VBA and Macros for Microsoft Excel

Copyright© 2004 by Que Publishing

International Standard Book Number: 0789731290

Warning and Disclaimer

When reviewing corrections, always check the print number of your book. Corrections are made to printed books with each subsequent printing. To determine the print number of your book, view the copyright page. The print number is the right-most number on the line below the "First Printing" line. For example, the following indicates that this is the 1st printing of this title and it was printed in May 2003.

First Printing: May 2003

06 05 04 03 4 3 2 1

First and Second Printing Corrections

Pg	**Error**	Correction
48	Third paragraph, sixth sentence: In Figure 2.33, Visual Basic just selected ~~Row 1~~ (making A1 the ActiveCell).	In Figure 2.33, Visual Basic just selected **all cells** (making A1 the ActiveCell).

78	Replace code at bottom of page that continues to the top of page 79	With:
		`Function SheetExists(SName As String, Optional WB As Workbook) As Boolean`
		`    Dim WS As Worksheet`
		`    ' Use active workbook by default`
		`    If WB Is Nothing Then`
		`        Set WB = ActiveWorkbook`
		`    End If`
		
		`    On Error Resume Next`
		`        SheetExists = CBool(Not WB.Sheets(SName) Is Nothing)`
		`    On Error GoTo 0`
		
		`End Function`

93	Code line after first sentence under "Retrieve Numbers from Mixed Text": RetrieveNumbers (Rng)	Code line after first sentence under "Retrieve Numbers from Mixed Text": RetrieveNumbers (MyString)
95	Third line of code at top of page: arr_str = Split(str, chr)	arr_str = Split(str, chr) 'Not compatible with XL97
95	14th line of code from the bottom of page: concat = 0#	Sortconcat = 0#

96	7th and 13th lines of code on the page:
	concat = Left(MySum, Len (MySum) - 2)
	concat = Err.Number & " - " & Err.Description
		Sortconcat = Left(MySum, Len (MySum) - 2)
		Sortconcat = Err.Number & " - " & Err.Description

106	Third line of code from the bottom of page: Cells(I, 1).resize(1, 8).Copy Destination:=Cells(NextRow, i)	Cells(I, 1).Resize(1, 8).Copy Destination:=Cells(NextRow, 1)
107	First line of code on page: ' Delete all rows where column C is the Internal rep - C54	' Delete all rows where column C is the Internal rep - S54
111	Seventh line from bottom of page: If Not (Data, 5) - "Total" then	If Not Left (Data, 5) - "Total" then
112	Fourth and fifth row of code block at bottom of page:	Should read: ' Apply the check to the oldest open invoices and Decrement AmtToApply
112	Add the following line of code as the next to last line of code at bottom of page and align it with third to last line:	NextRow = NextRow + 1
121	Case Study, fifth line of code: Range("D4").Formula = "=B2*C2"	Range("D4").Formula = "=B4*C4"
135	Line of code at top of page: Names.Add = "Company", Refersto:="CompanyA"	Names.Add Name:= "Company", RefersTo:="CompanyA"
140	Ninth line of code: NextRow = WSM.Range("A1").End(xlUp).Row + 1	NextRow = WSM.Range("A65536").End(xlUp).Row + 1
140	Add this line of code right before End If and align the new line with the line before it:	NextRow = NextRow + 1
140	Sixth line of code from the bottom: WSD.Cells("B1").EntireColumn.Insert	WSD.Range("B1").EntireColumn.Insert
149	Delete third to last line of code at bottom of page	

154	First line of code on page: Private Sub Chart_BeforeDoubleClick(byVal ElementID as Long, _	Private Sub MyChartClass_BeforeDoubleClick(byVal ElementID as Long, _
155	First two lines of code at top of page: Private Sub Chart_MouseDown(ByVal Button As Long, ByVal Shift As Long, _ ByVal x As Long, By Val y As Long)	Private Sub MyChartClass_MouseDown(ByVal Button As Long, ByVal Shift _ As Long, ByVal x As Long, By Val y As Long)
155	First two lines of code at bottom of page: Private Sub Chart_Select(ByVal ElementID As Long, ByVal Arg1 As Long, _ ByVal Arg2 As Long)	Private Sub Chart_Select(ByVal ElementID As Long, ByVal Arg1 _ As Long, ByVal Arg2 As Long)
155	Combine last two lines of code on the page:	To read: Sheets ("sheet1").Cells.Interior.ColorIndex = xlNone

156	Replace code at top of page:	With:
		`If ElementID = 3 Then`
		`If Arg2 = -1 Then`
		`' Selected the entire series in Arg1`
		`Sheets("Sheet1").Range("A2:A22").Offset(0, Arg1).Interior.ColorIndex = 19`
		`Else`
		`' Selected a single point in range Arg1, Point Arg2`
		`Sheets("Sheet1").Range("A1").Offset(Arg2, Arg1).Interior.ColorIndex = 19`
		`End If`
		`End If`
		End Sub

161	Replace code in middle of page:	With:
		`AveMos = InputBox(Prompt:="Enter the number " & _`
		`" of months to average", Title:="Enter Months", _`
		`Default:="3")`

200	Second to last line of code on page: ActiveChart.PieGroups(1).FirstSliceAngle = 100	Cht.PieGroups(1).FirstSliceAngle = 100
202	Second to last line of code on page: Cht.Export FileName:="C:/MyChart.gif", FilterName:="GIF"	Cht.Export FileName:="C:\MyChart.gif", FilterName:="GIF"

213	Code near middle of page:
	Beginning with the line With Me.lbCust and ending with End With, replace this section
		With:
		`With Me.lbCust`
		`    .RowSource = “”`
		`    .List = Cells(2, NextCol).Resize(LastRow - 1, 1).Value`
		`End With`

241	Middle of page, third sentence: For example, try to run a macro that would delete Column ~~0~~.	For example, try to run a macro that would delete Column **O**.
253	Fifth line of code in middle of page: With Range("A3").Resize(FinalReportRow - 2, 1)	With WSR.Range("A3").Resize(FinalReportRow - 2, 1)

364	Replace code on the page:	With:

```
Sub IsWordOpen()
    Dim wdApp As Object
    ActiveChart.ChartArea.Copy
    On Error Resume Next
    Set wdApp = GetObject(, "Word.Application")
    If wdApp Is Nothing Then
        Set wdApp = GetObject("", "Word.Application")
    End If
    If wdApp.ActiveDocument Is Nothing Then
        With wdApp
            .Documents.Add
            .Visible = True
        End With
    End If
    On Error GoTo 0
    With wdApp.Selection
        .EndKey Unit:=6 ' wdStory=6
        .TypeParagraph
        ' wdPasteOLEObject=0, wdInLine=0 so this works
        .PasteSpecial link:=False, DataType:=wdPasteOLEObject, _
            Placement:=wdInLine, DisplayAsIcon:=False
    End With
    Set wdApp = Nothing
End Sub
```

390	Delete first line of code at bottom of page	

419	Code block in middle of page, fourth and fifth lines: ActiveChart.Axes (xlValue).MaximumScale = _ ActiveChart.Axes(xlValue).MaximumScale - 50	XLChart.Axes (xlValue).MaximumScale = _ XLChart.Axes(xlValue).MaximumScale - 50
419	Code block in middle of page, eighth and ninth lines: ActiveChart.Axes(xlValue).MaximumScale = _ ActiveChart.Axes(xlValue).MaximumScale + 50	XLChart.Axes(xlValue).MaximumScale = _ XLChart.Axes(xlValue).MaximumScale + 50

425	Fourth paragraph, first sentence: Insert a new ~~module~~ for the collection and rename it clsEmployees.	Insert a new **class** module for the collection and rename it clsEmployees.
428	Delete second line of code after Figure 20.12	

436	Toggle Buttons section, first paragraph and first two sentences of second paragraph: Toggle ~~Buttons~~ ~~Toggle~~ buttons are similar to check boxes in that they can be used to select buttons. But, unlike check boxes, ~~toggle~~ buttons can be easily configured to allow only one selection out of a group (see Figure 21.3). A program can be used to allow only one item to be selected in a group, but ~~toggle~~ buttons have a property, Groupname, that does this automatically. ~~Toggle~~ buttons with the same GroupName are automatcially grouped together.	**Option Buttons** **Option** buttons are similar to check boxes in that they can be used to select buttons. But, unlike check boxes, **Option** buttons can be easily configured to allow only one selection out of a group (see Figure 21.3). A program can be used to allow only one item to be selected in a group, but **Option** buttons have a property, Groupname, that does this automatically. **Option** buttons with the same GroupName are automatcially grouped together.

436	Figure 21.3 caption: ~~Toggle~~ buttons allow only one selection out of a group.	**Option** buttons allow only one selection out of a group.

First, Second, and Third Printing Corrections

6	Insert new text before "Special Elements and Typographical Conventions" with D head.	**Versions** For the most part, code in this book will work in Excel 97 through Excel 2003 for the Windows version of Excel. The Pivot Table code in Chapter 12 will generally work with Excel 2000 and newer, although the case study at the end of the chapter discusses how to adapt the previous examples for Excel 97. The XML examples in Chapter 15 will only work with Excel 2003 or newer. While Excel for Windows and Excel for the Mac are similar in their user interface, there are a number of differences when you compare the VBA environment. Certainly, nothing in Chapter 22 that uses the Windows API will work on the Mac. The overall concepts discussed in the book will apply to the Mac, but differences will exist. A general list of differences as they apply to the Mac can be found at http://www.mrexcel.com/macvba.html.

59	Insert new text before "Next Steps" with D Head.	**Speeding Up Code**
		
		It is fun to sit back and watch a macro do actions without you touching the keyboard. However, if you have a macro that is running for 30 seconds or more, you can dramatically speed up the run time by having Excel not redraw the screen while the macro is running. Add this line to the start of your macro:
		
		Application.ScreenUpdating = False
		
		If you want to keep track of your macro’s progress, you can display messages to the Excel status bar with code like this:
		
		Application.StatusBar = “Processing Sheet1”
		
		At the end of the macro, remember to turn screen updating back to normal and to clear the status bar so that Excel can show normal messages like “Calculating”, “Saving”, and “Ready”.
		
		Application.ScreenUpdating = True
		
		Application.StatusBar = False
		
		In a workbook with thirty worksheets, the first code runs in 19 seconds and the second macro runs in 5 seconds, a 74% speed improvement!
		
		`Sub RegularCode()`
		`    Debug.Print Time`
		`    For Each ws In ThisWorkbook.Worksheets`
		`        ws.Select`
		`        For Each cell In Range("A1:Q31")`

		`            cell.Value = Time`
		`        Next cell`
		`    Next ws`
		`    Debug.Print Time`
		`    ' Hit Ctrl+G in VBA to see the printed times`
		`End Sub`
		
		`Sub FasterCode()`
		`    Debug.Print Time`
		`    Application.ScreenUpdating = False`
		`    For Each ws In ThisWorkbook.Worksheets`
		`        Application.StatusBar = "Processing " & ws.Name`
		`        ws.Select`
		`        For Each cell In Range("A1:Q31")`
		`            cell.Value = Time`
		`        Next cell`
		`    Next ws`
		`    Application.ScreenUpdating = True`
		`    Application.StatusBar = False`
		`    Debug.Print Time`
		`End Sub`

112	Last section of code, second line: AmtToApply = InputBox("Enter Amount of Check")	AmtToApply = InputBox("Enter Amount of Check") **+ 0**
172	Insert new text before "Next Steps" with C head.	**Getting a File Name** One of the most common client interactions is when you need the client to specify a path and

filename. Excel VBA has a built in function to display the file Open dialog box. The client browses to and selects a file. When the client chooses the Open button, Excel VBA does not open the file, but instead returns the selected file to you.

```
Sub SelectFile()
    ' Ask which file to copy
    x = Application.GetOpenFilename( _
        FileFilter:="Excel Files (*.xls), *.xls", _
        Title:="Choose File to Copy", MultiSelect:=False)
    MsgBox "You selected " & x
End Sub
```

The above code will allow the client to select one file. If you want them to specify multiple files, use this code:

```
Sub ManyFiles()
    Dim x As Variant
    x = Application.GetOpenFilename( _
        FileFilter:="Excel Files (*.xls), *.xls", _
        Title:="Choose Files", MultiSelect:=True)
    For i = 1 To UBound(x)
        MsgBox "You selected " & x(i)
    Next i
End Sub
```

In a similar fashion, you can use Application.GetSaveAsFileName to find the path and filename that should be used for saving a file.

249	Delete the paragraph that starts "Keep in mind there is absolutely…" and the four lines of code that follow. Substitute new text with C head.	A different technique is to turn on the first subtotal. This will automatically turn off the other 11 subtotals. You can then turn off the first subtotal to make sure that all subtotals are suppressed:
		`PT.PivotFields("Line of Business").Subtotals(1) = True`
		`PT.PivotFields("Line of Business").Subtotals(1) = False`