

第9章 Excel VBA 对象模型和应用程序对象

本章首先为读者简要介绍 Excel 2007 VBA 对象模型,随后则讲述 Application 对象的应用。本章以及本篇之后介绍对象的章节中,在讲解时都采用示例讲解的方式。每个对象都将属性与方法分开讲述。另外限于篇幅,本篇中介绍各个对象的属性与方法时,将主要针对比较常用的对象属性与方法加以介绍。在讲述 Excel 2007 VBA 对象模型时,并不涉及对象的事件。

9.1 Excel VBA 对象模型简介

从面向对象的观点来看,Excel 2007 VBA 对象模型应该是对 Excel 2007 应用程序的直观模拟,实质上也确实如此,这句话主要包含以下 3 层含义。

- Excel 2007 VBA 对象模型对于 Excel 2007 程序的模拟是基于直观视觉的。
- 用户可以通过 Excel 2007 VBA 对象模型来控制 Excel 2007 应用程序。
- 用户无法通过 Excel 2007 VBA 对象模型做到完全控制 Excel 2007 程序。

关于第 3 点,将在介绍定制 Excel 2007 界面以及 API 章节加以说明。从这里可以看到 Excel 2007 VBA 对象模型不是万能的。

9.1.1 认识 Excel 2007

前面章节中,用户从界面操作角度了解了 Excel 2007,本小节则使用面向对象的思考方法。Excel 2007 VBA 对象模型的基本框架不是以操作为模拟依据的。本节首先通过一个示例来初步了解 Excel 2007 VBA 对象模型中最为重要的构成,这几个基本构成了 Excel 2007 VBA 对象模型的基本框架。

本示例打开 6 个工作簿文件,然后将这些文件平铺显示到 Excel 2007 应用程序界面中。当完成了这些操作后,示例以获得的结果对 Excel 2007 VBA 对象基本架构的几个重要对象加以说明。操作步骤如下。

- (1) 启动 Excel 2007 程序,然后打开 6 个工作簿文件。
- (2) 选择菜单栏中的“视图”→“窗口”→“全部重排”命令。
- (3) 在弹出的“重排窗口”对话框中,点选“平铺”单选钮,然后单击“确定”按钮,如图 9-1 所示。当完成重排后,可以获得如图 9-2 所示的效果。

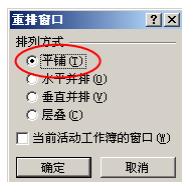


图 9-1 “重排窗口”对话框

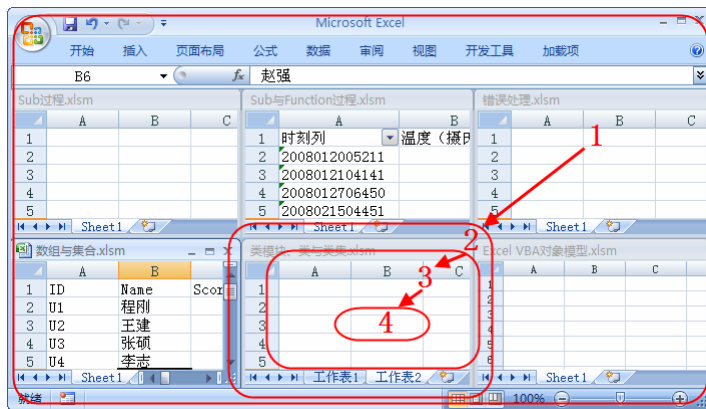


图 9-2 重排工作簿窗口结果

在图 9-2 中，显示了 6 个工作簿文件，这 6 个工作簿文件被平铺显示在 Excel 2007 的客户端窗格中。每一个窗格都划分了一个工作簿，每一个工作簿文件都包含几个工作表，每一个工作表中又包含了若干单元格。图中已经标识出了这些包含关系。其中 1 号区域对应 Excel 2007 应用程序，2 号圆形框对应工作簿，3 号圆形框对应工作表，4 号对应单元格对象。

说明：从图中不难看出 Excel 2007 VBA 对象模型基本框架。该基本框架一共包含了 4 个对象，Application 对象、Workbook 对象、Worksheet 对象以及 Range 对象。其中 Workbook 对象、Worksheet 对象应该还有该对象的类集即 Workbooks、Worksheets 以及 Ranges 集合。

9.1.2 深入了解 Excel 2007 VBA 对象模型

仅仅了解 Excel 2007 VBA 对象模型的基本框架对象是远远不够的。虽然这 4 个对象可能在实际应用中占用了大量的代码篇幅，但是依然不能解决所有开发中碰到的实际问题，甚至于还有部分问题是这几个对象不可能解决的。读者应该深入了解 Excel 2007 VBA 对象模型，以扩展视野，对 Excel 2007 的认识更上一层楼。Excel 2007 对象模型中包含的对象与集合非常多，这里也不可能将所有的对象列出，只对界面操作中使用到的以及部分较为实用的对象加以介绍。

读者可以通过录制相关的宏简单直接地获取这些对象属性与方法的应用，如自定义视图 CustomView 对象及其集合 CustomViews 对象、页面设置 PageSetup 对象等。以下为读者简要介绍 Excel 2007 VBA 对象以及方法。

- CustomView 对象及集合 CustomViews：CustomView 对象代表自定义工作簿视图；CustomViews 为自定义工作簿视图对象的集合。该对象对应 Excel 2007 中“视图”选项卡下的“自定义视图”命令。读者需要注意的是该对象只是代表自定义的工作簿视图，而未包含系统定义的视图。系统定义视图需通过 Window 对象的 View 属性访问。
- PageSetup 对象：PageSetup 对象代表了当前的页面设置。该对象对应 Excel 2007 中“页面布局”选项卡下的内容。

- ❑ **FormatCondition 对象：**FormatCondition 对象代表条件格式。该对象对应 Excel 2007 中“开始”选项卡的“样式”选项组中的“条件格式”命令。
- ❑ **Shape 对象及 Shapes 集合：**Shape 对象代表绘图层中的形状，包括自选图形、任意多边形、OLE 对象或图片；Shapes 则是绘图层中所有形状的集合。这些形状可以通过插入选项卡中插图组、文本组中的命令插入。
- ❑ **Hyperlink 对象及 Hyperlinks 集合：**Hyperlink 对象代表一个超级链接；HyperLinks 则代表了 Excel 2007 中所有超级链接对象构成的集合。
- ❑ **Chart 对象及 Charts 集合：**Chart 对象代表插入的某个图表；Charts 则代表所有插入到当前对象中的所有图表。这些对象可以是 Excel 应用程序、工作簿或工作表。该对象对应 Excel 2007 中“插入”选项卡的“图表”选项组中的命令。
- ❑ **Sort 对象：**Sort 对象代表数据区域的排序方式。该对象对应 Excel 2007 中“数据”选项卡的“排序与筛选”选项组中的“排序”命令。
- ❑ **AutoFilter 对象：**AutoFilter 对象代表对指定工作表的自动筛选。该对象对应 Excel 2007 中“数据”选项卡的“排序与筛选”选项组中的“筛选”命令。
- ❑ **AdvancedFilter 方法：**AdvancedFilter 实际上不构成一个对象，它只是 Range 对象的一个方法。这里为了和第 1 章的介绍对应起来，仍然将其列出来。该方法对指定区域按条件进行高级筛选。它对应 Excel 2007 中“数据：选项卡的”排序与筛选：选项组中的“高级筛选”命令。
- ❑ **SubTotal 方法：**SubTotal 和 AdvanceFilter 一样，也是 Range 对象的一个方法。其用于对指定区域或当前区域进行分类汇总操作。它对应 Excel 2007 中“数据”选项卡的“分级显示”选项组中的“分类汇”总命令。
- ❑ **Validation 对象：**Validation 对象代表工作表区域的数据有效性规则。它对应 Excel 2007 中“数据”选项卡的“数据”选项组中的“数据有效性”命令。
- ❑ **PivotTable 对象：**PivotTable 对象代表工作表中的数据透视表。它对应 Excel 2007 中“插入”选项卡的表选项组中的“数据透视表”命令。
- ❑ **Window 对象及 Windows 集合：**Window 对象代表 Excel 2007 中的某个窗口；Windows 则代表 Excel 2007 中各个窗口对象构成的集合。它对应 Excel 2007 中的“视图”选项卡中的“窗口”选项组。
- ❑ **Dialog 对象及 Dialogs 集合：**Dialog 对象代表 Excel 2007 中某个系统对话框；Dialogs 则代表 Excel 2007 中所有系统对话框构成的集合。该对话框对象在 Excel 2007 中没有固定的选项卡命令可以对应。几乎所有的在 Excel 2007 中可以打开的对话框都可以通过该对象访问并显示，如打开文件对话框、查找对话框、单元格设置对话框等。
- ❑ **CommandBar 对象及 CommandBars 集合：**CommandBar 对象代表 Excel 2007 中一个命令栏对象；CommandBars 则代表了 Excel 2007 中所有命令栏构成的集合。该对象对于在 Excel 2007 之前的版本建立菜单十分有效，但是在 Excel 2007 中使用时存在局限。

9.2 Application 对象

Application 应用程序对象代表了 Excel 2007 应用程序，它是 VBA 对象模型基本框架的顶层对象。该对象包含了十分丰富的属性、方法与事件，通过这些属性以及方法可以完成众多有关 Excel 2007 的设置、获取 Excel 程序当前的状态、控制 Excel 2007 交互等。Application 对象犹如 Excel 2007 VBA 对象模型中的根对象。

Application 对象的属性十分众多，不仅可以从中获取 Excel 2007 的状态信息，如 Visible 属性，也可以设置部分属性来控制 Excel 2007 的交互方式，如 DisplayAlerts 属性。用户还可以直接从该对象获取子对象或集合对象，如 ActiveWorkbook 对象、Workbooks 集合对象等。本节将只为读者讲述其中几类较为重要的属性，包括活动对象、显示状态、交互设置、子对象及集合 4 类。

Application 对象的方法与其属性一样数目众多，本节将抽出 Application 对象中 4 个非常典型的方法为读者加以介绍。这些方法分别是 GetOpenFilename、GetSaveAsFilename、MacroOptions、OnKey 和 SendKeys。

9.2.1 获取当前活动对象

Application 对象包含了 6 个获取当前活动对象的属性。这些属性都是只读属性，它们分别是 ActiveCell、ActiveChart、ActivePrinter、ActiveSheet、ActiveWindow 和 ActiveWorkbook 对象，这些活动对象之间还存在部分联系。如，ActiveCell 对象一定位于 ActiveSheet 内，ActiveSheet 对象一定位于 ActiveWorkbook 对象内。如果当前 Excel 2007 没有相应的活动对象，则该属性返回 Nothing。

从属性中返回的活动对象，其对象的类型由该活动对象本身决定。例如，ActiveCell 为 Range 对象等。因而从中可以看出，这些活动对象可以被看成相应类型的对象。相同类型对象的属性与方法同样适用于该对象。

这里需要再次提醒读者注意的是，在认识活动单元格 ActiveCell 时，需要理解一个现象，即活动单元格为选定区域内部的一个单元格。而选定区域可以包含多个单元格，但是始终只有一个单元格为活动单元格。当用户按<F2>键时，只有活动单元格处于编辑状态。

本示例过程通过代码将工作表 Sheet1 中的 B3~F6 单元格区域选中，然后通过 ActiveCell 对象获取该活动单元格区域包含的单元格数量以及其地址。本示例主要为了说明当有多个单元格被选中时，只有一个单元格为活动单元格。

(1) 新建工作簿，进入 VBE，添加新的模块，然后在该模块的代码窗口中输入如下代码。

```
Sub ActiveCellCount()  
Worksheets(1).Range("B3:F6").Select  
MsgBox "活动单元单元格个数: " & ActiveCell.Count & _  
    "个" & vbCrLf & "活动单元格地址: " & _  
    Replace(ActiveCell.Address, "$", "")  
End Sub
```



图 9-3 获取活动单元格
个数及地址

(2) 运行程序代码, 得到的结果如图 9-3 所示。

示例过程中, 首先通过单元格区域的 `Select` 方法将 B3 到 F6 间的单元格选中, 然后通过活动单元格的 `Count` 属性以及 `Address` 属性分别获取活动单元格区域的单元格个数以及地址。过程中为了避免出现单元格绝对引用标记符, 使用了 `Replace` 函数将获取的地址字符串中的 `$` 符号去除。

9.2.2 获取对象的关联信息

在 VBE 中, 用户可以通过程序代码获取各 Excel 对象之间的隶属关系。在本小节中, 将利用一个典型例子来说明这种方法。

本示例中将活动单元格的父对象与活动工作表进行比较, 将活动工作表的父对象与活动工作簿比较, 通过比较说明活动对象之间存在的包含关系。

```
Sub testrelate()  
    MsgBox "活动单元格位于活动工作表内?" & (ActiveSheet Is ActiveCell.Parent)  
    MsgBox "活动工作表位于活动工作簿中?" & (ActiveWorkbook Is ActiveSheet.Parent)  
End Sub
```

运行程序代码, 得到的结果如图 9-4 和图 9-5 所示。

通过两个信息框的内容来看, 可以知道活动工作簿、活动工作表以及活动单元格之间的包含关系。由于在过程中需要对两个对象进行比较, 所以使用了 `Is` 关键字。另外读者需要注意的是必须将比较表达式使用括号括起来, 否则会引发如图 9-6 所示的错误。

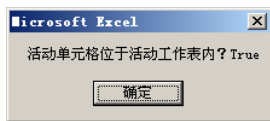


图 9-4 第一层关系的判断



图 9-5 第二层关系的判断

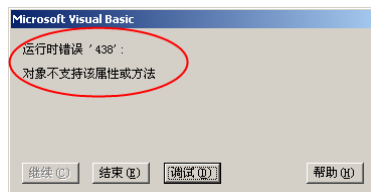


图 9-6 对象不支持该属性与方法错误

提示: 出现上面结果的原因十分简单, 连接运算符要比比较运算符优先级高, 因而没有括号分隔比较表达式时, `&` 连接运算符将一个字符串和一个对象进行连接, 此时即会造成该错误发生。

9.2.3 获取对象类型

在用户编写 VBE 代码的初期, 了解各种程序代码的对象是十分重要的。本小节中通过 `TypeName` 函数获取活动对象的类型。

本示例主要通过 `TypeName` 函数来获取相应活动对象的类型。读者在使用这些对象时, 可以将其作为相应对象进行处理。

```
Sub getinfo()
```

```
MsgBox "活动单元格类型: " & TypeName(ActiveCell) & vbCrLf & _
      "活动工作表类型: " & TypeName(ActiveSheet) & vbCrLf & _
      "活动工作簿类型: " & TypeName(ActiveWorkbook) & vbCrLf & _
      "活动窗口类型: " & TypeName(ActiveWindow) & vbCrLf & _
      "活动图表类型: " & TypeName(ActiveChart) & vbCrLf & _
      "活动打印机: " & TypeName(ActivePrinter)
End Sub
```

运行程序代码，得到如图 9-7 所示的信息框。

由于工作簿中没有图表对象被激活，此时图表对应类型为 Nothing。另外需要注意的是活动打印机返回的类型是字符串类型的。此外读者还需要了解一点，即活动工作表与活动单元格之间是互不影响的。这句话的意思是，如果当前选中了某个图表对象，此时活动单元格对象不一定是 Nothing，但如果没有选中图表，则活动图表一定返回 Nothing。



图 9-7 活动对象的类型

9.2.4 获取或设置程序状态

Application 对象中包含了很多与状态信息相关的属性，如 Excel 2007 应用程序左上角到显示器窗口的顶部、左边界的距离信息。应用程序的宽高、标题、句柄、所在路径、可见性等，这些信息都是有关 Excel 2007 应用程序十分有价值的基本信息。通常通过这些属性，可以直接控制 Excel 2007 应用程序在屏幕中的显示形式。另外 Application 对象还包含了部分可以控制 Excel 2007 组成部件是否显示的属性。通过这些属性的设置，可以显示或隐藏类似名称框、编辑框、状态条等部件。

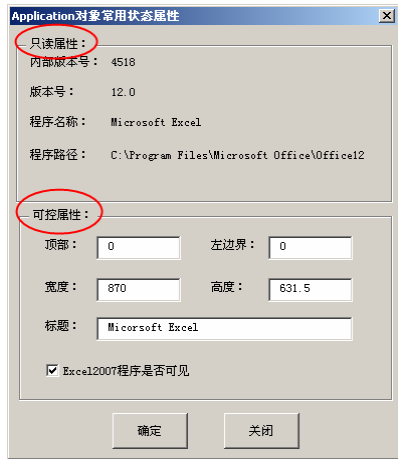


图 9-8 Application 对象常用状态属性控制窗口

本示例中对 Excel 2007 常用状态属性集中进行设置，其中包括 Build、Version、Value、Path、Left、Top、Width、Height、Caption 以及 Visible 属性。为了综合显示这些信息，本小节需要借用窗体对象，但是并不涉及到窗体的具体知识。当窗体初始化时，会获取 Excel 2007 的初始状态数据。本示例演示如何控制 Excel 2007 应用程序常用状态属性。示例中的窗体效果如图 9-8 所示。

要显示上面的效果，需要使用到窗体对象。但是，这里并不详细介绍窗体创建的过程，只对设置的控件加以说明。窗体中用于显示只读属性的标签共 4 个，于设置可读写的属性的控件共 6 个，其中包含一个复选框控件，另外还包含了两个按钮控件。这些控件详细设置信息如表 9-1 所示。

表 9-1 窗体控件设置表

控件名	控件类型	说 明
lblBuild	标签	该标签用于显示 Excel 2007 程序的内部版本号信息

续表

控件名	控件类型	说 明
lblVersion	标签	该标签用于显示 Excel 2007 程序的版本信息
lblValue	标签	该标签用于显示 Excel 2007 的程序名称
lblPath	标签	该标签用于显示 Excel 2007 程序所在文件夹的路径
txtTop	文本框	该文本框用于显示或设置当前 Excel 2007 程序顶部和屏幕顶部之间的距离
txtLeft	文本框	该文本框用于显示或设置当前 Excel 2007 程序左侧与屏幕左侧之间的距离
txtWidth	文本框	该文本框用于显示或设置当前 Excel 2007 程序的宽度
txtHeight	文本框	该文本框用于显示或设置当前 Excel 2007 程序的高度
txtCaption	文本框	该文本框用于显示或设置当前 Excel 2007 程序的标题
chkVisible	复选框	该复选框用于显示或设置当前 Excel 2007 程序的可见性
btnOK	按钮	该按钮被单击时，程序将把当前设置的属性应用到 Excel 2007 程序
btnClose	按钮	该按钮被单击时，将关闭并卸载本窗口

提示：用户需要认真注意控件名称的命名方式。命名除了依据统一的标识符命名方案外，每个名称的前缀后都跟随的是该控件指向 Excel 2007 的属性名称。这种命名方式为后面编写程序代码提供了方便。

建立好窗体后，即可为窗体创建相应的代码。窗体中一共包含了 3 个事件过程以及一个自定义函数过程，另外还包含了一个自定义枚举数据类型的定义。整个代码并不复杂，但是由于处理的属性比较多，并且在不同情况下还需要多次分情况设置属性，因而造成自定义函数过程的代码比较长。建立窗体代码的过程如下。

(1) 在工程资源管理器中找到窗体 frmAppInfo，然后右击该窗体，在弹出的菜单中选择“查看代码”命令，在代码窗口的对象选择下拉列表框中选择“UserForm”，然后在其右侧的事件过程下拉列表框中选择“Initialize”，然后在该事件过程中输入如下代码。

```

Enum RAndWType                                '自定义枚举数据类型定义
    typeRead = 0                               '读属性操作
    typeWrite = 1                             '写属性操作
End Enum

Private Sub UserForm_Initialize()
Dim objCtl As Control, strTemp$
For Each objCtl In Me.Controls                '遍历窗体中所有控件对象
    strTemp = objCtl.Name                     '获取控件的代码名称
    Select Case UCase(Left(strTemp, 3))       '检测控件代码名称前缀
        Case "LBL"                           '检测控件是否为标签控件
            strTemp = Right(strTemp, Len(strTemp) - 3) '获取控件名称中包含的 Application 属性名
            objCtl.Caption = PropertyVal(strTemp) '获取对应属性值然后显示到标签
        Case "TXT"                           '检测控件是否为文本框
            strTemp = Right(strTemp, Len(strTemp) - 3) '获取控件名称中包含的 Application 属性名
            objCtl.Text = PropertyVal(strTemp) '获取对应属性值然后显示到文本框
        Case "CHK"                           '检测控件是否为复选框
            strTemp = Right(strTemp, Len(strTemp) - 3) '获取控件名称中包含的 Application 属性名
            objCtl.Value = PropertyVal(strTemp) '获取对应属性值然后设置复选框勾选状态
    End Select

```

```
Next
Set objCtl = Nothing
End Sub
```

(2) 在代码窗口的对象选择下拉列表框中选择“btnOK”，然后在其右侧的事件过程下拉列表框中选择“Click”，随后在该单击事件过程中输入如下代码。

```
Private Sub btnOK_Click()
Dim objCtrl As Control
Dim objCtl As Control, strTemp$
For Each objCtl In Me.Controls           '遍历窗体中所有控件
    strTemp = objCtl.Name                 '获取控件名称
    Select Case UCase(Left(strTemp, 3))   '检测控件的类型
        Case "TXT"                       '检测控件是否为文本框
            strTemp = Right(strTemp, Len(strTemp) - 3) '获取控件名称中 Application 属性名
            PropertyVal strTemp, objCtl.Text, typewrite '将相应属性值设置给 Application 属性
        Case "CHK"                       '检测控件是否为复选框
            strTemp = Right(strTemp, Len(strTemp) - 3) '获取控件名称中 Application 属性名
            PropertyVal strTemp, objCtl.Value, typewrite '将相应属性值设置给 Application 属性
    End Select
Next
Set objCtl = Nothing
End Sub
```

(3) 在代码窗口的对象选择下拉列表框中选择“btnClose”，然后在其右侧的事件过程下拉列表框中选择“Click”，随后在该单击事件过程中输入如下代码。

```
Private Sub btnClose_Click()
Unload Me
End Sub
```

(4) 以下为自定义 PropertyVal 的定义代码，读者直接将以下代码输入到代码模块中即可。

```
Private Function PropertyVal$(strProperty$, Optional strVal$, Optional intType As RAndWType = 0)
Dim strTemp$
Select Case LCase(strProperty)           '检测 Application 的属性
    Case "build"                           '检测属性是否为只读属性 build
        PropertyVal = Application.Build    '函数返回 Application 的 build 属性值
    Case "version"                         '检测属性是否为只读属性 version
        PropertyVal = Application.Version  '函数返回 Application 的 version 属性值
    Case "value"                           '检测属性是否为只读属性 value
        PropertyVal = Application.Value    '函数返回 Application 的 value
    Case "path"                             '检测属性是否为只读属性 path
        PropertyVal = Application.Path     '函数返回 Application 的 path 属性值
    Case "top"                             '检测属性是否为可读写属性 top
        If intType Then                    '检测操作类型
            Application.WindowState = xlNormal '设置 Excel2007 窗口模式为普通模式
            Application.Top = strVal         '设置 Application 的 top 属性值
        Else
            If Application.Top < 0 Then      '检测 Application 的属性是否为负
                PropertyVal = 0              '返回 0
            Else
                PropertyVal = Application.Top '函数返回 Application 的 top 属性
            End If
        End If
    End Select
End Function
```



```

        End If
    End If
    Case "left"
        If intType Then
            Application.WindowState = xlNormal
            Application.Left = strVal
        Else
            If Application.Left < 0 Then
                PropertyVal = 0
            Else
                PropertyVal = Application.Left
            End If
        End If
    Case "width"
        If intType Then
            Application.WindowState = xlNormal
            Application.Width = strVal
        Else
            PropertyVal = Application.Width
        End If
    Case "height"
        If intType Then
            Application.WindowState = xlNormal
            Application.Height = strVal
        Else
            PropertyVal = Application.Height
        End If
    Case "caption"
        If intType Then
            Application.Caption = strVal
        Else
            strTemp = Application.Caption
            If InStr(1, strTemp, "-") Then
                strTemp = Left(strTemp, InStr(1, strTemp, "-") - 1)
            End If
            PropertyVal = Trim(strTemp)
        End If
    Case "visible"
        If intType Then
            Application.Visible = strVal
        Else
            PropertyVal = Application.Visible
        End If
    End Select
End Function

```

'检测属性是否为可读写属性 left
'检测操作类型
'设置 Excel2007 窗口模式为普通模式
'设置 Application 的 left 属性值
'检测 Application 的属性是否为负
'返回 0
'函数返回 Application 的 left 属性
'检测属性是否为可读写属性 width
'检测操作类型
'设置 Excel2007 窗口模式为普通模式
'设置 Application 的 width 属性值
'函数返回 Application 的 width 属性
'检测属性是否为可读写属性 height
'检测操作类型
'设置 Excel2007 窗口模式为普通模式
'设置 Application 的 height 属性值
'函数返回 Application 的 Height 属性
'检测属性是否为可读写属性 caption
'检测操作类型
'设置 Application 的 caption 属性值
'获取 Application 的 caption 属性值
'查找 Application 的 caption 中第一个 "-" 的位置
'取 caption 中第一个 "-" 前的字符串
'函数返回 Application 的 caption 属性
'检测属性是否为可读写属性 visible
'检测操作类型
'设置 Application 的 visible 属性值
'函数返回 Application 的 visible 属性

(5) 运行上面的窗体代码, 此时立即会弹出如图 9-9 所示的窗口。在可控框架内部的文本框中设置相应的属性值, 然后单击“确认”按钮即可改变当前 Excel 2007 的显示状态。

上述窗体的初始化与确定按钮单击事件代码中, 都通过检测控件的名称来确定控件的类型, 并且通过名称中保存的 Application 对象属性名来确定给该标签、文本框或复选框返回的设置值类型。如此处理极大地方便了对窗体中控件的操作, 从而可以将循环窗体的控件与获取控件的设置值两个功能分离开来。程序中正是通过 PropertyVal 自定义 Function 过程来设置

控件的显示内容以及为 Excel 2007 程序指定相应属性值。

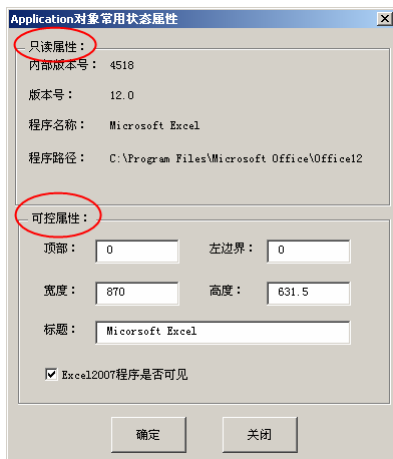


图 9-9 窗体结果

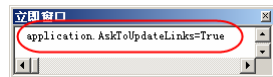
9.2.5 控制 Excel 2007 交互方式

Application 对象包含了很多用于控制交互方式的属性,这些属性在实际开发中相当实用,它们直接决定了 Excel 2007 程序对用户操作的反应方式。例如,ScreenUpdating 属性设置为 True 时,可以开启 Excel 2007 程序的自动屏幕刷新。该功能是实际应用中十分常用的属性,在恰当的时机关闭自动刷新可以大幅度提高程序运行效率。另外还有部分交互属性设置可以针对实际情况弹出对话框,征求用户具体的操作方式。

本示例以 AskToUpdateLinks 属性的设置为例,说明会针对实际情况弹出询问窗口的交互属性设置的方法。操作步骤如下。

(1) 进入 VBE,按下<Ctrl>+<G>快捷键打开立即窗口,在立即窗口中输入如下代码,如图 9-10 所示。

```
application.AskToUpdateLinks=True
```



(2) 返回到 Excel 2007 操作界面,并连续打开两个工作簿文件,然后选择菜单栏中的“视图”→“窗口”→“全部重排”命令,弹出“重排窗口”对话框,点选“平铺”单选钮,然后单击“确定”按钮。

图 9-10 设置 AskToUpdateLinks 属性为真

(3) 在第一个工作簿的 A1 单元格中,输入一个等号。然后单击另一工作簿某工作表的标签,并选择该工作表的 A1 单元格,如图 9-11 所示。

(4) 将第一个工作簿文件保存后,关闭两个工作簿,然后再次重新打开第一个工作簿文件。打开该工作簿后,在 Excel 2007 选项卡下方显示如图 9-12 所示的自动更新链接选项安全警告。

(5) 单击“选项”按钮,弹出对话框,提示是否更新链接,如图 9-13 所示。如果点选“启用此内容”单选钮,当单击“确定”按钮后,Excel 2007 将自动更新工作簿中的链接公式数据。

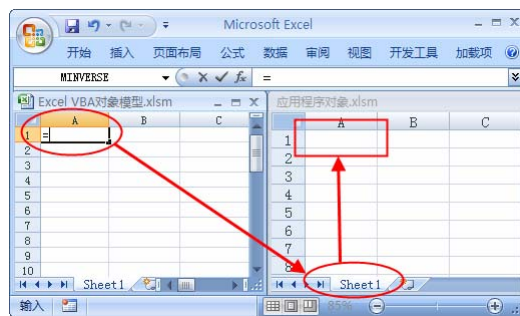


图 9-11 在工作簿中创建链接公式

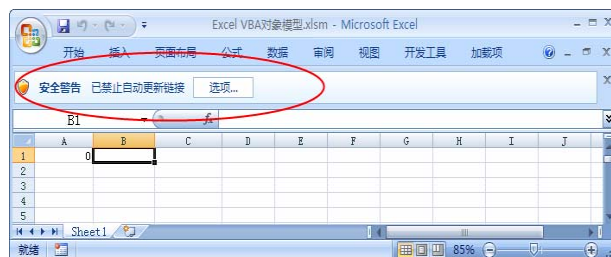


图 9-12 自动更新链接选项安全警告

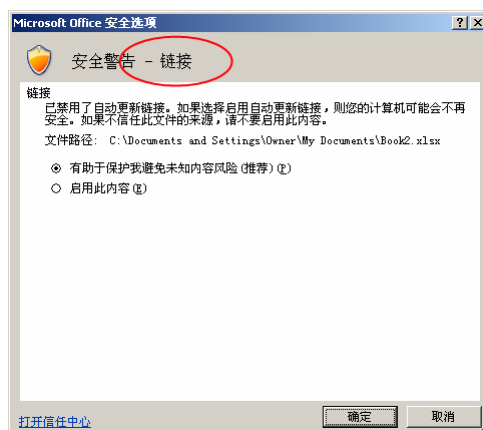


图 9-13 更新链接提示框

9.2.6 了解 Application 的子对象

通过 Application 对象除了可以获取 Workbooks、Worksheets、Range 等基本框架对象与集合外，还可以获取诸如 CommandBars、Dialogs、Names 等对象。这些对象的属性与方法的用法在随后的各个章节依次做重点介绍。这里主要说明 WorksheetFunction 对象的应用，该对象的应用较为简单，因而在本节范围内说明比较适宜。

WorksheetFunction 该对象与 Excel 2007 的工作表函数密切相关。用户可以将工作表的公

式通过 WorksheetFunction 对象直接应用到 Excel 2007 VBA 代码中。

本示例通过使用 WorksheetFunction 对象来说明这些函数在 VBA 中的应用。很多用户发现在 VBA 中没有简便的处理全角与半角字符的函数，但是通过 Excel 2007 中的工作表函数 ASC、DBCS 可以轻松实现全角与半角字符的相互转换。以下两行代码将分别在立即窗口中输出半角字符串、全角字符串。

```
Sub stachange()
Debug.Print WorksheetFunction.Asc("Excel 2007 VBA")
Debug.Print WorksheetFunction.Dbcs("Excel 对象")
End Sub
```

运行程序代码，得到的结果如图 9-14 所示。

从图中可以看出，ASC 函数将字符串中所有的双字节字母以及数字都转换为了单字节字母与数字；而 DBCS 函数则将字符串中所有单字节字母转换为双字节字母。

说明：全角字符在双字节字符集（DBCS）语言中表示双字节字符，半角字符则表示单字节字符。

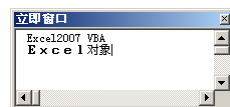


图 9-14 运行结果

9.2.7 使用“打开”与“另存为”对话框

在 Excel 2007 VBA 应用开发过程中，经常需要使用“打开”或“另存为”工作簿操作。用直接的语言来描述就是，获取某工作簿文件的路径或工作簿另存时的路径字符串。用户在使用 Excel 2007 “打开”对话框时，会弹出一个类似如图 9-15 所示的对话框。本章将要为读者介绍如何在 VBA 代码中打开并使用该对话框。

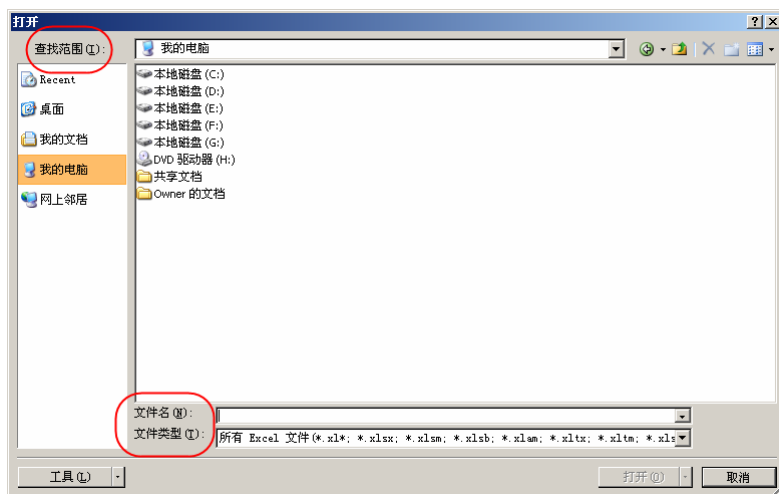


图 9-15 标准的“打开”对话框

Application 对象有两个方法分别用于显示标准的“打开”与“另存为”对话框。这两个方法的名称分别是 GetOpenFilename 和 GetSaveAsFilename。两个方法的参数格式与设置基本相似，但也存在部分差异。这两个方法的调用格式如下。

从上面的调用格式可以看出两个方法的参数有很多是相同的。在实际应用中，其意义也是一致的。另外需要注意的是，两个方法的参数都是可以省略的，其意义是两个方法都可以使用无任何参数的调用格式，但是在实际应用不建议采用这种方法。这两个方法各个参数的具体意义如表 9-2 所示。

参数名称	说明
InitialFilename	指定建议的另存文件名，它只在“另存为”方法中有效。如果省略该参数，Microsoft Excel 将使用活动工作簿的名称
FileFilter	指定一个文件筛选条件的字符串。该字符串中包含了单个或多个文件后缀名的组合。“打开”或“另存为”对话框根据 FilterIndex 参数找到指定后缀，然后筛选出当前文件夹下所有该类文件
FilterIndex	指定默认文件筛选条件的索引号，取值范围为 1 到由 FileFilter 所指定的筛选条件数目。如果省略该参数，或者该参数的值大于可用筛选条件数，则使用第一个文件筛选条件
Title	指定对话框的标题。如果省略该参数，则“打开”对话框标题为“打开”，“另存为”对话框标题为“另存为”
ButtonText	该参数在 Windows 操作系统中无效
MultiSelect	该参数确认用户是否可以在打开对话框中选多个文件。如果为 True，则允许选择多个文件名；如果为 False，则只允许选择一个文件名

(1) 新建工作簿，进入 VBE，添加新的模块，然后在该模块的代码窗口中输入如下代码。

```
Sub openfiles()  
Dim strOtherWB As Variant, strFilter$, strCaption$  
  
strFilter = "Excel97-2003 工作簿(*.xls),*.xls," & _  
            "Excel2007 工作簿(*.xlsx),*.xlsx," & _  
            "Excel 工作簿(*.xls;*.xlsx),*.xls;*.xlsx," & _  
            "所有的文件(*.*),*.*"  
strCaption = "打开多个工作簿"  
strOtherWB = Application.GetOpenFilename(strFilter, 3, strCaption, , True)  
If strOtherWB = False Then Exit Sub  
For i = 1 To UBound(strOtherWB)  
    Workbooks.Open (strOtherWB(i))  
Next  
End Sub
```

(2) 运行程序代码，得到如图 9-16 所示的对话框。

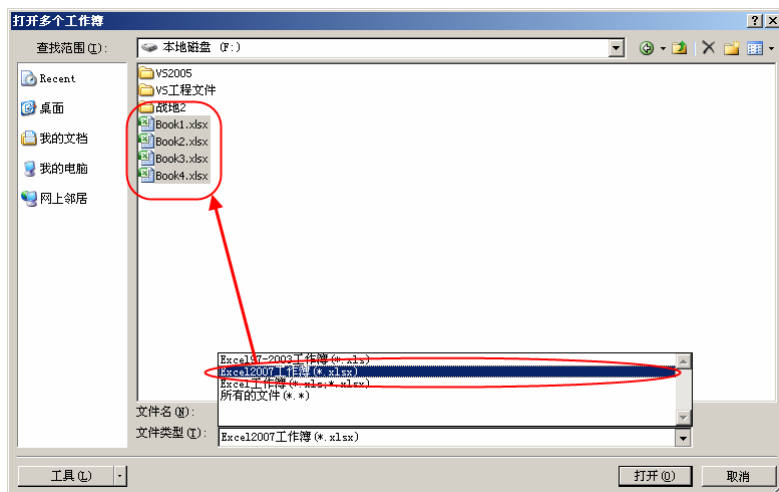


图 9-16 打开多个工作簿对话框

在“文件类型”下拉列表框中选择“Excel 2007 工作簿(*.xlsx),*.xlsx”选项（默认的项目是显示所有工作簿文件），此时将会在文件浏览区看到所有 Excel 2007 工作簿文件。使用 Application 对象的 GetOpenFilename 方法，关键需要理解以下 3 个方面的内容。

- ❑ 文件类型筛选字符串的设置。
- ❑ 如何使用方法返回多工作簿路径字符串数组。
- ❑ 当用户未选择打开按钮并直接关闭了窗口时如何处理方法返回值。

下面将依次说明这 3 个问题。

- ❑ 文件类型筛选器设置：文件类型筛选设置字符串的内容可以分为两类，即显示到文件类型下拉列表框中的提示文字信息、实际用于筛选类型的文件后缀。在设置这些内容时，通过逗号(,)来分隔各个组成部分。并且其出现顺序应遵循相应规律，提示文字信息在前，筛选文件后缀名紧随其后。当一条筛选文件类型中包含了多个文件类型时，文件类型之间需要使用分号(;)来分隔。
- ❑ 使用方法返回值：当用户在对话框中选择了多个工作簿文件后并选择打开按钮时，方法将返回一个数组。打开多少个工作簿，数组就包含了多少个元素。每个数组元素都是工作簿文件的绝对路径字符串，将该字符串传递给 Workbooks 集合对象的 Open 方法可以打开该工作簿文件。
- ❑ 窗口直接关闭的处理：用户在“打开”对话框中单击“关闭”按钮或单击右上角的叉号都会直接退出窗口，此时方法将返回一个逻辑值 False。通过一个 If 语句对函数返回值进行判断，如果等于 False 则立即退出过程即可。

9.2.8 MacroOptions 设置宏选项

本节将介绍如何通过 VBA 代码实现设置宏的功能，另外通过该节所介绍的命令还可以为自定义函数建立自定义函数分类，用户也可以将自定义函数设置到原有的系统函数分类里。本节介绍 MacroOptions 方法，其调用格式如下。

Application.MacroOptions(Macro, Description, HasMenu, MenuText, HasShortcutKey, ShortcutKey, Category, StatusBar, HelpContextID, HelpFile)

Application 对象的 MacroOptions 方法参数众多，该方法详细的参数说明如表 9-3 所示。

表 9-3 MacroOptions 方法参数表

参数名称	说明
Macro	宏的名称或用户自定义函数的名称
Description	用户指定的宏描述字符串
HasMenu	忽略该参数
MenuText	忽略该参数
HasShortcutKey	如果为 True，则为宏指定一个快捷键（还必须指定 ShortcutKey）；如果该参数为 False，则不为宏指定快捷键。如果宏已经有快捷键，则将该参数设置为 False 可删除快捷键。默认值为 False
ShortcutKey	如果 HasShortcutKey 为 True，则该参数为必选参数；否则忽略该参数
Category	一个指定现有的宏函数类别的整数（如财务、日期与时间或用户定义），还可指定自定义类别的字符串。如果提供了一个字符串，它将作为类别名称显示在“插入函数”对话框中。如果此类别名称从未使用过，则将该名称定义一个新的类别。如果使用的类别名称与某个内置名称相同，则 Excel 会将用户定义的函数映射为此内置类别
StatusBar	宏的状态栏文本
HelpContextID	一个指定分配给宏的帮助主题上下文 ID 的整数
HelpFile	包含 HelpContextID 定义的帮助主题的帮助文件名

本示例为建立的自定义函数“STRSPLIT”建立函数描述信息，并将该自定义函数分配到由系统定义的文字函数分组中。操作步骤如下。

（1）新建工作簿，进入 VBE，添加新的模块，然后在该模块的代码窗口中输入如下代码。

```
Sub SetMaroOption()  
Dim strTemp$  
strTemp = "按照某分隔字符串分割字符串。" & vbCrLf & _  
    "通过该函数可以获取分隔后的某子字符串或位置信息。" '设置描述字符串  
Application.MacroOptions macro:="STRSPLIT", _  
    Description:=strTemp, Category:="文本" '设置 STRSPLIT 函数  
strTemp = "本数组公式可以将某一列数据按降（升）序排列。" '设置描述字符串  
Application.MacroOptions macro:="SORTED", _  
    Description:=strTemp, Category:="数据分析" '设置 SORTED 函数  
End Sub
```

（2）运行上面的程序代码，然后返回 Excel 2007 操作界面。

（3）新建一个工作表，然后选中该工作表的 A1 单元格。选择插入函数，打开“插入函数”对话框，在“选择类别”下拉列表框中选择“文本”选项，然后在“选择函数”下拉列表框中选择“STRSPLIT”函数，此时在对话框底部会显示如图 9-17 所示的提示信息。

（4）在“插入函数”对话框的“选择类别”下拉列表框中选择“数据分析”选项，此时在选择函数列表框中将会显示该项下所有的函数名称，此处只建立一个该类别的函数即 SORTED 函数，如图 9-18 所示。

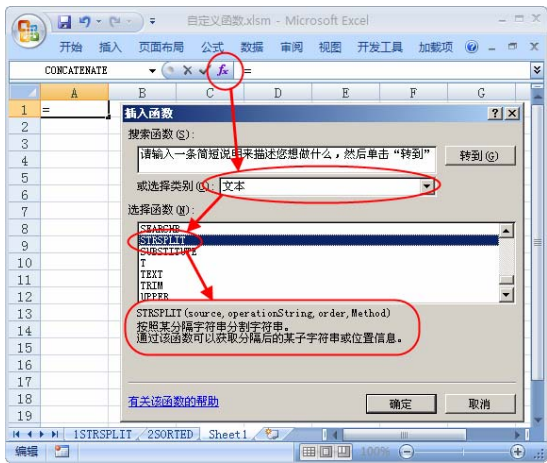


图 9-17 “插入函数”对话框

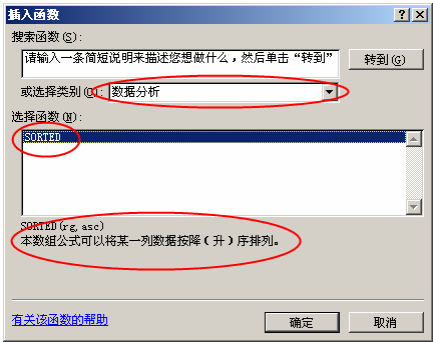


图 9-18 数据分析项的函数以及描述信息

在为 MacroOptions 方法指定 Category 参数时,用户可以使用两种类型的数据。一种是 1~32 间的整型数据;另一种是字符串。当字符串与系统定义的分类字符串相等时,则函数将被分配到该系统分类中,否则将建立该种函数分类并将该函数分配到该分类中。另外,1~13 间的整数分别对应一个系统分类,其他整数数值都对应着自定义函数分类。Category 参数值与系统分类如表 9-4 所示。

表 9-4 Category 参数值与系统分类

整型值	系统分类	整型值	系统分类
1	财务	8	逻辑
2	日期与时间	9	信息
3	数学与三角函数	10	命令
4	统计	11	自定义
5	查找与引用	12	宏控件
6	数据库	13	DDE/外部
7	文本		

9.2.9 OnKey 与 SendKeys 方法

读者已经了解了通过 VBA 代码,实现部分在宏管理窗口中选项按钮功能的实现方法。但是还有一个问题没有解决,即如何为宏过程指定快捷键,本节将为读者介绍通过 Application 对象的 OnKey 方法为宏过程指定快捷键。另外还介绍与按键相关的方法,即 SendKeys,该方法将模拟击键效果。这两个方法的调用格式如下。

```
Application.OnKey(Keys, Procedure)
Application.SendKeys(Keys, Wait)
```

两个方法的参数都很少, OnKey 方法可以检测击键,当该键为指定键或组合键时将执行

指定的宏过程。**SendKeys** 则将指定的键发送给 Excel 2007，通知程序键盘按下了某个键或组合键。该方法事实上是在“欺骗”Excel 2007，因为此时用户可能根本就没有按击该键或组合键。这两个方法的参数表如表 9-5 所示。

表 9-5 OnKey、SendKeys 方法参数表

名称	说明
Keys	按键或组合键的字符串。OnKey 方法检测用户按键是否为该键或组合键，SendKeys 方法将把该按键或组合键的按击消息发送给 Excel 2007，让其对按键做出相应动作
Procedure	表示要运行的过程名称的字符串。如果 Procedure 为空文本 (" ")，则按 Key 时不发生任何操作。该格式的 OnKey 将更改键击在 Microsoft Excel 中产生的正常结果。如果省略 Procedure 参数，则 Key 恢复为 Microsoft Excel 中的正常结果，同时清除先前使用 OnKey 方法所做的特殊键击设置
Wait	如果为 True，则 Microsoft Excel 会等到处理完按键后将控制权返回给宏；如果为 False（或者省略该参数），则继续运行宏而无需等待处理完按键

在两个方法中，指定 **Keys** 参数的方法都是一致的（这里将把设置该参数的注意事项归纳后一起加以介绍），该参数可指定任何单个键或与<Ctrl>、<Shift>或<Alt>的组合键（或者这些键的组合）。每个键可用一个或多个字符表示。例如，"a"表示字符 a，或者"{ENTER}"表示 Enter。另外，屏幕无回显按键对应代码表以及功能键的代码如表 9-6 和表 9-7 所示。

表 9-6 屏幕无回显按键对应代码表

按键	对应设置代码	按键	对应设置代码
<BACKSPACE>	{BACKSPACE}或{BS}	<Ins>	{INSERT}
<Break>	{BREAK}	<←>	{LEFT}
<Caps Lock>	{CAPSLOCK}	<NumLock>	{NUMLOCK}
<Clear>	{CLEAR}	<PageDown>	{PGDN}
<Delete>或	{DELETE}或{DEL}	<PageUp>	{PGUP}
<↓>	{DOWN}	<Return>	{RETURN}
<End>	{END}	<→>	{RIGHT}
<Enter>（数字小键盘）	{ENTER}	<Scroll Lock>	{SCROLLLOCK}
<Enter>	~（波形符）	<Tab>	{TAB}
<Esc>	{ESCAPE}或{ESC}	<↑>	{UP}
<Help>	{HELP}	<F1>到<F15>	{F1}到{F15}
<Home>	{HOME}		

表 9-7 功能键的代码表

按键	对应设置代码
<Shift>	+（加号）
<Ctrl>	^（插入符号）
<Alt>	%（百分号）

本示例将通过 OnKey 代码为一个自定义宏过程指定快捷键，然后通过 SendKeys 方法向 Excel 2007 放松该快捷键激发宏过程。宏过程被调用后，会弹出一个信息窗口，告知用户宏过程被执行成功。操作步骤如下。

(1) 新建工作簿，进入 VBE，添加新的模块，然后在该模块的代码窗口中输入如下代码。

```
Sub OnKeySendKeys()
Application.OnKey "+^{q}", "ShowInfo"
Application.SendKeys "+^{q}", True
End Sub

Sub ShowInfo()
MsgBox "宏过程调用成功！"
End Sub
```

(2) 返回到 Excel 工作表中，然后打开宏管理窗口，如图 9-19 所示。

(3) 选择 OnKeySendKeys 宏过程，然后单击“执行”按钮，系统弹出如图 9-20 所示的信息框。

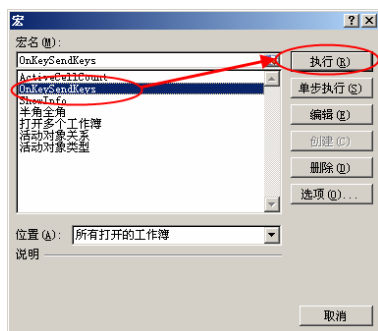


图 9-19 执行 OnKeySendKeys 宏过程



图 9-20 信息框

提示：不要在 VBE 中直接运行 OnKeySendKeys 宏过程，此时将不会产生任何效果，因为 OnKey 方法设置的侦测只对 Excel 2007 操作界面有效。当在 VBE 中使用 Application 对象的 OnKeySendKeys 方法时，激发的是 VBE 中对应的快捷键。而 VBE 中没有对应 <Ctrl>+<Shift>+<Q>快捷键的操作，因而 OnKeySendKeys 方法被执行后不会有任何反应。

9.3 FileDialog 与 Dialogs 对象

在笔者刚开始接触 Excel VBA 的时候曾经为如何获取文件路径或者保存文件路径苦恼了很久，后来了解到可以通过 GetOpenFilename 和 GetSaveAsFilename 达到该目的。正是通过认识这两个方法，笔者了解到 Excel 内置对话框窗口的强大，从而又接触到了 Dialog 对话框对象这位“好朋友”。通过该对话框对象，可以帮助开发者轻松实现众多以往难以想象的功能。

Excel 2007 中的 Dialogs 对话框集合对象包含了所有 Excel 2007 中内置的对话框。用户在 Excel VBA 中可以通过实例化一个相应类型的对话框对象来直接使用该种对话框。这也就意味着在用户的 Excel VBA 应用中，可以借助 Excel 2007 内置对话框来完成众多的界面交互操作。这些对话框的调用大部分都比较简单，这不仅让开发者“省心”很多，也大大提高了开

发效率。Dialog 对话框对象功能强大，但是读者需要清楚了解其局限。Dialog 对话框对象仅限于 Excel 2007 已提供的对话框。如果开发者的应用程序有更多的交互需求，那么只有选择开发自定义窗体了。

9.3.1 FileDialog 文件对话框对象

FileDialog 文件对话框对象专门用于处理与文件相关的打开和保存操作，其功能与本章前面介绍的 GetOpenFilename 和 GetSaveAsFilename 两方法类似。另外读者也可以使用 Dialogs 集合对象完成相应操作。但是使用该对象具有比较明显的“过人之处”，这也是笔者将其单列一节加以介绍的原因之一。

使用该对象书写 VBA 代码可以让代码书写更加清晰、更具结构性。通过该对象还可以控制文件对话框初始显示时的浏览方式。另外该对象在应用中一个最大的优点就是可以获取文件夹路径，该功能是 GetOpenFilename 和 GetSaveAsFilename 两方法无法相比的。对于在 Excel 2007 中进行开发的读者而言，其选择是多种多样的。

FileDialog 对象的属性并不多，这里将其所有有用属性逐一介绍包括 AllowMultiSelect、ButtonName、DialogType、FilterIndex、Filters、InitialFileName、InitialView、SelectedItems 和 Title。其中只有 DialogType、Filters、InitialView 和 SelectedItems 属性是该对象的特有属性，其他属性与 GetOpenFilename 和 GetSaveAsFilename 方法的参数具有相同的意义。FileDialog 对象的特有属性如下。

- ❑ DialogType 属性：该属性返回 MsoFileDialogType 枚举的常量值，代表 FileDialog 对象文件对话框的类型。一旦对话框被创建，用户将不能修改该属性来改变对话框的类型。
- ❑ Filters 属性：该属性将返回一个 FileDialogFilters 集合。通过该集合对象的 Add 方法，用户可以向文件对话框添加新文件筛选器。同时用户也可以遍历该对话框中所有的已设置文件筛选器。
- ❑ InitialView 属性：获取或设置一个 MsoFileDialogView 枚举常量值，代表文件对话框中文件和文件夹的初始显示形式。
- ❑ SelectedItems 属性：获取一个 FileDialogSelectedItems 集合。此集合包含用户在使用 FileDialog 对象的 Show 方法显示的文件对话框中所选的文件的路径列表。

以上 4 个属性中，DialogType 和 Filters 属性是只读属性，其他两个属性都是可读写属性。因而用户不要试图修改已创建文件对话框的 DialogType 属性来改变对话框的类型，否则将会引发错误。但是对于 Filters 集合对象而言，用户可以使用其 Add 方法来改变其筛选器。DialogType 属性和 InitialView 属性允许的常量值如表 9-8 和表 9-9 所示。

表 9-8 MsoFileDialogType 枚举值表

常量名	常量值	描述
msoFileDialogFilePicker	3	“文件选取器”对话框
msoFileDialogFolderPicker	4	“文件夹选取器”对话框

续表

常量名	常量值	描述
msoFileDialogOpen	1	“打开”对话框
msoFileDialogSaveAs	2	“另存为”对话框

表 9-9 MsoFileDialogView 枚举值表

常量名	常量值	描述
msoFileDialogViewDetails	2	文件以带有详细信息的列表形式显示
msoFileDialogViewLargeIcons	6	文件显示为大图标
msoFileDialogViewList	1	文件以不带详细信息的列表形式显示
msoFileDialogViewPreview	4	文件以列表形式显示，并在预览窗格中显示选定的文件
msoFileDialogViewProperties	3	文件以列表形式显示，并在一个窗格中显示选定文件的属性
msoFileDialogViewSmallIcons	7	文件显示为小图标
msoFileDialogViewThumbnail	5	文件显示为缩略图
msoFileDialogViewTiles	9	文件显示为平铺图标
msoFileDialogViewWebView	8	文件显示在 Web 视图中

FileDialog 对象的方法仅有两个，分别是 Show 和 Execute。Show 方法将按照当前对话框设置显示该对话框。通过该方法显示的对话框是模态对话框，当用户退出该对话框后，程序才会继续执行后续任务。而 Execute 方法显示的对话框将不等待用户退出对话框，一旦对话框显示完毕就继续执行程序中后续任务，用户退出对话框时具有返回值。当用户单击“确定”按钮时，其返回值为-1；当用户单击“取消”按钮或者直接单击对话框右上角的关闭按钮，其返回值为 0。

本示例用于演示 FileDialog 对象的使用方法。用户可以在本示例中自行设置对话框的类型以及预览模式。设置完成后，双击工作表的某个单元格时将按照这些设置打开相应类型的对话框。如果用户在对话框中选择了工作簿或路径，这些信息将会输出到工作表中。操作步骤如下。

（1）新建工作簿，将工作簿中某空白工作表的标签修改为“FileDialog”，然后在工作表的 A~B 列中建立两个表格数据。其中 A 列的数据是起到提示效果，B 列的数据将作为程序中的设置代码。在 D1:E1、D3:E3 和 D7 单元格区域中分别设置相应内容的标题，如图 9-21 所示。

（2）选择 D2 单元格，并选择菜单栏中的“数据”→“数据有效性”命令，弹出“数据有效性”对话框。在“允许”下拉列表框中选择“序列”选项，在“来源”下拉列表框中选择 A2:A5 单元格区域。然后单击“确定”按钮。设置完成后，当选中 D2 单元格时，即可从下拉列表框中选择相应类型，其操作的详细步骤如图 9-21 所示。

（3）选中 D5 单元格，同步骤（2）一样进行数据有效性设置，只是此处选择的数据区域应该换为 A8:A16 而已。

（4）在 E2 和 E5 单元格中分别输入如下公式。该公式根据用户选定的类型从 A~B 列

所示的表格中查找相应的常量值。

```
=INDEX(B1:B5,MATCH(D2,A1:A5,0))
=INDEX(B7:B16,MATCH(D5,A7:A16,0))
```

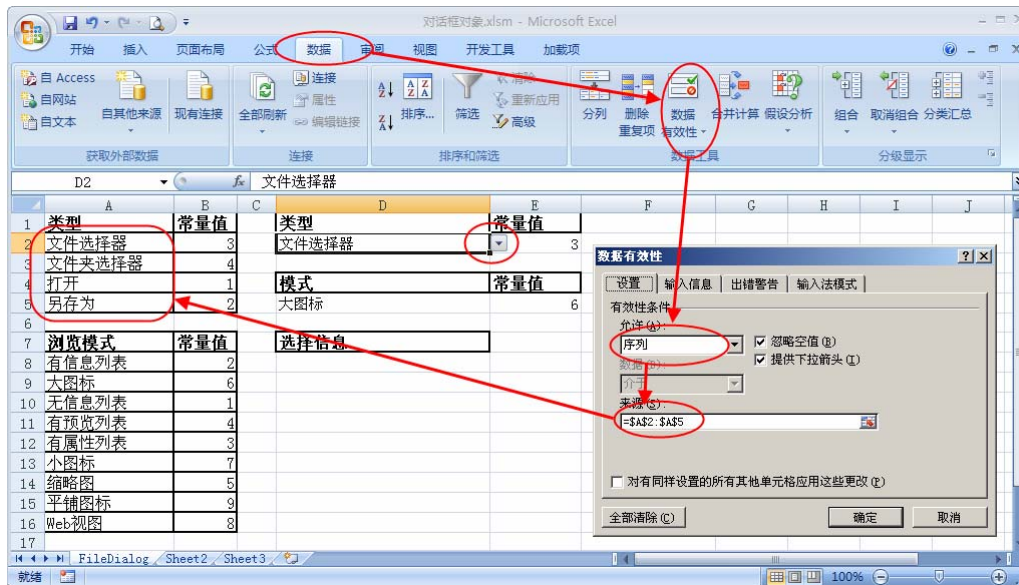


图 9-21 设置数据有效性

(5) 进入 VBE，在工程资源管理器中找到“FileDialog”工作表对象。随后双击该对象，打开其代码窗口。在代码窗口的对象选择器中选择“Worksheet”，在事件下拉列表框中选择“BeforeDoubleClick”，最后在该事件过程输入如下代码。

```
Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)
Dim fleDlg As FileDialog, intRow%
Dim intReturn%
Set fleDlg = Application.FileDialog([E2])           '按指定类型获取一文件对话框对象
With fleDlg
    .InitialView = [E5]                             '设置对话框对象的预览模式
    .AllowMultiSelect = True
    With .Filters
        .Clear                                       '清除对话框的文件筛选器
        .Add "所有文件", "*"                        '添加第一个筛选器
        .Add "Excel2007 工作簿文件", "*.xlsx;*.xlsm" '添加第二个筛选器
        .Add "Excel97-07 工作簿文件", "*.xls;*.xls" '添加第三个筛选器
    End With
    .FilterIndex = 2                                '设置默认的筛选器为第二个筛选器
    intReturn = .Show                                '显示对话框并获取用户操作
    intRow = Range("D" & Rows.Count).End(xlUp).Row '获取 D 列末行有数据单元格的行号
    If intRow > 7 Then Range("D8:D" & intRow).Clear '清除原来的选择信息
    If intReturn = -1 Then                            '检测用户是否选择确定按钮
        For i = 1 To .SelectedItems.Count            '遍历所有选择项
            Range("D" & i + 7) = .SelectedItems(i) '输出选择项字符串到工作表
        Next
    End If
End Sub
```

```
End With
Cancel = True
End Sub
```

'取消系统默认双击事件

(6) 返回 Excel 2007 操作界面。在 D2 单元格中选择“文件选择器”，在 D5 单元格中选择“大图标”，然后在该工作表中双击单元格，此时会弹出如图 9-22 所示的文件对话框。

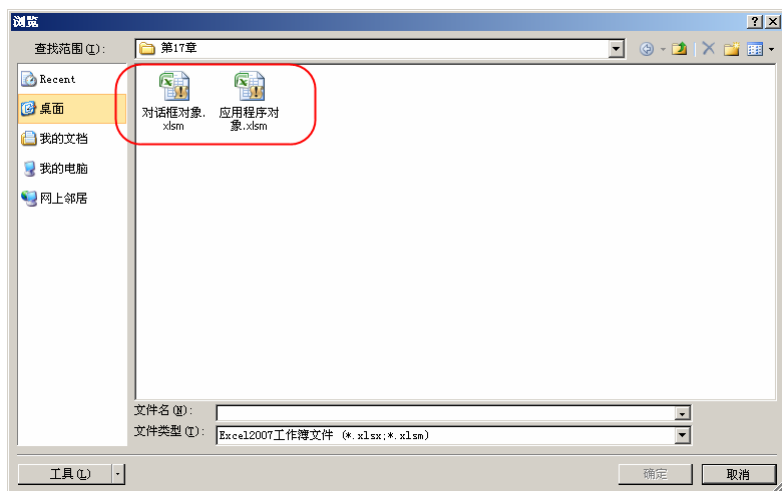


图 9-22 大图标文件选择对话框

当用户单击“确定”按钮后，在工作表的 D7 单元格下方将会获取相应选择项的字符串。

说明：本示例中只对文件对话框的类型和浏览模式进行了自定义，其他的设置都是通过代码直接设置的。读者也可以将本示例进行修改，从而可以达到通过工作表中的输入数值来控制文件对话框目的。例如，读者可以在工作表中定义文件对话框的筛选器。

9.3.2 Dialogs 对话框集合对象

在 Excel 2007 中，用户可以使用多种对话框。Dialogs 集合对象所包含的可使用对象基本上涵盖了所有这些对话框，如打开、另存对话框、单元格格式设置对话框、查找对话框等常用对话框。Excel 2007 部分数据分析功能中也有相应的对话框，如插入公式、条件格式、数据有效性、高级筛选等。

在 9.3.1 节中已经为读者介绍了 FileDialog 对象的使用方法。其实该对话框也可以通过 Dialogs 集合对象访问，并且其他各种类型的内置对话框都是通过该方法访问的。这些对话框对象大多都有其相应的工作环境，读者在使用 Dialogs 集合对象访问对话框时应该注意相应的使用环境。如果当前活动对象为一图表，而用户却试图打开高级筛选对话框，那么将会引发错误。

本示例主要演示在 Excel 2007 中如何通过 Dialogs 集合对象获取相应对话框。在示例的工作表表格中包含了 Excel 2007 中所有可使用对话框的 XlBuiltInDialog 枚举值，其内容包括 3 列信息：常量名、常量值和描述。当用户在工作表记录行中双击单元格，将会弹出相应类型的对话框。操作步骤如下。

(1) 打开前面例子的工作表，然后修改某空白工作表的标签为“XlBuiltInDialog”，在工作表的 A~C 列中分别建立常量名、常量值和描述信息列，如图 9-23 所示。

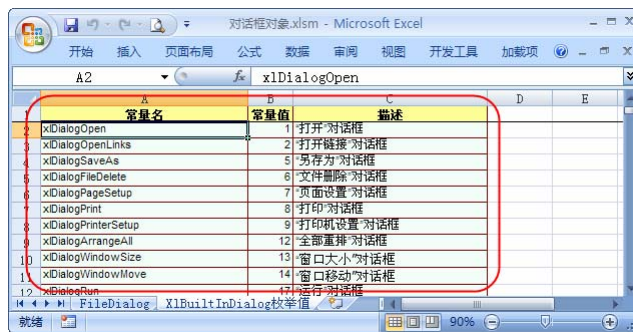


图 9-23 XlBuiltInDialog 枚举值

(2) 进入 VBE，在工程资源管理器中选择“XlBuiltInDialog”工作表，然后双击该对象打开其代码窗口。随后在对象下拉列表框中选择“Worksheet”对象，在事件下拉列表框中选择“BeforeDoubleClick”事件，最后在该事件过程中输入如下代码。

```
Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)
Dim rg As Range
On Error Resume Next
Set rg = Range("B" & Target.Row)           '获取相应的常量值
If Len(rg) Then                             '检测 rg 单元格内容是否空
    Application.Dialogs(rg.Value).Show      '显示相应类型的对话框
End If
On Error GoTo 0
Cancel = True                               '取消系统默认事件过程
End Sub
```

(3) 返回 Excel 2007 操作界面。在 A13 单元格上双击，此时将弹出如图 9-24 所示的“设置打印标题”对话框。

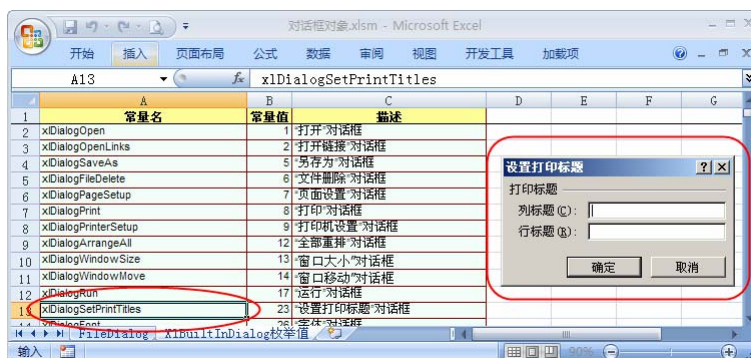


图 9-24 “设置打印标题”对话框

通过以上示例代码，可以看到 Dialogs 集合对象获取对话框时可以设置的内容十分少。Dialog 对象仅包含了 3 个属性，即 Application、Creator 和 Parent。这 3 个属性实际上很少使用，而其方法仅有一个即 Show。对于习惯于采取自定义的读者而言，这无疑是该对象十分

巨大的缺陷。另外需要说明的一点是，当读者在工作表中某些有记录数据行中双击单元格时，可能并没有相应的对话框弹出，这是因为当前 Excel 2007 所处状态并无相应对话框可以开启。

9.4 小结

本章在 9.1 节中讨论了 Excel 2007 VBA 对象模型的基本框架以及各种各样非常实用的对象及集合。其基本框架由 4 部分构成，即 **Application** 应用程序对象、**Workbook** 工作簿对象、**Worksheet** 工作表对象以及 **Range** 单元格区域对象，这些对象中 **Workbook**、**Worksheet** 与 **Range** 对象都有其集合对象。了解该基本框架对象有利于把握 Excel 2007 VBA 对象模型的整体结构、理清认识 Excel 2007 的思路。

除了基本框架对象外，还有众多非常实用的对象与集合。这些实用对象与集合数目繁多，在 Excel 2007 的选项卡中大部分都可以找到对应命令按钮。通过录制宏，读者可以对这些对象与集合的使用方法获取直观的认识。在本篇的随后章节中将逐步为读者介绍这些对象。掌握这些对象与集合的使用，不仅可以丰富读者对于 Excel 2007 VBA 对象模型的认识，而且在实际开发中使用这些对象的功能可以大大提高应用程序的效率。

本章在随后通过众多的示例为读者介绍了 **Application** 对象中各个属性与方法的使用方法。限于篇幅，本章不能将所有属性以及方法为读者加以介绍，讲述的内容大部分都是在实际开发中较为常用的属性与方法，其他一些 **Application** 对象的属性与方法请见本书的附录。另外本章还为读者介绍了在 Excel 2007 VBA 应用开发中可能经常使用到的 **FileDialog** 对象和 **Dialogs** 对话框集合对象。这些对话框可以被用户直接调用，从而极大地简化了交互界面的开发。