

基于 VBA 的 Excel 上下文敏感菜单的设计

Design for VBA-baesd Excel Context-sensitive Menu

彭江平 (湖南大学会计学院信息管理系 410079)

摘要: 本文讨论了在Excel中,将用户自定义菜单与工具栏设计成上下文敏感的必要性,给出了基于VBA与Excel事件模型设计上下文敏感菜单的实现方法,并以具体实例进行了说明。

关键词: 上下文敏感菜单 Excel VBA 事件模型

1 引言

通常情况下,Excel系统的菜单能依据当前单元格状态、Excel的编辑模式等上下文的内容自动地修改它的状态,这可以在很大程度上避免用户的错误操作。使用Office中嵌入的编程环境VBA,可以设计用户自定义的菜单与工具栏,但是自定义的菜单与工具栏的状态表面上总是可用的,而事实上,其可用性会因为Excel的状态、单元格状态等上下文信息而发生改变,如在单元格的数据输入与编辑模式下,所有用户自定义的菜单与工具栏都是不可用的。这种可用性与菜单或按钮的状态的不一致性给用户操作带来了许多不便,使得用户将许多时间浪费在仅能获得诸如“该功能在当前状态不可使用”,“请选择其他类型的单元格”等无用信息的操作中。

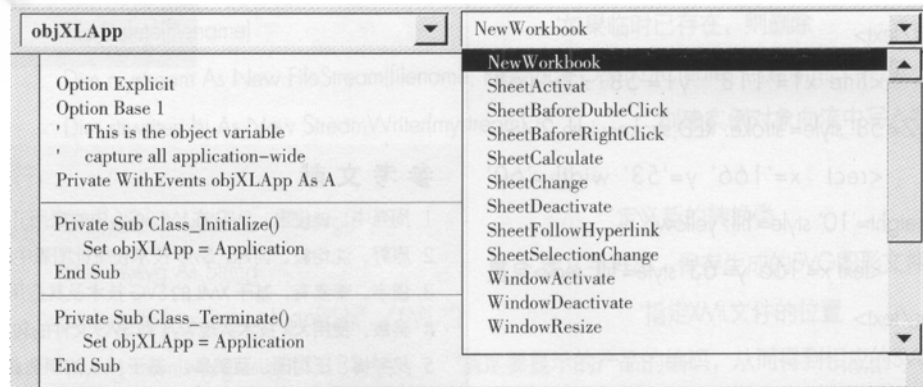
2 上下文敏感菜单与工具栏的设计

设计上下文敏感的菜单或工具栏,关键是要在Excel的不同操作所产生的事件中,对相应的菜单与工具栏进行必要的设置。虽然在Office 97以前的版本的Excel环境中,就提供了内嵌的编程环境VBA (Visual Basic Application),使得使用者能依据需要,设计各种各样的宏,方便使用并简化需要重复进

行的工作,但是,只有在Office 97及以后的版本中,才提供了事务处理的能力,因此,本文给出的方法,仅适用于Office 97及以后的版本。

在Microsoft Excel中,可以编写工作表级、图表级、查询表级、工作簿级或应用程序级的事件处理过程。工作表、图表工作表和工作簿事件处理过程对任意打开的工作表或工作簿都有效;这时只需要进入VBA的编程环境,就可以直接在这些事件中输入相应的代码,以实现上下文敏感的菜单或工具栏,但是这种菜单与工具栏仅是针对特定的工作表、图表工作表和工作簿对象。嵌入式图表、查询表或“Application”对象编写事件处理过程,则必须在类模块中用“WithEvents”关键字创建新的类模块,并在类模块中与特定的对象关联起来。下面以“Application”级的事件处理为例进行说明。

在使用“Application”对象的事件之前,必须新建一个类并定义一个带有事件声明的Application类型对象。如在应用程序中新建了一个“clsEvents”类模块,在该模块中包含下列代码:“Public WithEvents App As Application”。在完该对象的事件声明之后,该对象将出现在类模块的“对象”下拉式列表框中,并可以为此对象编写相应的事件过程,如图1。在“对象”框中选取该新对象时,“过程”下拉式列表框中就会自动显示该对象的有效事件。



EMBED PBrush

图1 对象及事件列表

在图1左边的对象列表中选择对象"Class",并在图1右边的类的事件中选择类的初始化事件"Initialize",并输入相应的代码,将事件对象与Excel Application对象链接起来,即:"Set objXLApp = Application";而在类的终止事件"Terminate"中将事件对象释放,即:"Set objXLApp = Nothing"。在应用类模块之前,必须定义与类模块对象并连接到"Application"对象,即在应用模块中使用语句:"Public EventClass As clsEvents"申明事件对象,并在宏打开时自动执行的例程"Auto_Open"中使用"Set EventClass = New clsEvents"与"Application"对象关联,而在宏关闭时自动执行的例程"Auto_close"中使用语句"Set EventClass = Nothing"释放对象。

3 上下文敏感菜单与工具栏的实现

下面以基于应用级的事件处理与VBA来实现上下文敏感菜单为例,说明具体的实现过程。该实例首先定义了一个宏,它能将Excel中含有公式的单元格中的公式转换为相应的值,而且将实现的宏与主菜单中的"编辑菜单"、单元格右键菜单、行右键菜单、列右键菜单中的自定义菜单"公式转换为值"相对应。而且这些菜单能依据单元格本身或指定行或列中是否含有包含公式的单元格来动态地确定其可用状态。

程序在Office 97与Office 2000环境下调试通过。代码主要由两大模块组成,第一部分是类模块,而另一部分是应用类的模块。下面给出类模块的程序代码,应用类模块因篇幅较长,略去。

```
'第一部分:类模块 clsEvents
Option Explicit*
Option Base 1
Private WithEvents objXLApp As Application
'建立事件处理类
Private Sub Class_Initialize() '类的初始化
    Set objXLApp = Application
End Sub
```

```
Private Sub Class_Terminate() '类的终止事件
    Set objXLApp = Nothing
End Sub
Private Sub objXLApp_Sheet
BeforeRightClick(ByVal wshSheet As Object, _
    ByVal rngTarget As Excel.Range,
    blnCancel As Boolean) '右键菜单处理
    Dim cbrMenuBar As CommandBar
    Dim cbiButton As CommandBarButton
    Dim blnCellsContain As Boolean
    Dim lngMenuCount As Long
    Dim rngFormulae As Range
    Application.EnableEvents = False '为
防止事件循环,暂时关闭事件处理
    Set rngFormulae = fFormulaeRange
(rngTarget)
    Application.EnableEvents = True
    blnCellsContain = Not rngFormulae Is
Nothing
    For lngMenuCount = 1 To 3
        Select Case lngMenuCount
            Case 1
                Set cbrMenuBar = Application.
CommandBars("Cell")
            Case 2
                Set cbrMenuBar = Application.
CommandBars("Row")
            Case 3
                Set cbrMenuBar = Application.
CommandBars("Column")
        End Select
        Set cbiButton = cbrMenuBar.
FindControl(Tag:=MENU_ITEM_TAG)
        If Not cbiButton Is Nothing Then
            cbiButton.Enabled = blnCellsContain
        Next
    End Sub
'下面所有的事件都会引起单元格的改变
Private Sub objXLApp_NewWorkbook
(ByVal Wb As Excel.Workbook)
```

```
MenuToggle
End Sub
Private Sub objXLApp_SheetActivate(ByVal
Sh As Object)
    MenuToggle
End Sub
Private Sub objXLApp_SheetSelection
Change(ByVal Sh As Object, _
    ByVal Target As
Excel.Range)
    MenuToggle
End Sub
Private Sub objXLApp_WindowActivate
(ByVal Wb As Excel.Workbook, _
    ByVal Wn As
Excel.Window)
    MenuToggle
End Sub
Private Sub objXLApp_WorkbookActivate
(ByVal Wb As Excel.Workbook)
    MenuToggle
End Sub
Private Sub objXLApp_Workbook
NewSheet(ByVal Wb As Excel.Workbook, _
    ByVal Sh As
Object)
    MenuToggle
End Sub
Private Sub objXLApp_WorkbookOpen
(ByVal Wb As Excel.Workbook)
    MenuToggle
End Sub
Public Sub MenuToggle() '菜单的动态设
置
    Dim winWindow As Window
    Dim shtSheet As Worksheet
    Dim blnWorkBookVisible As Boolean
    Dim blnWorkSheetActive As Boolean
```

下转第58页 >>

```

Dim cbiButton As rngFormulae Is Nothing
CommandBarButton Application.EnableEvents = True
Dim blnCellsContain As Boolean End If
Dim rngFormulae As Range End If
blnCellsContain = False Set cbiButton = 'CommandBars
For Each winWindow In Windows ("Worksheet Menu Bar"). _
    If winWindow.Visible Then FindControl(Tag:
        blnWorkBookVisible = True =MENU_ITEM_TAG, recursive:=True)
    Exit For If Not cbiButton Is Nothing Then
End If cbiButton.Enabled = blnCellsContain
Next End Sub
If blnWorkBookVisible Then
    If TypeName(Selection) = "Range"
Then
        Application.EnableEvents = False
        Set rngFormulae =
fFormulaeRange(Selection)
        blnCellsContain = Not

```

4 结论及需要进一步做的工作

借助Excel的事件模型及VBA编程环境，实现了上下文敏感的菜单与工具栏，这对设计用户友好的Excel VBA应用程序具有一定的指导意义，而且该方法也可以推广到Office中的其他工具的中类似功能的实现。

但是，细心的读者通过实践也许会发现，基于Excel的事件模型与VBA实现的上下文敏感菜单，在使用鼠标操作时，的确功能完善，但如果使用键盘，则有些功能就不那么有效了，这是因为在Excel的事件模型中，有大量与鼠标操作有关的事件，但却没有与键盘操作有关的事件，因而我们也就不能对这些事件进行处理。

参考文献

- 1 晶辰工作室编著, Excel2000 VBA开发实例指南, 电子工业出版社, 2000. 8.
- 2 刘曜, 在 Excel 上设置自定义函数定义报表, 计算机系统应用, 2002. 6, P71-73.
- 3 彭江平, 基于 Excel 与 VBA 的通用报表系统的设计, 计算机系统应用, 2002. 12.