



在 MS Word 2000 下 用 VBA 实现自定义词组功能

常新功 贾伟 (太原山西财经大学信息与管理学院 030006)

摘要	本文利用 VBA 和 Word 2000 中的对象编程, 提供了一种简便的实现自定义词组功能的方法。
关键词	VBA Selection 对象 自定义词组

“工欲善其事, 必先利其器”。对于常写文章的人来说, 一个好的输入法是必不可少的。特别是对于专业性较强的写作者来说, 其所用输入法的自定义词组功能显得尤为重要。然而, 市面上流行的输入法却大多不尽如人意, 有的没有提供自定义词组功能, 有的提供了, 但却存在这样那样的问题。事实上在 WORD2000 下, 我们可以利用 VBA 自己动手实现自定义词组功能, 从而加快录入速度, 提高工作效率。

在着手实现自定义词组功能之前, 我们先介绍一些将要用到的基础知识。

1 VBA 和 SELECTION 对象简介

1.1 VBA

VBA (Visual Basic for Applications) 是 Microsoft 公司的产品, 它嵌入在 Office 2000 中, 可用于定制和扩展 Office 2000 的功能。VBA 和 VB (Visual Basic) 是不同的, VB 是一种功能强大、简便易用的通用程序设计语言, 可用于开发基于 Windows 操作系统的应用软件和

Windows 组件; 而 VBA 可视为 VB 的子集, 它可以广泛地应用于 Microsoft 公司开发的各种软件尤其是 Office 中, 它与 Office2000 中的各个应用程序如 Word 2000、Excel 2000、Access 2000、PowerPoint 2000、Outlook 2000 底层的对象协同工作, 可实现许多 Office 本身不具有的功能。VBA 和 VB 使用相同的 Visual Basic 编辑器, 在 Word 2000 下, 逐层打开菜单“工具 | 宏 | Visual Basic 编辑器”即可进入 Visual Basic 编辑器, 在其中可以定义工程、窗体、模块, 编写、调试、运行代码, 其工作过程和工作方法几乎和 VB 完全一样。

1.2 SELECTION

Word 2000 中定义了许多对象, 如 Document 对象代表打开的文档, Paragraph 对象代表文档中的某个段落, Range 对象代表文档中的连续区域, Table 对象代表文档中的表格……。我们在此要用到的是 Selection 对象, 它代表选定的内容, 也就是我们通常所说的“块”。在 Word 中有多种定义块的方法, 最常用的是用鼠标定义块的方法, 按下鼠标左键

从一个位置拖动到另一个位置, 释放左键, 则这两个位置之间的文本就定义成了一个块; 用键盘定义块的方法是按下 shift 键, 然后移动光标移动键, 光标经过的地方即反白显示, 此即为定义好的块。Selection 对象的 text 属性表示当前块的文本内容, 如将“菜单”两个字定义为块, 此时 Selection.text 的值即为“菜单”。Selection 对象的 TypeText 方法可以插入文本, 如 Selection.TypeText(Text) 就将 Text 代表的内容输入到了当前位置 (现在没定义块) 或代替了当前块的内容 (如果现在定义了块的话)。

2 用 VBA 实现自定义词组功能

2.1 实现方案

我们分两步来实现自定义词组功能:

(1) 定义一个自定义词组。在文档中选定一小段文本作为我们欲定义的词组, 在此假设我们选定了“工欲善其事, 必先利其器”。接下来, 按下一个快捷键, 在此假设为 Alt+Ctrl+F4, 我们称这个快捷键为自定义词组快捷键, 稍后会用宏来实现

它。这时弹出一个对话框要求输入该词组的编码,假设输入“awuk”,这样一行文本“awuk,工欲善其事,必先利其器”就插入到了一个专门的文件中,在此我们假设其文件名为phrasedef.txt,存放在c:\my documents下。该词组定义完毕。

(2) 输入一个自定义词组。由于本文所述的自定义词组功能是用Office 2000自带的VBA来实现的,因此它与第三方提供的输入法无关。我们采取如下方案来输入一个已定义好的自定义词组:先在文档中直接输入该自定义词组的编码,比如awuk,然后选中它,按下一个快捷键,在此假设为Alt+Ctrl+F5,我们称这个快捷键为输入词组快捷键,此时,程序就会在phrasedef.txt文件中查到该编码对应的词组,比如“工欲善其事,必先利其器”,并将其输入到文档中,替代了刚才选中的内容(即该词组对应的编码)。

2.2 实现步骤

(1) 定义自定义词组功能的实现。我们通过宏来实现自定义词组功能:

第一步,打开菜单“工具|宏|录制新宏...”,此时弹出“录制宏”对话框,在该对话框中选择宏的名字为缺省名,假设为Macro1,然后选择“将宏保存到所有文档(Normal.dot)”,这样这个宏可适用于所有文档;另外,选择“将该宏指定到键盘”,在随后弹出的对话框中输入新快捷键Alt+Ctrl+F4,单击“指定”按钮,随后关闭该对话框。此时即进入了宏录制状态,什么都不要录制,直接选择停止录制按钮,退出宏录制状态。

第二步,打开菜单“工具|宏|宏...”,此时弹出“宏”对话框,在该对话框中选择宏Macro1,按下“编辑”按钮即进入了Visual Basic编辑器,在

代码窗口Sub Macro1()和End Sub之间加入如下代码:

```
Dim str1 As String '声明字符串变量str1,用于存放欲定义的自定义词组
```

```
Dim str2 As String '声明字符串变量str2,用于接收用户输入的该自定义词组的编码
```

```
Dim strFileName As String
```

```
Dim FileHandle%
```

```
str1 = Selection.Text '将文档中选中的内容存放到字符串变量str1中  
str2 = InputBox("请输入词组 [" & str1 & "] 的编码","自定义词组")  
'用输入框接收str1的编码,并将其存放到str2中
```

```
If str2 <> "" Then
```

```
strFileName = "c:\my documents \ phrasedef.txt"
```

```
FileHandle% = FreeFile '为该文件分配一个尚未使用的文件句柄号
```

```
Open strFileName For Append As #FileHandle% '以追加方式打开文件
```

```
MousePointer = vbHourglass '将鼠标指针形状设置为沙漏形状
```

```
Print #FileHandle%, str2 & "," & str1 '将数据按“编码,词组”格式存入文件
```

```
MousePointer = vbDefault '将鼠标指针形状恢复为原始形状
```

```
Close #FileHandle% '关闭文件
```

```
End If
```

(2) 输入自定义词组功能的实现。我们通过定义另一个宏来实现此功能:

第一步,打开菜单“工具|宏|录制新宏...”,在弹出的“录制宏”对话框中选择宏的名字为缺省名,假设为Macro2,然后选择“将宏保存到所有文档(Normal.dot)”,选择“将宏指定到键盘”,在随后弹出的对话框中

输入新快捷键Alt+Ctrl+F5,单击“指定”按钮,关闭该对话框。此时即进入了宏录制状态,同样什么都不录制,直接选择停止录制按钮,退出宏录制状态。

第二步,打开菜单“工具|宏|宏...”,在弹出的“宏”对话框中选择宏Macro2,按下“编辑”按钮,进入Visual Basic编辑器,在代码窗口Sub Macro2()和End Sub之间加入如下代码:

```
Dim strSel As String '声明字符串变量strSel,用于存放当前文档中选中的内容
```

```
Dim strCode As String 'strCode用于存放文件某一行中“,”前面的编码部分
```

```
Dim strPhrase As String 'strPhrase用于存放文件某一行中“,”后面的词组部分
```

```
Dim lngPos As Long 'lngPos用于存放文件某一行中“,”所处的位置
```

```
Dim strFileName As String
```

```
Dim strbuffer As String
```

```
Dim FileHandle%
```

```
strSel = Selection.Text '将当前文档中选中的内容存放到字符串变量strSel中,它事实上就是欲输入的词组的编码
```

```
strFileName = "c:\my documents \ phrasedef.txt"
```

```
FileHandle% = FreeFile '为文件phrasedef.txt分配一个尚未使用的文件句柄号
```

```
Open strFileName For Input As #FileHandle% '以输入方式打开文件  
MousePointer = vbHourglass
```

```
Do While Not EOF(FileHandle%)
```

```
Line Input #FileHandle%,
```

```
strbuffer '从文件中每次取一行数据到strBuffer中
```

Application

lngPos = InStr(strbuffer, ",") '定位在该行数据中“,”所处的位置
strCode = Left(strbuffer, lngPos - 1) '“,”前面的数据即编码存放到 strCode 中

strPhrase = Right(strbuffer, Len(strbuffer) - lngPos) '“,”后面的数据即词组存放到strPhrase中
If strCode = strSel Then '如用户输入的编码和本行数据的编码相同

Selection.Text = strPhrase '将当前文档中选中的内容换为相应的词组

Exit Do
End If
Loop
MousePointer = vbDefault
Close #FileHandle%

整个操作完毕。容易看出，我们

是通过定义两个空的宏，然后在其中插入 VBA 代码来实现自定义词组功能的。接下来再看一下它的使用方法：假设我们要将“所见即所得”定义为一个词组，先在文档中选中这几个字，即将这几个字做成一个块，然后按 Alt+Ctrl+F4，此时出现一个对话框让你输入“所见即所得”的编码，假设输入“rmvt”，词组定义完毕；以后当我们输入“所见即所得”这个词组时，可以这样输入，直接在文档中而不是在输入条中输入其编码“rmvt”，选中它，然后按 Alt+Ctrl+F5，“所见即所得”即输入并覆盖了其编码“rmvt”。

3 结束语

应该看到，本文所述的自定义词组功能有一定的局限性，它只适用于支持 VBA 的软件中。但出于

它实现简单、方便，且不存在其他输入法广泛存在的随着自定义词组的增加重码率增加等问题，因而具有一定的可行性和可用性。另外，本文所述的自定义词组功能还有许多细节问题可以改进，比如，可以设计一个输入条以接收用户的输入，从而改进自定义词组的输入方式，这可以通过在 Visual Basic 编辑器中定义用户窗体来实现，实现起来并不复杂，在此就不再赘述了。■

参考文献

- 1 王川、阮兴平、于平剑、何建波编著，《Office 2000 VBA 一册通》，北京人民邮电出版社，2000.1。
- 2 Bob Reselman, Richard Peasley 著，何健辉、董方鹏、冯毅译，《实用 Visual Basic 6 教程》，北京清华大学出版社，2001.3。