	233
Executing a procedure from the Immediate window	233
Passing Arguments to Procedures	234
Error-Handling Techniques	237
Trapping errors	238
Error-handling examples	239
A Realistic Example	241
The goal	241
Project requirements	242
What you know	242
The approach	243
What you need to know	243
Some preliminary recording	243
Initial setup	245
Code writing	246
Sort procedure writing	247
More testing	250
Fixing the problems	251
Utility availability	254
Evaluating the project	254

Chapter 10: Creating Function Procedures 257

Sub Procedures versus Function Procedures	257
Why Create Custom Functions?	258

An Introductory Example	258
A custom function	258
Using the function in a worksheet	259
Using the function in a VBA procedure	260
Analyzing the custom function	260
Function Procedures	262
Declaring a function	262
A function's scope	263
Executing function procedures	263
Function Arguments	266
Function Examples	266
A function with no argument	266
Another function with no argument	267
A function with one argument	268
A function with two arguments	271
A function with an array argument	272
A function with optional arguments	273
A function that returns a VBA array	274
A function that returns an error value	276
A function with an indefinite number of arguments	278
Emulating Excel's SUM function	279
Debugging Functions	281
Dealing with the Insert Function Dialog Box	282
Specifying a function category	283
Adding a function description	284
Using Add-ins to Store Custom Functions	286
Using the Windows API	287
Windows API examples	287
Determining the Windows directory	288
Detecting the Shift key	289
Learning more about API functions	290

Chapter 11: VBA Programming Examples and Techniques 291

Working with Ranges	292
Copying a range	292
Moving a range	293
Copying a variably sized range	293
Selecting or otherwise identifying various types of ranges	294
Prompting for a cell value	296
Entering a value in the next empty cell	298
Pausing a macro to get a user-selected range	299
Counting selected cells	300
Determining the type of selected range	301
Looping through a selected range efficiently	302
Deleting all empty rows	305
Determining whether a range is contained in another range	305
Determining a cell's data type	306

Reading and writing ranges	307
A better way to write to a range	308
Transferring one-dimensional arrays	310
Transferring a range to a variant array	310
Selecting the maximum value in a range	311
Selecting all cells with a particular format	312
Working with Workbooks and Sheets	314
Saving all workbooks	314
Saving and closing all workbooks	315
Accessing workbook properties	315
Synchronizing worksheets	316
VBA Techniques	317
Toggling a Boolean property	317
Determining the number of printed pages	318
Displaying the date and time	318
Getting a list of fonts	320
Sorting an array	321
Processing a series of files	323
Some Useful Functions for Use in Your Code	325
The FileExists function	325
The FileNameOnly function	325
The PathExists function	326
The RangeNameExists function	326
The SheetExists function	326
The WorkbooksOpen function	327
Retrieving a value from a closed workbook	328
Some Useful Worksheet Functions	329
Returning cell formatting information	329
Displaying the date a file was saved or printed	330
Understanding object parents	331
Counting cells between two values	332
Counting visible cells in a range	332
Determining the last nonempty cell in a column or row	333
Does a string match a pattern?	335
Extracting the nth element from a string	336
A multifunctional function	337
The SHEETOFFSET function: Version 1	338
The SHEETOFFSET function: Version 2	339
Returning the maximum value across all worksheets	339
Returning an array of nonduplicated random integers	340
Randomizing a range	342
Windows API Calls	344
Determining file associations	344
Determining default printer information	345
Determining the current video mode	346
Adding sound to your applications	347
Reading from and writing to the Registry	349

Part IV: Working with UserForms**353****Chapter 12: Custom Dialog Box Alternatives 355**

Before You Create That UserForm.	355
Using an Input Box	355
VBA's InputBox function	356
Excel's InputBox method	357
VBA's MsgBox Function	359
Excel's GetOpenFilename Method	364
Excel's GetSaveAsFilename Method	367
Prompting for a Directory	368
Using a Windows API function to select a directory	368
Using the FileDialog object to select a directory	370
Displaying Excel's Built-In Dialog Boxes	371
Using the Dialogs collection	371
Learning more about built-in dialog boxes	373
Using arguments with built-in dialog boxes	373
Executing a menu item directly	374

Chapter 13: Introducing UserForms 377

How Excel Handles Custom Dialog Boxes	377
Inserting a New UserForm	378
Displaying a UserForm	378
Adding Controls to a UserForm	379
Controls Available to You	380
CheckBox	380
ComboBox	380
CommandButton	380
Frame	380
Image control	380
Label	381
ListBox	381
MultiPage	381
OptionButton	381
RefEdit	381
ScrollBar	381
SpinButton control	382
TabStrip	382
TextBox	382
ToggleButton	382
Adjusting UserForm Controls	382
Adjusting a Control's Properties	385
Using the Properties window	385
Common properties	386
Learning more about properties	387
Accommodating keyboard users	387

Displaying and Closing UserForms	389
Displaying a UserForm	389
Closing a UserForm	389
About event-handler procedures	391
Creating a UserForm: An Example	391
Creating the UserForm	391
Writing code to display the dialog box	394
Trying it out	395
Adding event-handler procedures	396
Validating the data	398
Now it works	398
UserForm Events	398
Learning about events	399
UserForm events	399
Example: SpinButton events	400
Pairing a SpinButton with a TextBox	402
Referencing UserForm Controls	404
Customizing the Toolbox	406
Changing icons or tip text	406
Adding new pages	406
Customizing or combining controls	406
Adding new ActiveX controls	407
Creating UserForm “Templates”	408
A UserForm Checklist	408

Chapter 14: UserForm Examples 411

Creating a UserForm “Menu”	411
Using CommandButtons	411
Using a ListBox	412
Selecting Ranges	413
Creating a “Splash Screen”	414
Disabling a UserForm’s Close Button	416
Changing a Dialog Box’s Size	417
Zooming and Scrolling a Sheet from a UserForm	418
ListBox Techniques	420
About the ListBox control	421
Adding items to a ListBox control	421
Determining the selected item	425
Determining multiple selections	426
Multiple lists in a single ListBox	426
ListBox item transfer	427
Moving items in a ListBox	428
Working with multicolumn ListBox controls	430
Using a ListBox to select worksheet rows	432
Using a ListBox to activate to a sheet	434
Using the MultiPage Control	436

Chapter 15: Advanced UserForm Techniques 439

Displaying a Progress Indicator	439
Creating a standalone progress indicator	440
Showing progress using a MultiPage control	442
Showing progress without using a MultiPage control	445
Creating Wizards	445
Setting up the MultiPage control	446
Adding the buttons	447
Programming the buttons	447
Programming dependencies	449
Performing the task	450
Final steps	451
Emulating the MsgBox Function	452
MyMsgBox code	452
How it works	453
Using the MyMsgBox function	455
A Modeless Dialog Box	455
Multiple Buttons, One Event-Handler	458
Procedure	459
Adapting this technique	461
A Color Picker Dialog	461
Displaying a Chart in a UserForm	463
Method 1: Save the chart as a file	463
Method 2: Use the OWC ChartSpace control	464
Displaying a Spreadsheet in a UserForm	467
An Enhanced Data Form	471
Description	472
Installing the add-in	473
Using the Enhanced Data Form	473

Part V: Advanced Programming Techniques

475

Chapter 16: Developing Excel Utilities with VBA 477

About Excel Utilities	477
Using VBA to Develop Utilities	478
What Makes a Good Utility?	478
Text Tools: The Anatomy of a Utility	479
Background	479
Project goals for Text Tools	480
How it works	480
The Text Tools workbook	481
The FormMain UserForm	481
The modMain module	483
The ApplyButton_Click procedure	486
The “task” procedures	488
The undo technique	497
The ShowStats procedure	498

User help technique	500
Create menu and delete menu procedures	500
Evaluation of the project	501
Understand the Text Tools utility	502
More About Excel Utilities	502
Chapter 17: Working with Pivot Tables	505
An Introductory Example	505
Creating a pivot table	506
Examining the recorded code	508
Cleaning up the recorded code	508
Creating a More Complex Pivot Table	509
The data	509
The pivot table	510
The code that created the pivot table	511
How it works	513
Creating a Pivot Table from an External Database	513
Creating Multiple Pivot Tables	515
Modifying Pivot Tables	518
Chapter 18: Working with Charts	521
About Charts	521
Chart locations	521
The Chart object model	522
Recording Chart Macros	524
Macro recorder output	524
The “cleaned up” macro	526
Common VBA Charting Techniques	527
Activating a chart	527
Deactivating a chart	529
Determining whether a chart is activated	529
Deleting from ChartObjects or charts	530
Applying chart formatting	530
Looping through all charts	531
Sizing and aligning ChartObjects	532
More Charting Examples	533
Using names in a SERIES formula	534
Specifying the data used by a chart	535
Determining a chart’s source data: Method 1	538
Determining a chart’s source data: Method 2	543
Displaying arbitrary data labels on a chart	546
Displaying a chart in a UserForm	549
Understanding Chart Events	551
An example of using Chart events	552
Enabling events for an embedded chart	555
Example: Using Chart events with an embedded chart	557

Charting Tricks	559
Printing embedded charts on a full page	559
Creating a “dead chart”	560
Controlling a data series by hiding data	560
Storing multiple charts on a chart sheet	562
Using linked pictures in a chart	563
Animated charts	565
Creating a hypocycloid chart	566
Creating a “clock” chart	567
Drawing with an XY chart	569
Chapter 19: Understanding Excel’s Events	571
Event Types That Excel Can Monitor	572
What You Should Know about Events	572
Understanding event sequences	572
Where to put event-handler procedures	573
Disabling events	574
Entering event-handler code	576
Event-handler procedures that use arguments	577
Workbook-Level Events	578
The Open event	579
The Activate event	580
The SheetActivate event	580
The NewSheet event	581
The BeforeSave event	581
The Deactivate event	581
The BeforePrint event	582
The BeforeClose event	583
Worksheet Events	585
The Change event	585
Monitoring a specific range for changes	586
The SelectionChange event	590
The BeforeRightClick event	591
Chart Events	592
Application Events	592
Enabling Application-level events	595
Determining when a workbook is opened	595
Monitoring Application-level events	597
UserForm Events	598
Events Not Associated with an Object	599
The OnTime event	599
The OnKey event	600
Chapter 20: Interacting with Other Applications	603
Starting Another Application	603
Activating Another Application	607
Running Control Panel Dialog Boxes and Wizards	608

Automation	609
Working with foreign objects	609
Early versus late binding	610
A simple example	613
Controlling Word from Excel	613
Controlling Excel from another application	616
Working with ADO	619
Using SendKeys	621

Chapter 21: Creating and Using Add-Ins 625

What Is an Add-In?	625
Comparing an add-in to a standard workbook	625
Why create add-ins?	626
Understanding Excel's Add-In Manager	627
Creating an Add-In	628
An Add-In Example	629
Setting up the workbook	629
Testing the workbook	630
Adding descriptive information	630
Creating the add-in	631
Installing the add-in	632
Distributing the add-in	633
Modifying the add-in	633
Comparing XLA and XLS Files	635
File size and structure	635
Collection membership	635
Windows	635
Sheets	636
Accessing VBA procedures in an add-in	637
Manipulating Add-Ins with VBA	639
The AddIns collection	639
AddIn object properties	641
AddIn object events	644
Optimizing the Performance of Add-ins	644
Code speed	644
File size	645
Special Problems with Add-Ins	646
Ensuring that an add-in is installed	646
Referencing other files	648
Specifying the proper Excel version	649

Part VI: Developing Applications 651

Chapter 22: Creating Custom Toolbars 653

About Command Bars	653
Toolbar Manipulations	654

How Excel Handles Toolbars	654
Storing toolbars	654
When toolbars don't work correctly	655
Manipulating Toolbars and Buttons Manually	655
About command bar customization mode	656
Distributing toolbars	660
Manipulating the CommandBars Collection	661
Command bar types	662
Listing all CommandBar objects	662
Creating a command bar	664
Referring to command bars	665
Deleting a command bar	665
Properties of command bars	666
Referring to controls in a command bar	671
Listing the controls on a command bar	672
Listing all controls on all toolbars	672
Adding a control to a command bar	673
Deleting a control from a command bar	674
Properties of command bar controls	675

Chapter 23: Creating Custom Menus 685

A Few Words about Excel's Menu Bar	685
What You Can Do with Excel's Menus	686
Menu terminology	686
Removing menu elements	687
Adding menu elements	688
Changing menu elements	688
VBA Examples	689
Listing menu information	689
Adding a new menu to a menu bar	690
Deleting a menu from a menu bar	694
Adding menu items to a menu	694
Displaying a shortcut key with a menu item	698
Fixing a menu that has been reset	700
Working with Events	701
Adding and deleting menus automatically	701
Disabling or hiding menus	702
Working with checked menu items	703
The Easy Way to Create Custom Menus	706
Creating a Substitute Worksheet Menu Bar	708
Working with Shortcut Menus	710
Adding menu items to shortcut menus	712
Deleting menu items from shortcut menus	713
Disabling shortcut menu items	713
Disabling shortcut menus	714
Resetting shortcut menus	714
Creating new shortcut menus	714

Chapter 24: Providing Help for Your Applications 719

Help for Your Excel Applications?	719
Help Systems That Use Excel Components	721
Using cell comments for help	722
Using a Text Box for help	722
Using a worksheet to display help text	722
Displaying help in a UserForm	723
Using the Office Assistant to display help	727
Using the WinHelp and HTML Help Systems	730
About WinHelp	730
About HTML Help	731
Associating a Help File with Your Application	732
Other Ways of Displaying WinHelp or HTML Help	736
Using the Help method	736
Displaying Help from a message box	736
Displaying Help from an input box	737

Chapter 25: Developing User-Oriented Applications 739

What is a User-Oriented Application?	739
the Loan Amortization Wizard	739
Using the application	740
The workbook structure	742
How it works	743
Potential enhancements	748
Application Development Concepts	748
Some Final Words	748

Part VII: Other Topics

751

Chapter 26: Compatibility Issues 753

What Is Compatibility?	753
Types of Compatibility Problems	754
Excel File Formats Supported	755
Avoid Using New Features	756
Applications That Use Windows API Calls	757
But Will It Work on a Mac?	758
Creating an International Application	759
Multilanguage applications	761
VBA language considerations	762
Using “local” properties	762
Identifying system settings	763
Date and time settings	766

Chapter 27: Manipulating Files with VBA 767

Performing Common File Operations	767
VBA file-related commands	768
Using the FileSearch object	771
Locating files that contain specific text	773
Using the FileSystemObject object	773
Working with Text Files	775
Opening a text file	776
Reading a text file	777
Writing a text file	777
Getting a file number	777
Determining or setting the file position	778
Statements for reading and writing	778
Text File Manipulation Examples	779
Importing data in a text file	779
Exporting a range to a text file	779
Importing a text file to a range	781
Logging Excel usage	782
Filtering a text file	783
Importing more than 256 columns of data	783
Exporting a range to HTML format	786

Chapter 28: Manipulating Visual Basic Components 789

Introducing the IDE	789
The IDE Object Model	791
The VBProjects collection	791
An Introductory Example	794
Replacing a Module with an Updated Version	795
Using VBA to Write VBA Code	797
Adding Controls to a UserForm at Design Time	800
Design-time versus runtime UserForm manipulations	800
Adding 100 CommandButtons at design time	802
Creating UserForms Programmatically	804
A simple example	804
A useful (but not so simple) example	806

Chapter 29: Understanding Class Modules 811

What is a Class Module?	811
Example: Creating a NumLock Class	812
Inserting a class module	813
Adding the VBA code	813
Using the NumLock class	814
More about Class Modules	815
Naming the object class	815
Programming properties	816
Programming methods	817
Class module events	818

Example: A CSV File Class	818
Class module-level variables	819
Property procedures	819
Method procedures	820
Using the CSVFileClass object	822
Chapter 30: Frequently Asked Questions	
about Excel Programming	825
General Excel Questions	826
The Visual Basic Editor	830
Procedures	834
Functions	839
Objects, Properties, Methods, and Events	842
UserForms	851
Add-Ins	856
CommandBars	858
Appendix A: Excel Resources Online	861
Appendix B: VBA Statements and Functions Reference	867
Appendix C: VBA Error Codes	877
Appendix D: ANSI Code Reference	881
Appendix E: What's on the CD-ROM	891
Index	907
End-User License Agreement	943
CD-ROM Installation Instructions.	946