

第9章 自定义 Excel 2007 界面

Excel 2003 及以前版本使用菜单栏和工具栏进行操作，在 Excel 2007 中引进行功能区（RibbonX）。本章实例分别介绍定制这两类控制的方法。

9.1 使用工具栏对象

命令栏包括菜单栏和工具栏，在 Excel VBA 中，用 CommandBars 集合对象表示所有的命令栏集合。在 Excel 2003 中创建的自定义菜单栏，在 Excel 2007 中也可打开使用，只是这些菜单都添加到“加载项”选项卡中。

案例 170 创建自定义菜单

1. 功能说明

菜单栏和工具栏是用户与 Excel 进行交互的工具，在 Excel 中，将菜单栏、工具栏和快捷菜单合并为一种功能，称为命令栏，都放置到 CommandBars 集合中。在 Excel 中，可以使用 VBA 代码控制菜单（如增加自定义菜单）。

2. 语法说明

在 Excel VBA 中，CommandBarControl 对象表示具体的命令按钮或菜单项，该对象是 CommandBarControls 集合中的成员。每个 CommandBar 对象包含一个 CommandBarControls 集合对象。由 CommandBar 对象（如主菜单、“常用”工具栏和“格式”工具栏等命令栏）组成 CommandBars 集合对象。CommandBarControl 对象常用属性和方法如下：

- Caption 属性，返回或设置指定命令栏控件的题注文字。
- OnAction 属性，返回或设置一个 VBA 宏（子过程）名，该宏在用户单击或更改某命令栏控件的值时运行。

CommandBarControls 集合对象由 CommandBarControl 对象组成，通过 Add 方法可新建一个 CommandBarControl 对象，并将其添加到指定命令栏上的控件集合中。其语法格式如下：

```
expression.Add(Type, Id, Parameter, Before, Temporary)
```

各参数含义的如下：

- Type 设置添加到指定命令栏的控件类型。可以为下列 MsoControlType 常量之一：msoControlButton、msoControlEdit、msoControlDropdown、msoControlComboBox 或 msoControlPopup。

- **Id** 用来指定内置控件的整数。如果该参数为 1，或者忽略该参数，将在命令栏中添加一个空的指定类型的自定义控件。
- **Parameter** 设置参数信息，对于内置控件，该参数用于容器应用程序运行命令。对于自定义控件，可以使用该参数向 VBA 过程传递信息，或用其存储控件信息。
- **Before** 设置位置信息，表示新控件在命令栏上位置的数字，新控件将插入到该位置控件之前。如果忽略该参数，控件将添加到指定命令栏的末端。
- **Temporary** 为 **True** 将使新命令栏为临时命令栏。临时命令栏在关闭容器应用程序时删除，其默认值为 **False**。

CommandBars 集合对象由 **CommandBar** 对象组成，使用 **CommandBars** 对象的 **Add** 方法可新建一个命令栏并添加到命令栏集合。其语法格式为：

```
expression.Add(Name, Position, MenuBar, Temporary)
```

各参数的含义如下：

- **Name** 为新命令栏的名称。如果忽略该参数，则为默认名称（例如 **Custom 1**）。
- **Position** 设置新命令栏的位置或类型。
- **MenuBar** 为 **True** 将以新命令栏替换活动菜单栏。默认值为 **False**。
- **Temporary** 为 **True** 将使新命令栏为临时命令栏。临时命令栏在关闭容器应用程序时删除。默认值为 **False**。

3. 案例说明

本例的主要功能是，创建自行创建的菜单命令。在默认情况下，Excel 2007 的加载项选项中不包含任何特殊的命令。

4. 编写代码

（1）添加菜单选项的代码如下：

```
Sub CreatMyMenu()
    Dim myMenubar As CommandBar
    Dim myMenu As Object
    Dim myMenuItem As Object
    Set myMenubar = CommandBars.ActiveMenuBar

    Set myMenu = myMenubar.Controls.Add(msoControlPopup, , , True)

    myMenu.Caption = "我的菜单选项"
    Set myMenuItem = myMenu.Controls.Add(Type:=msoControlButton, ID:=1)
    myMenuItem.Caption = "选项 1"
    myMenuItem.Style = msoButtonCaption
    myMenuItem.OnAction = "Memo1"

    Set myMenuItem = myMenu.Controls.Add(Type:=msoControlButton, ID:=2)
    myMenuItem.Caption = "选项 2"
    myMenuItem.Style = msoButtonCaption
    myMenuItem.OnAction = "Memo2"
```

```

Set myMenuItem = myMenu.Controls.Add(Type:=msoControlButton, ID:=3)
myMenuItem.Caption = "选项 3"
myMenuItem.Style = msoButtonCaption
myMenuItem.OnAction = "Memo3"
End Sub

```

(2) 为每个菜单项编写子过程，本例只显示一个提示信息对话框，不编写具体操作的代码。在实际应用中，应在这些子过程中编写完成具体功能的代码。

```

Sub Memo1()
    MsgBox "用户选择了选项 1", vbInformation + vbOKOnly, "我的菜单选项"
End Sub
Sub Memo2()
    MsgBox "用户选择了选项 2", vbInformation + vbOKOnly, "我的菜单选项"
End Sub
Sub Memo3()
    MsgBox "用户选择了选项 3", vbInformation + vbOKOnly, "我的菜单选项"
End Sub

```

5. 运行结果

打开工作簿，默认的菜单选项如图 9.1 所示。当用户运行程序代码后，Excel 自动在加载项中添加“我的菜单选项”命令菜单，并添加对应的菜单命令，如图 9.2 所示。

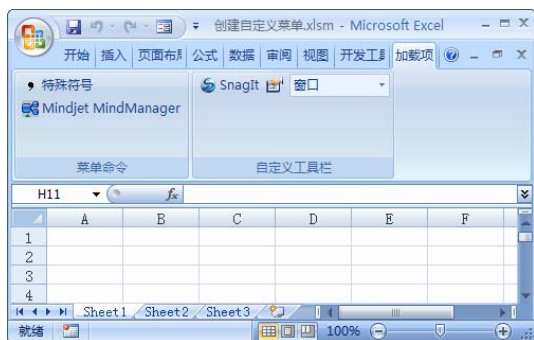


图 9.1 默认的菜单选项

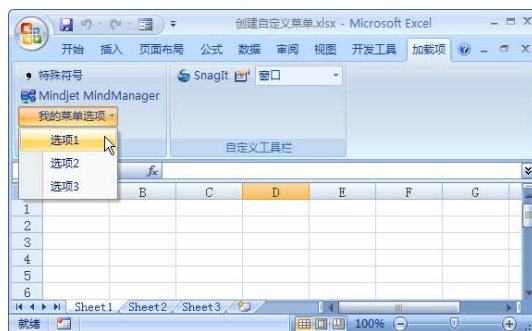


图 9.2 添加的菜单选项

当用户选择相应的菜单选项后，系统触发菜单事件，显示提示信息，如图 9.3 所示。



图 9.3 显示提示信息

6. 程序分析

在上面的代码中，首先获取当前活动菜单栏，接着在活动菜单栏中添加一个下拉菜单，最后为下拉菜单添加菜单项，并设置各菜单项的名称、调用的子过程等参数。

案例 171 删除自定义菜单

1. 功能说明

在 Excel 中，用户可以通过 VBA 代码删除自行添加的自定义菜单。

2. 语法说明

在 Excel VBA 中，用户可以通过 `CommandBars.ActiveMenuBar` 属性获取一个 `CommandBar` 对象，该对象代表容器应用程序中的活动菜单栏。然后在活动菜单栏中确定要删除的菜单选项。

3. 案例说明

本案例的主要功能是删除前面例子中的自定义菜单。

4. 编写代码

删除自定义菜单的具体代码如下：

```
Sub DeleteMyMenu()  
    Dim myMenubar As Object, myMenu As Object  
    Set myMenubar = CommandBars.ActiveMenuBar  
    For Each myMenu In myMenubar.Controls  
        If myMenu.Caption = "我的菜单选项" Then  
            myMenu.Delete  
        End If  
    Next  
End Sub
```

5. 运行结果

本例延续前面小节例子进行。在默认情况下，工作表中包含用户创建的菜单“New Items”，如图 9.4 所示。当用户运行程序后，删除的自定义菜单如图 9.5 所示。

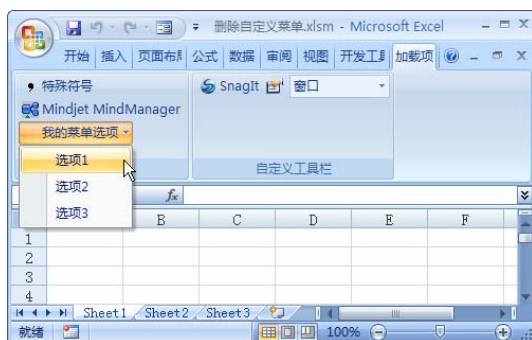


图 9.4 默认添加的菜单选项

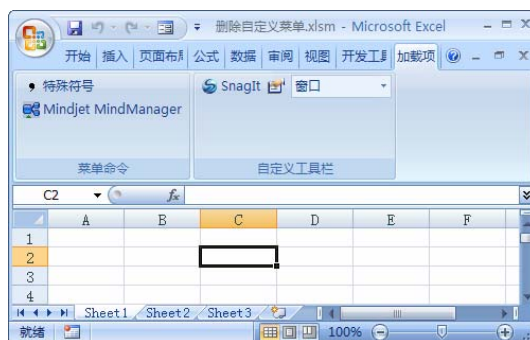


图 9.5 删除菜单选项后的工作表

6. 程序分析

在上面的代码中，首先获取当前活动菜单的引用 `myMenubar`，接着使用 `For Each ... Next` 循环，逐个比较 `myMenubar` 菜单栏中的菜单（包含在 `Controls` 集合中），找到 `Caption` 属性为“New Items”的菜单，则使用 `Delete` 方法将其删除。

案例 172 创建快捷菜单

1. 功能说明

在 Excel 中，用户可以根据需要创建个性化的快捷菜单选项。

2. 语法说明

Excel 的快捷菜单有很多，在不同的对象上单击鼠标右键将弹出不同的快捷菜单项目。若要向这些快捷菜单中添加菜单项，首先需获取对应菜单的引用，再使用以下代码添加菜单项。

```
Application.CommandBars("Cell").Controls.Add
```

3. 案例说明

本案例的主要功能在操作工作表的时候，创建自定义的快捷菜单。

4. 编写代码

（1）本例中添加单元格快捷菜单的代码如下：

```
Sub AddItemcontrol()
Dim cbcs As CommandBarControls, cbc As CommandBarControl
Set cbcs = Application.CommandBars("Cell").Controls
Set cbc = cbcs.Add(Type:=msoControlButton, before:=1, temporary:=True)
With cbc
.Caption = "自定义的菜单选项"
.Tag = "MyCellMenu"
End With
```

End Sub

以上代码首先获取单元格的快捷菜单，接着使用 Add 方法向单元格快捷菜单中添加自定义菜单项。行标签和列标签快捷菜单的代码与此类似。

(2) 关闭工作簿之前，删除上步新增的菜单项，具体代码如下：

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)
Dim cbc As CommandBarControl
For Each cbc In Application.CommandBars("Cell").Controls
If cbc.Tag = "MyCellMenu" Then cbc.Delete
Next
End Sub
```

5. 运行结果

本例的主要功能是为单元格对象添加快捷菜单，添加的结果如图 9.6 所示。

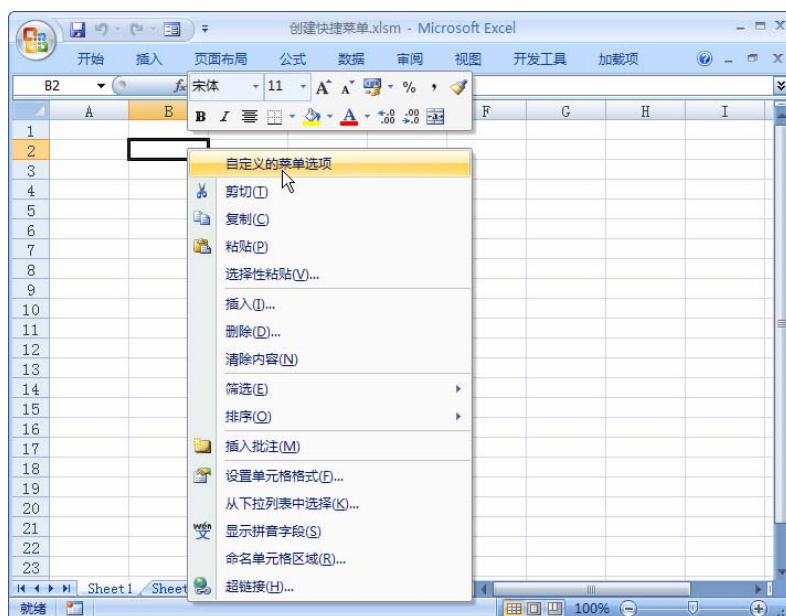


图 9.6 添加的快捷菜单

6. 程序分析

在实际开发中，用户可以为多种对象添加快捷菜单。例如，用户可以为选择工作表的列标签事件添加快捷菜单，也可以为工作表的行标签事件添加快捷菜单。这里就不详细展开介绍了。

9.2 自定义 Excel 2007 的功能区

功能区 (RibbonX) 是 Office 2007 新引进的用户界面, 用户对其也可以进行自行定制。但是, 自定义 RibbonX 不是通过 VBA 代码, 而是 XML 语言实现的。本小节中, 将详细讲解如何使用 XML 定义功能区。

案例 173 自定义 Office 按钮

1. 功能说明

用户可以通过编写 XML 代码来自定义 Excel 2007 中的 Office 按钮中包含的选项。

2. 语法说明

在本例中的 Office 菜单按钮, 多了个自定义菜单项, 该菜单项中包含了三个控件, 分别是按钮、切换按钮和复选框。共使用到了四种控件, 分别是 menu、button、toggleButton 和 checkBox。另外在使用这些控件时, 还涉及到了多个属性的使用, 如 label、supertip、imageMso 和 itemSize。其中关键控件的内容如下:

- **menu 控件:** 可以包含按钮、复选框、菜单和切换按钮等多种控件元素作为其子元素。其可以指定的属性分为两种, 其一是通用属性, 其二是特有属性。通用属性十分众多, 例如 label 属性可以指定标签、supertip 可以指定提示文字。menu 还具有一个其特有的属性即 itemSize, 该属性可以设置 normal 和 large 两种值。默认情况下为 normal, 此时其图标为小图标。当设置 large 时, 将使用大图标显示其图标。
- **button 控件:** 该控件不包含任何子元素, 大部分的可设置属性都为通用属性。示例中指定该控件的三个属性, label、supertip 和 imageMso 属性。imageMso 属性通过字符串的形式指定内置控件的图像名称。设置该属性后, 按钮的图标将使用该指定图标。
- **toggleButton 和 checkBox 控件:** 这两个控件具有一定的相似性。不同的是 toggleButton 控件可以由用户指定其显示的图标, 而 checkBox 控件无法指定。当用户选择了这两种控件后, 其图标将会切换到选中状态。toggleButton 按钮控件和 button、checkBox 两控件都具有相似性, 是两种控件的功能结合。

4. 编写代码

有关使用 XML 文件自定义 Excel 2007 工作簿的具体步骤可参见上例, 下面列出自定义菜单选项的 XML 代码。

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
  <ribbon startFromScratch="true">
    <officeMenu>
      <menu id="MyMenu" label="My Menu" supertip="My Menu"
        imageMso="DrawingCanvasScale" itemSize="large" >
        <button id="Mybtn1" label="My Button" supertip="My Button"
```

```
imageMso="DrawingObjectFormatDialog" />
<toggleButton id="MyTogBtn" label="My toggleButton"
    supertip="My toggleButton" imageMso="DrawingCanvasExpand"
/>

<checkBox id="Mychk" label="My CheckBox" supertip="My CheckBox" />
</menu>
</officeMenu>
</ribbon>
</customUI>
```

5. 运行结果

本例的主要功能是自定义 Office 的菜单按钮。在 Excel 2007 中，为用户提供了新的 Office 按钮。用户可以通过 XML 来自定义 Office 菜单按钮，如图 9.7 所示。同时，还可以为自定义的菜单按钮添加被选中的效果，如图 9.8 所示。

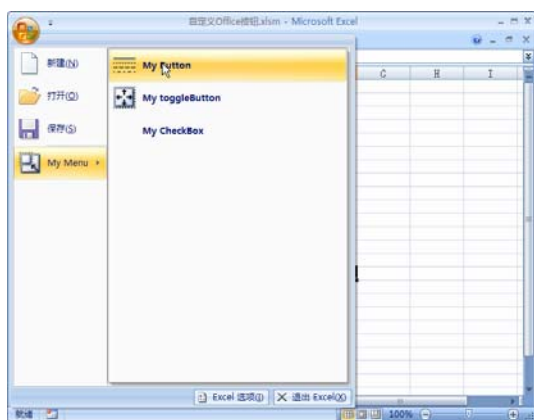


图 9.7 自定义 Office 按钮

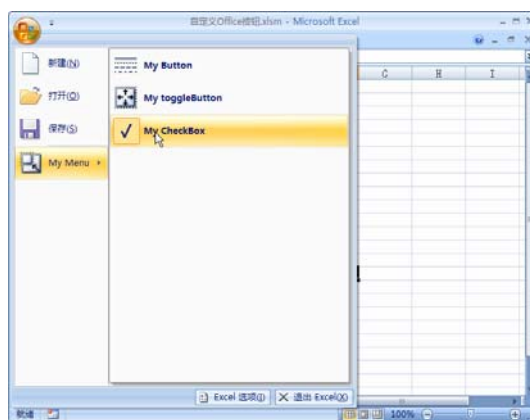


图 9.8 设置菜单的选中状态

同时，用户还可以为菜单设定提示信息，这里就不详细说明。

6. 程序分析

同样地，用户可以通过 IE 查看上面的 XML 代码，如图 9.9 所示。

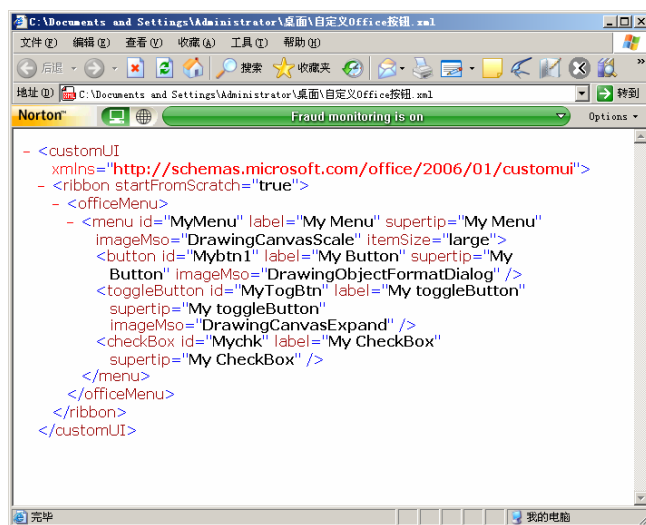


图 9.9 查看 XML 代码

案例 174 自定义快速访问栏

1. 功能说明

在 Excel 中，用户可以根据自己操作习惯来定义快速访问栏。将自己最常用的操作按钮添加到快速访问栏中。

2. 语法说明

对于大多数的用户而言，复制与粘贴操作是日常处理中最常用的功能。用户可以使用快捷键完成复制与粘贴操作。但是在进行特殊的复制与粘贴时，用户必须到开始选项卡的剪贴板组中找寻相应的命令。例如复制为图片、粘贴公式、粘贴值、转置粘贴以及粘贴图片等。本示例将把这些功能按钮通过 XML 代码将其显示在快捷访问工具栏。

4. 编写代码

有关使用 XML 文件自定义 Excel 2007 工作簿的具体步骤可参见上例，下面列出自定义快速访问工具栏有关的 XML 代码。

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
  <ribbon startFromScratch="true">
    <qat>
      <sharedControls>
        <button idMso="Copy" />
        <button idMso="PasteFormulas" imageMso="AccessTableContacts" />
        <button idMso="PasteValues" imageMso="BlogHomePage" />
        <button idMso="CopyAsPicture" imageMso="PhotoAlbumInsert" />
        <button idMso="PasteAsPicture" imageMso="PictureInsertFromFile" />
      </sharedControls>
    </qat>
  </ribbon>
</customUI>
```

```
</sharedControls>
</qat>
</ribbon>
</customUI>
```

5. 运行结果

本例的主要功能是自定义 Excel 的快速访问工具栏。在 Excel2007 界面的顶部、Office 菜单按钮的右侧包含了众多可以被直接使用的快捷访问按钮。用户可以通过 XML 来定制这些按钮，结果如图 9.10 所示。

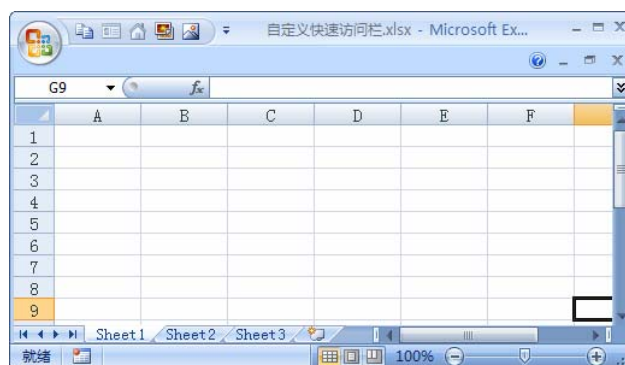


图 9.10 自定义快速访问工具栏

6. 程序分析

在上面的 XML 代码中，建立 button 按钮，使用 idMso 属性指定了按钮对应的内置功能。如果该项内置功能存在相应的图标且用户不指定该按钮的 imageMso 属性，系统将使用其默认图标。