**Word编程代码集**

序

本书是作者在EXCELHOME(http：//club.excelhome.net/index.asp)Word版中的部分原自创作品，其中的部分代码是作者耗费大量精力所创，在已知的国内外各WORD论坛中所未见。读者在阅读本书相应代码时，可从相关链接中进行对原帖的查阅，以便能够更好地理解和掌握代码的含义和适用范围。

作者整理此书的目的之一，就是希望让有一定WORD基础的朋友能够通过此书的讲解，以提高对WORD以及MS（Microsoft ）的认识和操作技能，并希望本书能对想学习VBA和正在学习VBA的读者有所裨益。

在阅读本书之前，作者先阐述一下VBA(Visual Basic for Applications)的作用原理：VBA是捆绑在Appliation对象（此处则指Word.Application，简称Word）的一个后台程序;VBE（Visual Basic Editor）是指编辑VBA的一个程序/编辑器(在WORD中按下ALT+F11即可进入)，从对象角度看，有Application.VBE(即VBE是附属于Application对象的一个对象)，从工程角度看，有ActiveDocument.VBProject（当前文档的VBA工程）。我们知道，Microsoft 系统产品是以Windows（广义，非单指WIN系统）著称，是泛指以窗口型的可视化程序，用户与电脑通过程序进行数据交换和人机对话，用户所有的前台（直接用鼠标、键盘等）和后台（编程）操作，都是面向对象的操作。因此正确理解对象的概念、集合、属性、方法是非常必要的。从大范围讲，Application（应用程序）是一个大对象(最顶层)，任何允许用户操作的地方都存在指定的对象，比如常见的标题栏名称（Application.Caption），最大化最小化按钮（Application.WindowState），所有的菜单、工具栏、命令等等都是一个对象，用户最常用的是Selection对象，即选中的内容，Word中是作为Selection对象来处理的，如选中的文字，选中的表格，选中的图形等等，大到Application对象（最顶层），小到一个字符（Character）甚至一个光标，对于VBA而言，都是一个对象。根据对象分工不同，对象还有父对象、子对象等等。我们通常编程，可以以不同的方式访问对象、修改对象的属性或者指定对象进行特定的动作等。在WORD中，最重要的对象是Selection对象和Range对象，相当于Excel中的ActiveCell和Range对象。

下面我们来讨论一下为什么要编程：Word程序为用户设计了许许多多具有普遍规律的对象的操作方法和属性修改，对于常规的规范化操作，只要用户充分了解了WORD中的操作规律并进行了规范操作，使用WORD的前台功能，已经能基本解决常规问题；但对一些不具有普遍规律并且用户自定义的非规范内容的长时间多次数反复操作，则不可能提供一个完全的、千遍一律的解决方法，那么WORD（Microsoft Office）是如何来解决这个问题的呢？Microsoft Word（Office）在这里为我们提供了强有力的编程手段，编程语言正是解决虽然不具有普遍操作意义的命令，但能使用少数用户的部分WORD功能强化的代码；对于简单的重复组合命令，我们可以通过录制宏的方法进行录制，也可以在录制结束后进行适当的简化，但基本以Selection对象为主，而且录制的宏中每一个对象的所有属性几乎全部被记录，整个代码非常大，效率就低；更为主要的是录制的宏中，对于判断性结构语句，循环语句，函数等等的，是没有提供记录的，所以，要使我们的宏适用于用户、适用于一个特定的操作过程，就需要使用编程来完成。

认识宏安全性：宏是WORD（Microsoft Office）提供给用户或者编程人员的一个特定的后台操作环境中的一组代码，是用来完成指定操作的一个过程（大到工程），从广义角度讲，任何不是用户愿意看到的结果的代码，都可以称之为病毒；从狭义角度讲，病毒是专门用来危害用户的操作系统、危害用户的应用程序并使用户在无知情况下进行非自愿性操作的恶意代码并可能具有自身的复制和传播与变性。对Word而言，我们可以设定宏安全性，这样可以禁止宏的运行或者选择性地运行一些宏，从上述内容中我们也可以看到，如果不用宏，我们很多的自动化、复杂的工作等等都会因之不能使用。因此，宏是一柄双刃剑。但随着MS的防范机制的增加、用户水平的增高、用户杀毒软件的及时有效使用，都会避免恶意病毒（代码）的入侵。退一步而言，以VBA的形式编写的宏病毒，影响是有限的。因此，读者首先不要产生谈宏（Macro）色变的想法，要宏为我所用，同时也提醒编程人员，编程要有所为，有所不为。

了解宏的作用原理：WORD应用程序的所有内置命令、模块、过程，不会在VBE中出现，而所有的宏（录制宏、复制宏、代码编程等）均寄生于文档的VBProject中，当用户结合事件触发后或者运行后，才能作用于特定对象，完成特定操作或者返回特定数据。

本书中可能涉及的对象，主要有：节(Sections)的循环，段落中(Paragraphs)的循环，句子(Sentences)的循环，词组中(Words)的循环，字(Characters)的循环，表格(Tables)的循环，单元格(Tables(Item).Range.Cells)的循环，自选图形(Shapes)的循环，域(Fields)的循环，书签(Bookmarks)中的循环等等，函数的应用、选择性分支语句、判断语句、错误处理、类模块的使用、用户窗体的使用、数组的应用、Automation等等，不一一列举，读者可根据自身情况，逐一进行渐进式学习。

对于书中所涉及的一些代码可能读者还会有更好的、更简单的方法，也或者有些代码还会存在这样那样的问题，这都有待于读者的反馈与交流。在编程上，我们常说的一句话是：没有最好，只有更好；另外，我也可能会更新部分代码，请注意相关链接。

另外由于Office 版本号的不同，其中的属性对象方法等也不尽相同，但可向下兼容。

如读者有任何疑问可发送邮件到 shourou\_8@hotmail.com，我将尽快给予答复，有MSN的朋友也可以直接通过MSN进行交流。

**[001]空白段落的删除**：

功能简介：可以对指定长度的段落进行删除，当LEN=1时可对空白段落进行删除。

Sub DelBlank()

    Dim i As Paragraph， n As Long

    Application.ScreenUpdating = False           ’关闭屏幕刷新

    For Each i In ActiveDocument.Paragraphs         ’在活动文档的段落集合中循环

        If Len(i.Range) = 1 Then     ’判断段落长段，此处可根据文档实际情况

            i.Range.Delete          ’进行必要的修改可将任意长度段落删除

            n = n + 1           ’计数

        End If

    Next

    MsgBox "共删除空白段落" & n & "个!"

    Application.ScreenUpdating = True       ’恢复屏幕刷新

End Sub

**[002]以指定字符重新划分段落并插入时间序列数**

Sub CreateParagraph()

    Dim I As Long， N As Integer

    On Error Resume Next     ’忽略错误

    Application.ScreenUpdating = False     ’关闭屏幕更新

    With ActiveDocument

        ’将文档中所有段落标记删除

        .Content.Find.Execute FindText：="^p"， ReplaceWith：=""， Replace：=wdReplaceAll

        For I = 0 To .Content.End Step 10    ‘以10个字符位置(包括非打印字符)为步长循环

            ‘每段十个字符部分分成段落(注意插入的段落标记也是一个字符)

            .Range(I， I + 10 + N).InsertAfter Chr(13)

            N = N + 1    ‘计算插处的段落标记个数

        Next

    End With

    Application.ScreenUpdating = True    ‘恢复屏幕更新

    InsertTimer

End Sub

Sub InsertTimer()

    Dim I As Paragraph， N As Integer， TimeStr As String

    On Error Resume Next    ‘忽略错误

    Application.ScreenUpdating = False    ‘关闭屏幕更新

    For Each I In ActiveDocument.Paragraphs    ’在文档新的段落中循环

        If N < 10 Then     ’<10，TimeStr的分钟值为5(保持两位数05)

            TimeStr = "[00：0" & N & ".00]"

        ElseIf N = 60 Then    ‘N=60时时间数进一并保持该数据(不再向上)

            TimeStr = "[01：00.00]"

            N = 0

        Else    ’TimeStr的分钟数照计(两位数)

            TimeStr = "[00：" & N & ".00]"

        End If

        I.Range.InsertBefore TimeStr    ‘每个段前插入时间数值

        N = N + 5    ’以5为步长累加

    Next

    Application.ScreenUpdating = True    ’恢复屏幕更新

End Sub

**[003]段落样式与格式的应用**

功能简介：由于手动录入的段落编号不能被WORD所识别，为以后的样式与格式的设置以及目录索引等带来一系列的问题，本代码即是将其转换为指定样式的过程。

Sub Sample()

    Dim i As Paragraph， MyStr As String

    Application.ScreenUpdating = False

    MyStr = "一二三四五六七八九十"    ‘假定为手动加注每个段落开头为中文大写数字

    For Each i In Me.Paragraphs

        If i.Range Like "（#）\*" = True Then

            i.Style = wdStyleHeading9    ‘标题9是以（1）等开头的数字

        ElseIf i.Range Like "#.#.#.#\*" = True Then

            i.Style = wdStyleHeading8    ’标题8是以1.1.1.1的形式开头的段落

        ElseIf i.Range Like "#.#.#\*" = True Then

            i.Style = wdStyleHeading7    ‘标题7是以1.1.1的形式开头的段落

        ElseIf i.Range Like "#.#\*" = True Then

            i.Style = wdStyleHeading6    ‘标题6是以1.1形式开头的段落

        ElseIf InStr(MyStr， Me.Range(i.Range.Start， i.Range.Start + 1).Text) > 0 Then

            i.Style = wdStyleHeading5    ’标题5是以一等形式开头的段落

        Else

            i.Style = wdStyleNormal    ’其它为正文样式

        End If

    Next

    Application.ScreenUpdating = True

End Sub

**[004]根据预定义段落进行段落样式的设置和插入目录**

功能简介：对于网上复制的一些非正规编排的文档，没有大纲级别，也没有很好地样式格式以区分，利用此功能，可以根据先前的手动目录，更改为自动生成的目录，便于文档管理。

Sub Contents()

    Dim I As Paragraph， N As Byte， A As Byte， B As Byte， X As Long， DelRange As Range

    Application.ScreenUpdating = False

    A = 2

    B = 13

    With ActiveDocument

        For Each I In .Paragraphs    ’在段落中循环

            X = X + 1    ‘计数

            For N = A To B    ‘进入文档第二段落到第十三段落间的循环

                If X > B Then

                    If I.Range = .Paragraphs(N).Range Then

                        I.Style = .Styles(wdStyleHeading1)

                        A = A + 1    ’累计

                    End If

                End If

            Next

        Next

        Set DelRange = Range(.Paragraphs(2).Range.Start， .Paragraphs(13).Range.End)

        DelRange.Delete    ’删除原文档的第二~第十三个段落

        .Paragraphs(2).Range.Select        ’插入/引用/索引与目录

        .TablesOfContents.Add Range：=Selection.Range， RightAlignPageNumbers：= \_

               True， UseHeadingStyles：=True， UpperHeadingLevel：=1，LowerHeadingLevel：=3，\_

IncludePageNumbers：=True， AddedStyles：=""，\_

                     UseHyperlinks：=True， HidePageNumbersInWeb：=True

        .TablesOfContents(1).TabLeader = wdTabLeaderDots

        .TablesOfContents.Format = wdIndexIndent

    End With

    Application.ScreenUpdating = True

End Sub

**[005]表格集合中的循环与对单元格边框的设置**

Sub Example()

    Dim i As Table， N As Integer

    On Error Resume Next    ’忽略错误

    Application.ScreenUpdating = False    ’关闭屏幕更新

    For Each i In ActiveDocument.Tables    ’在表格中循环

        With i

            .Style = "列表型 4"    ’将所有表格设置为"列表型4"的样式

            With .Borders    ‘边框

                .InsideLineStyle = wdLineStyleSingle    ’设置内部边框线条

            End With

            With .Rows(1).Borders(wdBorderBottom)    ‘第一行的底边框

                .LineStyle = wdLineStyleDouble    ‘双线型

                .LineWidth = wdLineWidth050pt

                .Color = wdColorAutomatic

            End With

            If .Rows.Count > 1 Then    ’ 如果表格行数大于1

                If Len(.Cell(2， 1).Range) <= 2 Then    ’如果第二行第一列不为空

                    With .Rows(2).Shading    ’设置底纹

                        .Texture = wdTextureNone    ’无底底纹

                        .ForegroundPatternColor = wdColorAutomatic

                        .BackgroundPatternColor = wdColorGray125

                    End With

                End If

            End If

            For N = 2 To .Columns.Count    ’从第二列到最后一列

                .Columns(N).Select    ’单元格对齐方式为中部居中

                Selection.ParagraphFormat.Alignment = wdAlignParagraphCenter

                Selection.Cells.VerticalAlignment = wdCellAlignVerticalCenter

            Next N

        End With

    Next i

    Application.ScreenUpdating = True

End Sub

**[006]书签、数组与排序**

功能简介：本示例解决的问题是对所选表格中的单元格内文本进行不重复值排序功能，在此例中，是写在文档中，当然可以应用于组合框中。示例中以书签功能的自动同名替换功能来去除重复值，然后在数组中以冒泡排序法进行排序，再以段落方式插入到文档中，当前仅以排序为例，在段落中也可以进行直接排序。（表格/排序功能），此范例其实是如何按照排序顺序并去除重复值的文本项写入组合框中的前期应用。

Option Compare Text   ’以文本方式比较不区分大小写

Sub Sortly()

    Dim n As Cell， strCell() As String， MyStr As String， intCount As Integer， BkMark As Bookmark

    Dim First As Integer， Last As Integer， i As Integer， j As Integer， Temp As String

    intCount = 0

    With Selection

        For Each n In .Tables(1).Range.Cells    ‘在表格的单元格中循环(本示例中表格只有一列)

            MyStr = Me.Range(n.Range.Start， n.Range.End - 1)    ‘取得每一个单元格中的文本

            If MyStr Like "#\*" = True Then    ‘鉴别文本有效性

                MsgBox "此数据不能被程序识别，请勿在其首以任何数字形式出现!" &  vbCrLf  & """" &  MyStr  & """"

                Exit Sub

            Else

                Me.Bookmarks.Add Name：=MyStr    ‘增加为书签(目的是将单元格重复数据删除，不在表格中进行

            End If

        Next

        ReDim strCell(Me.Bookmarks.Count - 1)    ’声明一个动态数组

        For Each BkMark In Me.Bookmarks

            strCell(intCount) = BkMark.Name

            intCount = intCount + 1

        Next

        First = LBound(strCell)    ‘取得数组下标

        Last = UBound(strCell)    ‘取得数组上标

        For i = First To Last - 1    ’在数组中循环取值

            For j = i + 1 To Last    ‘冒泡法排序

                If strCell(i) > strCell(j) Then

                    Temp = strCell(j)

                    strCell(j) = strCell(i)

                    strCell(i) = Temp

                End If

            Next j

        Next i

        MyStr = ""   ’初始化MyStr变量

        Temp = ""    ’初始化Temp变量

        .EndKey Unit：=wdStory    ‘移到最后

        .InsertAfter Chr(13)    ‘插入一个回车符(段落)

        For x = First To Last    ‘将数组数据写到文档中

            Temp = strCell(x) & Chr(13)

            MyStr = MyStr & Temp     ’累加在内存中

        Next

        .InsertAfter MyStr ‘插入文档中

    End With

End Sub

**[007]WORD文档中文词组频率统计**

功能说明：对文档中的中文字符(词组)出现频率在二次以二次以上的进行列表。

Sub WordsCount()

    Dim i As Range， aVar As Variable， aString As String， MyString As String

      MsgBox "受文档字数和可用内存以及WORD自身限制，您的操作可能会需要一段时间，" & vbCrLf & \_

           "也许会出现可用内存不足的情况，您可能需要重启WORD以便接着下一次的工作!"， vbOKOnly + vbExclamation， "Warnning"

    VarClear    ‘清空文档变量

    BkClear    ‘清空书签

    For Each i In Me.Words    ‘词中循环

        Me.UndoClear    ’清空撤消，以便留有足够内存

        If i.Characters.Count > 1 Then    ‘按中文习惯超过二个字或者两个字者组为词，如果去掉这句，可以对字\词频次列表。

            ’判断是否为中文字符

            If Asc(i.Characters(1)) < -2050 And Asc(i.Characters(1)) > -20319 Then

                ’如果已存在该书签(相当于第二次以上出现该词/字

                If Me.Bookmarks.Exists(i.Text) = True Then

                    On Error Resume Next       ’添加文档变量

                    Me.Variables.Add Name：=i.Text      ’设置错误陷阱

                    If Err.Number <> 0 Then

                        Err.Clear    ‘清除错误

                        ‘将次数累加写入

                        Me.Variables(i.Text).Value = Me.Variables(i.Text).Value + 1

                    Else

                        ’首次写入文档变量时，其初始值为2

                        Me.Variables(i.Text).Value = 2

                    End If

                Else

                    ’添加新书签

                    Me.Bookmarks.Add Name：=i.Text

                End If

            End If

        End If

    Next

    Application.ScreenUpdating = False    ’关闭屏幕更新

    With Selection

        .EndKey unit：=wdStory    ’移到文档末尾

        ‘作一个区分标记

        .InsertAfter vbCrLf & "-----二次以上（含二次）词数频次统计列表-----" & vbCrLf

        .EndKey unit：=wdStory    ’移到文档末尾

        For Each aVar In Me.Variables    ‘在文档变量中循环

            ‘插入文档中

            aString = """" & aVar.Name & """出现频次：" & vbTab & aVar.Value & vbCrLf

            MyString = MyString & aString    ‘文本累加写入内存变量中，以加速运行

        Next

        .InsertAfter MyString

        MyString = ""    ’释放变量

        ‘根据出现频次排序，以降序方式进行

        .Sort FieldNumber：="域 1"， SortFieldType：= \_

                wdSortFieldNumeric， SortOrder：=wdSortOrderDescending

    End With

    Me.UndoClear    ’清空撤消

    VarClear    ‘清空文档变量

    BkClear    ’清空书签

    Application.ScreenUpdating = True    ’恢复屏幕更新

End Sub

Sub VarClear()

    Dim V As Variable

    For Each V In Me.Variables

        V.Delete    ‘删除文档变量

    Next

End Sub

Sub BkClear()

    Dim BK As Bookmark

    Me.UndoClear    ’清空撤消

    For Each BK In Me.Bookmarks

        BK.Delete    ‘删除书签

        Me.UndoClear

    Next

End Sub

‘以下方法较上述代码运行速度更快，更有效，并能满足不同用户统计的需要。

Sub WordsCountTwo()

    Dim i As Range， aVar As Variable， aString As String， MyString As String， YNC As Byte

    ‘友情提示

    MsgBox "受文档字数和可用内存以及WORD自身限制，您的操作可能会需要一段时间!"， vbOKOnly \_

                                                  + vbExclamation， "Warnning"

    YNC = MsgBox("按YES统计字的出现频次，按NO统计词的出现频次，按CANCEL统计字与词!"， vbYesNoCancel + vbInformation)

    Select Case YNC

    Case vbYes

        For Each i In Me.Characters

            If Asc(i) < -2050 And Asc(i) > -20319 Then

                If MyString = "" Then GoTo GNY

                If InStr(MyString， i.Text & "，") = 0 Then

GNY：                     aString = i.Text & "，"

                    MyString = MyString & aString

                Else

                    On Error Resume Next    ‘忽略错误

                    Me.Variables.Add Name：=i.Text    ‘添加文档变量

                    If Err.Number <> 0 Then    ‘设置错误陷阱

                        Err.Clear    ’清除错误

                        ‘将次数累加写入

                        Me.Variables(i.Text).Value = Me.Variables(i.Text).Value + 1

                    Else

                        ’首次写入文档变量时，其初始值为2

                        Me.Variables(i.Text).Value = 2

                    End If

                End If

            End If

        Next

    Case vbNo

        For Each i In Me.Words    ‘词中循环

            If i.Characters.Count > 1 Then    ‘按照中文习惯为二个以上为词组

                ‘判断是否为中文字符

                If Asc(i.Characters(1)) < -2050 And Asc(i.Characters(1)) > -20319 Then

                    If MyString = "" Then GoTo GNN    ‘循环初始阶段跳至GNN行标签

                    If InStr(MyString， i.Text & "，") = 0 Then

GNN：                         aString = i.Text & "，"    ‘加入"，"分隔符以便精确定位

                        MyString = MyString & aString

                    Else

                        On Error Resume Next    ‘忽略错误

                        Me.Variables.Add Name：=i.Text    ‘添加文档变量

                        If Err.Number <> 0 Then    ‘设置错误陷阱

                            Err.Clear    ’清除错误

                            ‘将次数累加写入

                            Me.Variables(i.Text).Value = Me.Variables(i.Text).Value + 1

                        Else

                            ‘首次写入文档变量时，其初始值为2

                            Me.Variables(i.Text).Value = 2

                        End If

                    End If

                End If

            End If

        Next

    Case vbCancel

        For Each i In Me.Words    ‘词中循环

            ‘判断是否为中文字符

            If Asc(i.Characters(1)) < -2050 And Asc(i.Characters(1)) > -20319 Then

                If MyString = "" Then GoTo GNC    ’循环初始阶段跳至GNC行标签

                If InStr(MyString， i.Text & "，") = 0 Then

GNC：                     aString = i.Text & "，"    ‘加入"，"分隔符以便精确定位

                    MyString = MyString & aString

                Else

                    On Error Resume Next    ‘忽略错误

                    Me.Variables.Add Name：=i.Text    ‘添加文档变量

                    If Err.Number <> 0 Then    ‘设置错误陷阱

                        Err.Clear    ‘清除错误

                        ‘将次数累加写入

                        Me.Variables(i.Text).Value = Me.Variables(i.Text).Value + 1

                    Else

                        ‘首次写入文档变量时，其初始值为2

                        Me.Variables(i.Text).Value = 2

                    End If

                End If

            End If

        Next

    End Select

    aString = ""： MyString = ""    ‘重新初始化变量

    Application.ScreenUpdating = False    ‘关闭屏幕更新

    With Selection

        .EndKey unit：=wdStory    ’移到文档末尾

        ‘作一个区分标记

        .InsertAfter vbCrLf & "-------词数频次统计列表-------" & vbCrLf

        .EndKey unit：=wdStory    ‘移到文档末尾

        For Each aVar In Me.Variables    ‘在文档变量中循环

            ‘插入文档中

            aString = """" & aVar.Name & """出现频次：" & vbTab & aVar.Value & vbCrLf

            MyString = MyString & aString    ‘文本累加写入内存变量中，以加速运行

        Next

        .InsertAfter MyString

        ’根据出现频次排序

        .Sort FieldNumber：="域 1"， SortFieldType：= \_

              wdSortFieldNumeric， SortOrder：=wdSortOrderDescending

    End With

    VarClear    ‘清空文档变量

    Me.UndoClear    ’清空撤消

    Application.ScreenUpdating = True    ‘恢复屏幕更新

End Sub

Sub VarClear()

    Dim V As Variable

    For Each V In Me.Variables

        V.Delete    ’删除文档变量

    Next

End Sub

**[008]查找与替换的基本代码用法之一**

Sub Example()

    With ActiveDocument.Content.Find

        .ClearFormatting    ’清除格式设置

        .Font.Name = "华文细黑"    ‘查找的字体格式

        With .Replacement    ’替换条件

            .ClearFormatting    ‘清除格式设置

            .Font.Name = "微软雅黑"    ‘替换成微软雅黑

        End With

        .Execute FindText：=""， ReplaceWith：=""， Format：=True， \_

                 Replace：=wdReplaceAll    ‘是格式替换，全部替换

    End With

End Sub

**[009]查找与替换的基本代码用法之二**

Sub Example()

    Dim FindChar As String， Fcount As Integer， RepChar As String

    On Error Resume Next

    Application.ScreenUpdating = False    ‘关闭屏幕更新

    FindChar = "《"

    RepChar = "["

    With ActiveDocument.Content.Find    ‘此处针对全文档

        Do While .Execute(findtext：=FindChar) = True    ‘如果发现

            Fcount = Fcount + 1    ‘计数器

        Loop

        If MsgBox("文档中共发现了" & Fcount & "个" & FindChar & vbCrLf \_

                & "，按Yes键将进行下一步的替换工作，按No取消"， vbYesNo + vbInformation) = vbYes Then

            .Execute findtext：=FindChar， Wrap：=wdFindContinue， replacewith：=RepChar， Replace：=wdReplaceAll

        End If

    End With

    Application.ScreenUpdating = True    ‘恢复屏幕更新

End Sub

**[010]查找与替换的基本代码用法之三(批量替换)**

功能简介：同时进行多个查找与替换，支持非通配符下的特殊字符的替换。

比如，适用于ISO文件，因组织机构调整，对所有原有部门一次输入后替换为新部门。

查找的各个内容之间，用英文逗号分隔（"，"），查找数量不限。

替换的各个内容之间，用英文逗号分隔（"，"），替换数量必须等同于查找数量，如是删除某个查找内容，替换中键入""(空空)

Private Sub Document\_Close()

    On Error Resume Next

    Application.CommandBars("Edit").Controls("多个替换").Delete    ’恢复原有菜单

End Sub

Private Sub Document\_Open()

    On Error Resume Next

    Dim NewButton As CommandBarButton

    CustomizationContext = ActiveDocument    ‘将自定义组合键和工具命令保存于活动文档中

    ‘指定CTRL+F为键盘快捷方式

    KeyBindings.Add wdKeyCategoryMacro， "MySub"， BuildKeyCode(wdKeyControl， wdKeyF)

    ‘指定F5为快捷方式

    KeyBindings.Add wdKeyCategoryMacro， "MySub"， BuildKeyCode(wdKeyF5)

    Application.CommandBars("Edit").Controls("多个替换").Delete    ‘预防性删除

    Set NewButton = Application.CommandBars("Edit").Controls.Add(Type：=msoControlButton， Before：=11)

    With NewButton

        .Caption = "多个替换"    ‘命令名称

        .FaceId = 100    ‘命令的FaceId

        .Visible = True    ‘可见

        .OnAction = "MySub"    ’指定响应过程名

    End With

End Sub

Sub MySub()

    UserForm1.Show

End Sub

Sub ComReset()   ‘恢复默认设置

    Application.CommandBars("Edit").Reset

End Sub

用户窗体-UserForm1

Private Sub CommandButton1\_Click()

    Me.TextBox1 = ""

    Me.TextBox2 = ""

    Me.TextBox1.SetFocus

End Sub

Private Sub CommandButton2\_Click()

    Dim MyFind() As String， MyRep() As String， i As Integer， aStory As Variant

    On Error Resume Next

    ‘检查是否为空

    If Me.TextBox1 = "" And Me.TextBox2 = "" Then Exit Sub

    ‘定义两个数组，以"，"分隔

    MyFind = Split(Me.TextBox1， "，")

    MyRep = Split(Me.TextBox2， "，")

    If UBound(MyRep) <> UBound(MyFind) Then

        ’如果两个文本框的分隔数目不一致，提示

        MsgBox "替换的数目与查找数目不一致!"， vbExclamation + vbOKOnly， "Warnning"

        Me.TextBox2.SetFocus

        Exit Sub

    End If

    Application.ScreenUpdating = False

    With ActiveDocument

        For i = 0 To UBound(MyFind)    ‘一个从下标为0的循环替换

            For Each aStory In .StoryRanges    ‘在文档的各个文字部分

                ‘如果是""，则相当于删除原查找内容

                aStory.Find.Execute findtext：=MyFind(i)，\_

                     replacewith：=VBA.IIf(MyRep(i) = """"""， ""， MyRep(i))， Replace：=2

                ‘如果有下一节中相同内容文字部分，也进行替换

                If Not aStory.NextStoryRange Is Nothing Then \_

                   aStory.NextStoryRange.Find.Execute findtext：=MyFind(i)， \_

                   replacewith：=VBA.IIf(MyRep(i) = """"""， ""， MyRep(i))， Replace：=2

            Next

        Next

    End With

    Application.ScreenUpdating = True

    Unload Me    ’卸载窗体

End Sub

Private Sub UserForm\_Initialize()

    Me.Caption = "多文本替换操作"

    Me.TextBox1.SetFocus

    Me.CommandButton2.Default = True

End Sub

**[011]查找与替换的基本代码用法之四-全文件夹替换**

功能简介：批量多文件（全文件夹）的多文本一次性替换操作。

运行本程序后，先输入需查找和与之对应的替换的文本，然后点击“选择文件夹”，您可以找到指定的文件夹中的部分或者所有文件，注意，您需要全选文件（CTRL+A），或者使用SHIFT/CTRL配合鼠标键选取多个文件），确定后自动进行批量替换。

Private Sub Document\_Open()

    Application.Windows(ThisDocument.Name).Visible = False

    MySub

End Sub

Sub MySub()

    UserForm1.Show

End Sub

用户窗体

Private Sub CommandButton1\_Click()

    Me.TextBox1 = ""

    Me.TextBox2 = ""

    Me.TextBox1.SetFocus

End Sub

Private Sub CommandButton2\_Click()

    Dim MyFind() As String， MyRep() As String， i As Integer， aStory As Variant

    Dim MyDialog As FileDialog， vrtSelectdeItem As Variant， Doc As Document

    On Error Resume Next

    ‘检查是否为空

    MsgBox "请先正确录入查找与对应替换的内容，以英文逗号分隔" & vbCrLf & \_

           "在选定文件夹中，您可以全选或部分选定文件(CTRL/SHIFT)+鼠标单击"， vbInformation

    If Me.TextBox1 = "" And Me.TextBox2 = "" Then Exit Sub

    MyFind = Split(Me.TextBox1， "，")

    MyRep = Split(Me.TextBox2， "，")

    If UBound(MyRep) <> UBound(MyFind) Then

        ’如果两个文本框的分隔数目不一致，提示

        MsgBox "替换的数目与查找数目不一致!"， vbExclamation + vbOKOnly， "Warnning"

        Me.TextBox2.SetFocus

        Exit Sub

    End If

    ‘定义一个文件夹选取对话框

    Set MyDialog = Application.FileDialog(msoFileDialogFilePicker)

    With MyDialog

        .Filters.Clear    ‘清除所有文件筛选器中的项目

        .Filters.Add "所有 WORD 文件"， "\*.doc"， 1    ‘增加筛选器的项目为所有WORD文件

        .AllowMultiSelect = True    ‘允许多项选择

        If .Show = -1 Then    ‘确定

            Application.ScreenUpdating = False

            For Each vrtselecteditem In .SelectedItems    ‘在所有选取项目中循环

                Set Doc = Documents.Open(FileName：=vrtselecteditem， Visible：=False)

                ‘定义两个数组，以"，"分隔

                With Doc

                    For i = 0 To UBound(MyFind)    ‘一个从下标为0的循环替换

                        For Each aStory In .StoryRanges    ’在文档的各个文字部分

                            ‘如果是""，则相当于删除原查找内容

                            aStory.Find.Execute findtext：=MyFind(i)， \_

                                                replacewith：=VBA.IIf(MyRep(i) = """"""， ""， MyRep(i))， Replace：=2

                            ’如果有下一节中相同内容文字部分，也进行替换

                            If Not aStory.NextStoryRange Is Nothing Then \_

                               aStory.NextStoryRange.Find.Execute findtext：=MyFind(i)， \_

                               replacewith：=VBA.IIf(MyRep(i) = """"""， ""， MyRep(i))， Replace：=2

                        Next

                    Next

                    Doc.Close True

                End With

            Next vrtselecteditem

        End If

    End With

    Application.ScreenUpdating = True

    Unload Me    ‘卸载窗体

End Sub

Private Sub UserForm\_Initialize()

    Me.Caption = "多文本替换操作"

    Me.TextBox1.SetFocus

    Me.CommandButton2.Default = True

End Sub

Private Sub UserForm\_QueryClose(Cancel As Integer， CloseMode As Integer)

    ThisDocument.Close False

End Sub

**[012]判断光标所在行是否有手动分页符**

Sub Example()

    Dim MyRange As Range， SelStart As Long， SelEnd As Long， StSel As Range

    On Error Resume Next

    Application.ScreenUpdating = False

    With Selection

        Set StSel = .Range

        SelStart = .Start

        .MoveDown ‘下移一行

        SelEnd = .Start + 1

        Set MyRange = ActiveDocument.Range(SelStart， SelEnd)

        If MyRange Like "\*" & Chr(13) = True And \_

           MyRange.Find.Execute(FindText：="^m") = True Then \_

 MsgBox "当前行中有手动分页符!"

        StSel.Select

    End With

    Application.ScreenUpdating = True

End Sub

**[013]认识Word的命令栏、控件按钮**

功能简介：通过代码循环，得到所有WORD的命令栏、工具栏中的所有控件、子控件的名称、ID、控件类型等

Dim X As Byte

Sub GetAllCommand()

    Dim i As CommandBar， j As Integer， n As CommandBarControl， PS As String

    Dim A As Integer， B As Integer， C As Integer， P As Paragraph

    On Error Resume Next

    Application.ScreenUpdating = False    ‘关闭屏幕更新

    For Each i In Application.CommandBars    ’在命令栏中循环

        A = A + 1    ‘命令栏计数器

        Select Case i.Position    ‘命令栏的位置

        Case Is = msoBarBottom

            PS = "msoBarBottom"

        Case Is = msoBarFloating

            PS = "msoBarFloating"

        Case msoBarLeft

            PS = "msoBarLeft"

        Case msoBarMenuBar

            PS = "msoBarMenuBar"

        Case Is = msoBarPopup

            PS = "msoBarPopup"

        Case Is = msoBarRight

            PS = "msoBarRight"

        Case msoBarTop

            PS = "msoBarTop"

        End Select

        Selection.InsertAfter A & "命令栏Name： " & i.Name & vbTab & "，Index索引号： " & i.Index & vbTab & "，命令栏位置： " & PS & vbCrLf

        For Each n In i.Controls    ‘在命令栏i中的控件集合中循环

            B = B + 1    ‘一级控件计数器

            X = n.Type

            Selection.InsertAfter vbTab & A & "." & B & "控件(按钮)Name： " & -

n.Caption & vbTab & "，ID/FaceId：" & n.ID & vbTab & "，控件类型：" & TP & vbCrLf

            On Error Resume Next    ‘设置错误陷阱

            For j = 1 To n.Controls.Count    ‘获得该控件下的控件数量

                If Err.Number <> 0 Then Err.Clear： Exit For    ‘如果没有下级控件则退出该循环

                C = C + 1    ‘第二级控件计数器

                X = n.Controls(j).Type

                Selection.InsertAfter vbTab & vbTab & A & "." & B & "." & C & "控件(按钮)Name： " &-

 n.Controls(j).Caption & vbTab & "，ID/FaceId： " & n.Controls(j).ID & vbTab &-

"，控件类型： " & TP & vbCrLf

            Next

            C = 0    ‘复零

        Next

        B = 0    ‘复零

    Next

    For Each P In Me.Paragraphs    ‘在活动文档中的所有段落集合中循环

        With P.Range

            If InStr(P.Range， Chr(9) & Chr(9)) = 1 Then    ‘如果为二级控件(段前有两个TAB键)

                .Font.Color = wdColorRed    ‘红色字体

                .Font.Size = 10    ‘10号大小

                .Font.Name = "Verdana"    ‘字体名称

            ElseIf InStr(P.Range， Chr(9)) = 1 Then    ‘如果为一级控件(段前只有一个TAB键)

                .Font.Color = wdColorBlue    ’兰色字体

                .Font.Size = 11    ‘11号大小

                .Font.Name = "Arial"    ’设置字体

            Else    ‘命令栏段落

                .Font.Color = wdColorBlack    ‘黑色字体

                .Font.Size = 12    ‘12号大小

                .Font.Name = "Tahoma"    ‘设置字体

                .Font.Bold = True    ‘粗体

            End If

        End With

    Next

    Application.ScreenUpdating = True    ’恢复屏幕更新

End Sub

Function TP() As String

    Select Case X   ‘控件类型

    Case Is = 0

        TP = "msoControlCustom"

    Case Is = 1

        TP = "msoControlButton"

    Case Is = 2

        TP = "msoControlEdit"

    Case Is = 3

        TP = "msoControlDropdown"

    Case Is = 4

        TP = "msoControlComboBox"

    Case Is = 5

        TP = "msoControlButtonDropdown"

    Case Is = 6

        TP = "msoControlSplitDropdown"

    Case Is = 7

        TP = "msoControlOCXDropdown"

    Case Is = 8

        TP = "msoControlGenericDropdown"

    Case Is = 9

        TP = "msoControlGraphicDropdown"

    Case Is = 10

        TP = "msoControlPopup"

    Case Is = 11

        TP = "msoControlGraphicPopup"

    Case Is = 12

        TP = "msoControlButtonPopup"

    Case Is = 13

        TP = "msoControlSplitButtonPopup"

    Case Is = 14

        TP = "msoControlSplitButtonMRUPopup"

    Case Is = 15

        TP = "msoControlLabel"

    Case Is = 16

        TP = "msoControlExpandingGrid"

    Case Is = 17

        TP = "msoControlSplitExpandingGrid"

    Case Is = 18

        TP = "msoControlGrid"

    Case Is = 19

        TP = "msoControlGauge"

    Case Is = 20

        TP = "msoControlGraphicCombo"

    Case Is = 21

        TP = "msoControlPane"

    Case Is = 22

        TP = "msoControlActiveX"

    Case Is = 23

        TP = "msoControlSpinner"

    Case Is = 24

        TP = "msoControlLabelEx"

    Case Is = 25

        TP = "msoControlWorkPane"

    Case Is = 26

        TP = "msoControlAutoCompleteCombo"

    End Select

End Function

**[014]认识WORD 中的对话框(Dialog)**

前言：WORD 中的内置对话框，提供了强大的人机对话功能，合理适当地应用对话框，可以极大地方便我们的代码过程和效论.另外，从本代码中，没有使用FOR EACH –NEXT循环，是因为在实际过程中，还可以通过INDEX直接访问该对话框，而不必去记很长的WdWordDialog 常量名(当然会有提示)，另外，WORD VBA帮助文件中提供的WORD 内置对话框的数量也有出入，说明部分对话框不是特别支持访问. Application.Dialogs.Count=227，而实际上利用以下代码可以得到748个对话框。

Sub GetDialogs()

    Dim i As Integer

    On Error Resume Next

    With Application

        .ScreenUpdating = False

        For i = 1 To 10000

            Selection.InsertAfter "对话框" & i & "：" & .Dialogs(i).CommandName & vbCrLf

        Next

        .ScreenUpdating = True

    End With

End Sub

**[015]自定义右键菜单（修改右键）**

 功能简介：在右键文本菜单的中部位置(相当于右击文本时出现的菜单)，添加一个自定义命令，并执行相应过程

Private Sub Document\_Close()

    On Error Resume Next

    Application.CommandBars("Text").Controls("Test").Delete    ’恢复原有菜单

End Sub

Private Sub Document\_Open()

    Dim Half As Byte

    On Error Resume Next

    Dim NewButton As CommandBarButton

    Application.CommandBars("Text").Controls("Test").Delete    ‘预防性删除

    Half = Int(Application.CommandBars("Text").Controls.Count / 2)    ‘中间位置

    Set NewButton = Application.CommandBars("Text").Controls.Add(Type：=msoControlButton， Before：=Half)

    With NewButton

        .Caption = "Test"        ’命令名称

        .FaceId = 100        ’命令的FaceId

        .Visible = True          ’可见

        .OnAction = "MySub" ‘指定响应过程名

    End With

End Sub

Sub MySub()

    MsgBox "It’s A Test For CommandBars(""Text"")!"， vbOKOnly + vbInformation

End Sub

Sub ComReset()    ‘重新设置右键菜单，彻底恢复默认设置

    Application.CommandBars("Text").Reset

End Sub

生成具有Commandbars(“Toolbar list”)或者当于CommandBars("View").Controls("工具栏(&T)")中的命令按钮形式：

Private Sub Document\_Close()

    On Error Resume Next

    Application.CommandBars("Text").Controls("New Menu").Delete    ’恢复原有菜单

End Sub

Private Sub Document\_Open()

    Dim i As Byte， Half As Byte， strName As String， NewButton As CommandBarPopup

    Dim MenuAdd As CommandBarButton

    On Error Resume Next

    Application.CommandBars("Text").Controls("New Menu").Delete    ‘预防性删除

    Half = Int(Application.CommandBars("Text").Controls.Count / 2)    ‘中间位置

    Set NewButton = Application.CommandBars("Text").Controls.Add(Type：=msoControlPopup， Before：=Half)

    With NewButton          ’这是弹出式菜单即右边带有小三角型的

        .Caption = "New Menu"    ’命令名称

        .Visible = True    ‘可见

    End With

    For i = 1 To 4    ’新建四个子命令，批量生成

        strName = "Menu" & i

        Set MenuAdd = NewButton.Controls.Add(Type：=msoControlButton)

        With MenuAdd

            .Caption = strName

            .OnAction = "MySub"

            .State = msoButtonDown    ‘带勾选的命令按钮

            .Visible = True

        End With

    Next

End Sub

Sub MySub()

    Dim ActionTag As String

    ActionCap = CommandBars.ActionControl.Caption

    MsgBox ActionCap

       Select Case ActionTag

              ’以此来区分各个命令并执行指定过程

       End Select

    With Application.CommandBars("Text").Controls("New Menu")

        If .Controls(ActionCap).State = msoButtonDown Then

            MsgBox "It’s A Test!"， vbOKOnly + vbInformation

            .Controls(ActionCap).State = msoButtonUp

        Else

            .Controls(ActionCap).State = msoButtonDown

        End If

    End With

End Sub

Sub ComReset()    ‘重新设置右键菜单，彻底恢复默认设置

    Application.CommandBars("Text").Reset

End Sub

以下为禁用命令和快捷键的常用方式与保存路径，提倡使用修改WORD命令更方便。

Sub Example()

‘将自定义菜单栏\工具栏或者自定义键盘的改变保存于活动文档中

    Application.CustomizationContext = ActiveDocument

    ‘利用CommandBars(Name).Controls(Caption)来定位按钮，具有唯一性

    Application.CommandBars("Standard").Controls("打开(&O)...").Enabled = False    ‘TRUE

    ‘ 利用来定位按钮，不太直观，容易受调整后的命令位置干扰

    Application.CommandBars("Standard").Controls(2).Enabled = True    ‘False

    ‘利用Findcontrol(ID：=)来定位按钮，具有唯一性，并可循环，作用多个此按钮命令

    Application.CommandBars.FindControl(ID：=23).Enabled = True    ‘False

    ‘利用CommandBars(Index).Controls(Index)来定位按钮，直观，但受调整后的命令位置干扰

    Application.CommandBars(1).Controls(2).Enabled = False    ‘True

End Sub

Sub FileOpen()    ‘可以将命令与快捷键一并禁用

    MsgBox "这是修改WORD命令/打开文件"

End Sub

Sub Sample()       ’将 CTRL+O快捷键重新分配或者修改并保存于当前文档中

    CustomizationContext = ActiveDocument

    KeyBindings.Add KeyCode：=BuildKeyCode(wdKeyControl， wdKeyO)， \_

KeyCategory：=wdKeyCategoryMacro， Command：="NoFileOpen"

End Sub

Sub NoFileOpen()

    MsgBox "This is only a test!"

End Sub

**[016]修改WORD命令**

在WORD中，我们可以通过修改WORD命令的方法，来方便地为WORD控件指定用户自定义的过程，完成或者转移(禁用)相应的内置方式.它的原理是利用相应宏名来置换过程的方法.在下面的三个部分中，我们可以体会其中的相同点与不同点。

Sub Example()

    Dim i As CommandBarControl

    For Each i In Application.CommandBars.FindControls

        If i.ID = 4 Then

            i.OnAction = "MySub"       ’指定宏名

        End If

    Next

End Sub

Sub ResetSub()

    Dim i As CommandBarControl

    For Each i In Application.CommandBars.FindControls

        If i.ID = 4 Then ‘ID=4    ‘相当于CTRL+P(文件/打印)

            i.OnAction = ""      ’恢复原有ID功能

        End If

    Next

End Sub

Private Sub Document\_Close()

    ResetSub       ’关闭文档后恢复

End Sub

Private Sub Document\_Open()

    Example ‘修改

End Sub

以上为第一部分，以下为第二部分：

Sub FilePrint()

    MySub

End Sub

Sub MySub()

    MsgBox "不能使用打印功能！"

End Sub

该过程为公用部分

简析：在EXCEL中，我们只能通过FindControls(ID)的方法为原有程序修改命令指定宏过程;在WORD中，我们也可能通过该方法进行;但如果我们采取第二部分的话，更为简单，这就是所谓的修改WORD命令.当然一个名为MySub的过程可以省略，直接写在FilePrint宏中。以下为实用修改WORD命令的一个例子(该例子放在自定义模板中)

Sub FilePrint()’修改WORD命令(文件/打印：CTRL+P)

    Dim Pc As Integer， Var As Integer

    With Application.Dialogs(wdDialogFilePrint)

        If .Show = -1 Then

            Pc = .NumCopies    ‘取得打印份数

            Var = Me.Variables("PrintPageCount").Value    ‘延续以前的打印份数

            Me.Variables("PrintPageCount").Value = Pc + Var    ‘至今共打印的张数

            Me.Save    ‘保存

            MsgBox "目前累计打印份数为" & Me.Variables("PrintPageCount").Value

        End If

    End With

End Sub

Sub FilePrintDefault’修改WORD命令(常用工具栏/打印活动文档)

    ActiveDocument.PrintOut    ‘默认打印

    Me.Variables("PrintPageCount").Value = \_

      Me.Variables("PrintPageCount").Value + 1

    Me.Save    ‘保存

    MsgBox "目前累计打印份数为" & Me.Variables("PrintPageCount").Value

End Sub

Private Sub Document\_Open()

    On Error Resume Next

    Me.Variables.Add Name：="PrintPageCount"    ‘预定一个文档变量

End Sub

如果快速得到WORD中对应命令的命令名称，有多种方法，一是使宏对话框中的WORD命令，我们可以知道所有WORD命令，还可以使用宏对话框中的”ListCommands”命令，将所有WORD命令自动列表;还可以使用自定义/命令/所有命令中获得;也可以通过插入域/MacroButton域中的宏名列表中获得;最方便的是使用CTRL+ALT+数字小键盘上的”+”号，当光标变成”中国结”时，点向所需按钮命令，即出现一个自定义对话框，在这个对话框中所显示的命令，就是你要的命令名称。

**[017]返回所选(当前)段落指定行号的文本内容一**

注意事项：第一个代码可以返回多段落选定区域的行号；第二个代码可以返回所选内容的第一个段落中的指定行号的文本内容。

Dim LineCount As Integer

Sub LinesCount()

    Dim l As String

    On Error Resume Next

    ‘如果光标未选中内容则将第一个光标所在段落选中

    If Selection.Type = wdSelectionIP Then Selection.Paragraphs(1).Range.Select

    Application.ScreenUpdating = False    ’关闭屏幕更新

    CommandBars("Word Count").Visible = True    ‘打开字数统计工具栏

    ‘执行字数统计(重新计数)

    CommandBars("Word Count").Controls(2).Execute

    ’返回第一个列表框中的第六个数据

    l = CommandBars("Word Count").Controls(1).List(6)

    ‘关闭字数统计

    CommandBars("Word Count").Visible = False

    Application.ScreenUpdating = True    ‘恢复屏幕更新

    LineCount = Int(Mid(l， 1， Len(l) - 1))    ‘返回行数值

    ‘返回所选段落(或光标所在段落)的行数

    MsgBox "Selection Paragraphs(1)’s Line Count Is " & LineCount

    ‘返回指定行数的内容

    MsgBox NumlineRange(LineNumber)

End Sub

Function NumlineRange(LineNumber As Variant) As Range

    Dim StRange As Long， EnRange As Long， SelStart As Range

0   LineNumber = InputBox("请输入你要定位的指定段落的行号"， "Microsoft Word")

    If LineNumber = "" Then Exit Function Else LineNumber = LineNumber \* 1

    With Selection

        Set SelStart = ActiveDocument.Range(.Paragraphs(1).Range.Start， .Paragraphs(1).Range.Start)

        Select Case LineNumber    ’行号数据

        Case 0， Is > LineCount    ’大于指定段落的行数

            MsgBox "行号过大或者过小的无效行号错误!"， vbOKOnly + vbInformation

            GoTo 0    ‘重新开始

        Case 1    ‘用户行号为1

            StRange = .Paragraphs(1).Range.Start    ‘所选段落的起点

            ‘所选段落的的第二行的起点(是wdGoToNext)

            EnRange = .GoTo(what：=wdGoToLine， which：=wdGoToNext， Count：=LineNumber).Start

        Case Is = LineCount    ‘用户行号为最后一行

            ‘定位至所选段落的最后一行

            StRange = .GoTo(what：=wdGoToLine， which：=wdGoToNext， Count：=LineNumber - 1).Start

            ‘所选段落的结束位置

            EnRange = .Paragraphs(1).Range.End

        Case Else    ’其它

            ‘返回用户行号的本行(wdGoToNext，所以要-1)的开始位置

            StRange = .GoTo(what：=wdGoToLine， which：=wdGoToNext， Count：=LineNumber - 1).Start

            SelStart.Select    ‘由于光标位置移动后，需要重新选定

            ‘取得下一行的起始位置

            EnRange = .GoTo(what：=wdGoToLine， which：=wdGoToNext， Count：=LineNumber).Start

        End Select

        ‘定义该用户指定行的区域

        Set NumlineRange = ActiveDocument.Range(StRange， EnRange)

    End With

End Function

**[018]返回指定行号文本二**

说明，利用文档属性中的行数进行判断，并根据Document对象的GoTo方法进行定位行号，无需移动光标位置，运行速度更快，版本兼容性更好。

Sub InsertLineRange()

    Dim i As Integer， LineCount As Integer， Doc As Document， StartRange As Long， EndRange As Long

    Dim MyRange As Range， TarLines As Integer

    On Error GoTo Start

Start：     i = InputBox("请输入需要插入的行号")

    If i <= 0 Then MsgBox "无效行号，请重新输入!"， vbOKOnly + vbInformation： GoTo Start

    Application.ScreenUpdating = False

    Set Doc = Documents.Open(FileName：=ThisDocument.Path & "\谢谢七兄.doc"， Visible：=False) ‘以隐藏方式打开指定文档

    ThisDocument.Activate    ‘防止版本不同，加上一句激活本文档

    With Doc

    TarLines = .BuiltInDocumentProperties("Number of lines").Value          ’DOC文档的行数

        If i > TarLines Then

            MsgBox "大于指定文档最大行号数" & TarLines & " ，请重新输入!"， vbOKOnly + vbInformation

            .Close False        ’关闭DOC文档

            GoTo Start           ’返回指定行标签

        Else

            StartRange = .GoTo(wdGoToLine， ， i).Start        ’指定行号的始点位置

            ‘如果输入行号与DOC的总行数一致，则终点位置为文档末位置，反之则为下一行的起点

            EndRange = VBA.IIf(i = TarLines， .Content.End， .GoTo(wdGoToLine， ， i + 1).Start)

            Set MyRange = .Range(StartRange， EndRange)     ’定义一个RANGE对象

‘            MsgBox MyRange

            Selection.InsertAfter MyRange          ’活动文档光标处插入指定行的文本内容

            .Close False   ’关闭文档

        End If

    End With

    Application.ScreenUpdating = True

End Sub

列出活动文档的文件/属性：内置属性列表的代码：

Sub ListProperties()         ’内置属性列表

    Dim rngDoc As Range

    Dim proDoc As DocumentProperty

    Set rngDoc = ActiveDocument.Content

    rngDoc.Collapse Direction：=wdCollapseEnd

    For Each proDoc In ActiveDocument.BuiltInDocumentProperties

        With rngDoc

            .InsertParagraphAfter

            .InsertAfter proDoc.Name & "= "

            On Error Resume Next

            .InsertAfter proDoc.Value

        End With

    Next

End Sub

标题 Title 字数 Number of words

主题 Subject 字符数 Number of characters

作者 Author 安全性 Security

关键词 Keywords 类别 Category

备注 Comments 格式 Format

模板 Template 经理 Manager

上一个作者 Last author 单位 Company

修订次数 Revision number 字节数 Number of bytes

应用程序名 Application name 行数 Number of lines

上次打印时间 Last print date 段落数 Number of paragraphs

创建时间 Creation date 幻灯片数 Number of slides

上次保存时间 Last save time 备注数 Number of notes

编辑时间总计 Total editing time 隐藏幻灯片数 Number of hidden Slides

页数 Number of pages 多媒体剪辑数 Number of multimedia clips

超级链接基础 Hyperlink base 带空格字符数 Number of characters (with spaces)

Number of hidden Slides  Number of multimedia clips

**[019]选定当前页文本**

功能简介：有时需要选定光标所在页的整页文本，此代码将以右键方式作用(右击/选定当前页命令)

Private Sub Document\_Close()

    On Error Resume Next

    Application.CommandBars("Text").Controls("选定当前页").Delete    ’恢复原有菜单

End Sub

Private Sub Document\_Open()

    Dim Half As Byte

    On Error Resume Next

    Dim NewButton As CommandBarButton

    Application.CommandBars("Text").Controls("选定当前页").Delete    ‘预防性删除

    Half = Int(Application.CommandBars("Text").Controls.Count / 2)    ’中间位置

    Set NewButton = Application.CommandBars("Text").Controls.Add(Type：=msoControlButton， Before：=Half)

    With NewButton

        .Caption = "选定当前页"        ’命令名称

        .FaceId = 100           ’命令的FaceId

        .Visible = True        ’可见

        .OnAction = "SelectCurrentPage"     ’指定响应过程名

    End With

End Sub

Sub SelectCurrentPage()

    Dim CurrentPageStart As Long， CurrentPageEnd As Long

    Dim CurrentPage As Integer， Pages As Integer

    On Error Resume Next

    With Selection

        CurrentPage = .Information(wdActiveEndPageNumber)    ’取得当前页页码

        Pages = .Information(wdNumberOfPagesInDocument)    ’取得文档总页数

        ‘返回当前页起点位置

        CurrentPageStart = .GoTo(What：=wdGoToPage， Which：=wdGoToNext， Name：=CurrentPage).Start

        If CurrentPage = Pages Then    ‘两者相等则最后位置为文档最后位置

            CurrentPageEnd = ActiveDocument.Content.End

        Else    ‘否则则为下一页的起点(本页的最后位置)

            CurrentPageEnd = .GoTo(What：=wdGoToPage， Which：=wdGoToNext， Name：=CurrentPage + 1).Start

        End If

        ActiveDocument.Range(CurrentPageStart， CurrentPageEnd).Select

    End With

End Sub

**[020]选定文档任意页(连续)之一**

功能简介：通过对话框，选择或者输入指定的起始页页码和结束页页码，进行选定。

Private Sub Document\_Close()

    On Error Resume Next

    Application.CommandBars("Text").Controls("AnyPagesSelect").Delete    ’恢复原有菜单

End Sub

Private Sub Document\_Open() ‘参见自定义右键菜单

    Dim Half As Byte

    On Error Resume Next

    Dim NewButton As CommandBarButton

    Application.CommandBars("Text").Controls("AnyPagesSelect").Delete    ‘预防性删除

    Half = Int(Application.CommandBars("Text").Controls.Count / 2)    ‘中间位置

    Set NewButton = Application.CommandBars("Text").Controls.Add(Type：=msoControlButton， Before：=Half)

    With NewButton

        .Caption = "AnyPagesSelect"    ‘命令名称

        .FaceId = 100    ‘命令的FaceId

        .Visible = True    ‘可见

        .OnAction = "MySub"    ’指定响应过程名

    End With

End Sub

Sub MySub()

    UserForm1.Show 0

End Sub

Sub ComReset()    ‘重新设置右键菜单，彻底恢复默认设置

    Application.CommandBars("Text").Reset

End Sub

用户窗体

Public Pe As Integer

Private Sub ComboBox1\_Change()

    If ComboBox1.ListCount = 0 Then        ’未选则禁止操作

        CommandButton1.Enabled = False

        Exit Sub

    ElseIf ComboBox1.ListIndex + 1 > Pe Or ComboBox1.ListIndex + 1 < 1 Then

        MsgBox "无效页码！"             ’如果大于总页数则无效

        CommandButton1.Enabled = False

    Else

        CommandButton1.Enabled = True

    End If

End Sub

Private Sub ComboBox2\_Change()

    If ComboBox1.ListCount = 0 Then

        CommandButton1.Enabled = False

        Exit Sub

    ElseIf ComboBox1.ListIndex + 1 > Pe Or ComboBox1.ListIndex + 1 < 1 Then

        MsgBox "无效页码！"

        CommandButton1.Enabled = False

    Else

        CommandButton1.Enabled = True

    End If

End Sub

Private Sub CommandButton1\_Click()

    Dim RangeStart As Long， RangeEnd As Long

    With Selection

        .HomeKey unit：=wdStory      ’将光标移到起始位置

        If ComboBox1.ListIndex + 1 = ComboBox2.ListIndex + 1 Then          ’如果起止页相等

            If ComboBox2.ListIndex + 1 = Pe Then      ’如果为末页则Range的起始点为该页0位置，止点为文档末

                RangeStart = .GoTo(What：=wdGoToPage， Which：=wdGoToNext，-

Name：=Me.ComboBox2.ListIndex + 1).Start

                RangeEnd = ActiveDocument.Content.End

            Else            ’如果不是则起点为指定页首，止点为指定尾页的下一页页首(相当于指定页页尾)

                RangeStart = .GoTo(What：=wdGoToPage， Which：=wdGoToNext，-

Name：=Me.ComboBox1.ListIndex + 1).Start

                RangeEnd = .GoTo(What：=wdGoToPage， Which：=wdGoToNext，-

Name：=Me.ComboBox1.ListIndex + 2).Start

            End If

        ElseIf ComboBox1.ListIndex + 1 = Pe Then        ’如果指定起始页为尾页，则定位到尾页的0位置，止点为文档末

            RangeStart = .GoTo(What：=wdGoToPage， Which：=wdGoToNext，-

Name：=Me.ComboBox2.ListIndex + 1).Start

            RangeEnd = ActiveDocument.Content.End

        ElseIf ComboBox2.ListIndex + 1 = Pe Then          ’如果指定的结束页为尾页，则定位到尾页的0位置，止点为文档末

            RangeStart = .GoTo(What：=wdGoToPage， Which：=wdGoToNext，-

Name：=Me.ComboBox1.ListIndex + 1).Start

            RangeEnd = ActiveDocument.Content.End

        ElseIf ComboBox1.ListIndex + 1 > ComboBox2.ListIndex + 1 Then        ’如果起始页大于结束页，则换个方向选定

            RangeStart = .GoTo(What：=wdGoToPage， Which：=wdGoToNext，-

Name：=Me.ComboBox2.ListIndex + 1).Start

            RangeEnd = .GoTo(What：=wdGoToPage， Which：=wdGoToNext，-

Name：=Me.ComboBox1.ListIndex + 2).Start

        Else                ’其它的则起点为指定起始页的0位置，止点为指定结束页的下一页的0位置

            RangeStart = .GoTo(What：=wdGoToPage， Which：=wdGoToNext，-

Name：=Me.ComboBox1.ListIndex + 1).Start

            RangeEnd = .GoTo(What：=wdGoToPage， Which：=wdGoToNext，-

Name：=Me.ComboBox2.ListIndex + 2).Start

        End If

        ActiveDocument.Range(RangeStart， RangeEnd).Select          ’选定指定区域

    End With

    Unload Me           ’卸载窗体，从内存中释放出来

End Sub

Private Sub UserForm\_Activate()

    Dim i As Integer

    CommandButton1.Enabled = False

    ‘取得文档总页数

    Pe = Selection.Information(wdNumberOfPagesInDocument)

    For i = 1 To  Pe              ‘向组合框添加页码

        Me.ComboBox1.AddItem i

        Me.ComboBox2.AddItem i

    Next

End Sub

**[021]选定文档任意页(连续)之二**

功能简介：之一是利用窗体/selection方法属性，本示例是利用Inputbox/数组/ActiveDocument的属性方法，在不移动插入点的情况下进行的选定，运行速度更快，代码更简洁。

Private Sub Document\_Close()

    On Error Resume Next

    Application.CommandBars("Text").Controls("AnyPagesSelect").Delete    ‘恢复原有菜单

End Sub

Private Sub Document\_Open()    ‘参见自定义右键菜单

    Dim Half As Byte

    On Error Resume Next

    Dim NewButton As CommandBarButton

    Application.CommandBars("Text").Controls("AnyPagesSelect").Delete    ‘预防性删除

    Half = Int(Application.CommandBars("Text").Controls.Count / 2)    ‘中间位置

    Set NewButton = Application.CommandBars("Text").Controls.Add(Type：=msoControlButton， Before：=Half)

    With NewButton

        .Caption = "AnyPagesSelect"    ‘命令名称

        .FaceId = 100    ‘命令的FaceId

        .Visible = True    ‘可见

        .OnAction = "Sample"    ‘指定响应过程名

    End With

End Sub

Sub Sample()

    Dim P As String， PS() As String， PageHome As Integer， PageEnd As Integer， EndPage As Long

    On Error Resume Next

    P = InputBox(prompt：="请在此输入连续页的首页-尾页，以-为分隔符!"， Title：="Word连续页选定")

    If P = "" Then Exit Sub

    PS = Split(P， "-")    ‘返回一个以"-"分隔的一维数组

    If UBound(PS) > 1 Then Exit Sub    ‘如果上标大于1，则退出(用户连续型输入如1-2-7")

    PageHome = PS(0)    ‘首页为数组下标

    PageEnd = PS(1)    ‘尾页为数组上标

    If PageHome > PageEnd Then Exit Sub    ’尾页大于首页则退出

    If PageHome < 1 Then Exit Sub    ’首页小于1则退出

    With ActiveDocument

        ‘EndPage为尾页位置，如果大于文档总页数，则为文档最后位置;反之则下一页的起始位置

        EndPage = VBA.IIf(PageEnd >= .GoTo(wdGoToPage， wdGoToNext， ， PageEnd).Information \_

                          (wdNumberOfPagesInDocument)， .Content.End， \_

                          .GoTo(wdGoToPage， wdGoToNext，PageEnd + 1).Start)

        ’选定指定区域

        .Range(.GoTo(wdGoToPage， wdGoToNext，PageHome).Start， EndPage).Select

    End With

End Sub

**[022]邮件合并中条件格式的设置**

示例代码用途：用邮件合并方式输出成绩单时，将不及格的成绩自动设为红色。

Public WithEvents App As Word.Application    ‘在类模块中声明对应于事件的对象变量。

‘编写指定事件的过程。

Private Sub App\_MailMergeBeforeRecordMerge(ByVal Doc As Document， Cancel As Boolean)

    Dim i As Byte

    ‘如果主文档数据源中的字段i中的数据小于60(分)时

    For i = 5 To 25

        If Doc.MailMerge.DataSource.DataFields(i).Value < 60 Then

            ‘主文档表格的对应单元格中的字体为红色

            Select Case i

            Case 5 To 9

                Doc.Tables(2).Cell(2， i - 3).Range.Font.Color = wdColorRed

            Case 10 To 11

                Doc.Tables(2).Cell(2， i - 1).Range.Font.Color = wdColorRed

            Case 12 To 16

                Doc.Tables(2).Cell(3， i - 10).Range.Font.Color = wdColorRed

            Case 17 To 18

                Doc.Tables(2).Cell(3， i - 8).Range.Font.Color = wdColorRed

            Case 19 To 23

                Doc.Tables(2).Cell(4， i - 17).Range.Font.Color = wdColorRed

            Case 24 To 25

                Doc.Tables(2).Cell(4， i - 15).Range.Font.Color = wdColorRed

            End Select

        Else ‘否则恢复默认字体颜色

            Select Case i

            Case 5 To 9

                Doc.Tables(2).Cell(2， i - 3).Range.Font.Color = wdColorAutomatic

            Case 10 To 11

                Doc.Tables(2).Cell(2， i - 1).Range.Font.Color = wdColorAutomatic

            Case 12 To 16

                Doc.Tables(2).Cell(3， i - 10).Range.Font.Color = wdColorAutomatic

            Case 17 To 18

                Doc.Tables(2).Cell(3， i - 8).Range.Font.Color = wdColorAutomatic

            Case 19 To 23

                Doc.Tables(2).Cell(4， i - 17).Range.Font.Color = wdColorAutomatic

            Case 24 To 25

                Doc.Tables(2).Cell(4， i - 15).Range.Font.Color = wdColorAutomatic

            End Select

        End If

    Next

End Sub

Dim X As New EventClassModule    ’从其他模块中初始化已声明的对象。

Private Sub Document\_Open()

    Set X.App = Word.Application

End Sub

**[023]分页保存-保留格式设置的代码**

功能简介：将主文档的每一页保存为一个文档，并保留中的页面设置，页眉设置，和字体样式等。

Sub SaveAsPage()

    Dim PageCount As Integer， StartRange As Long， EndRange As Long， MyRange As Range

    Dim Fn As String， MyDoc As Document， MyHeader As Range， MyFooter As Range

    On Error Resume Next

    With Selection

        PageCount = .Information(wdNumberOfPagesInDocument)

        .HomeKey unit：=wdStory

        For i = 1 To PageCount

            StartRange = .Start

            Set MyHeader = .Sections(1).Headers(wdHeaderFooterPrimary).Range

            MsgBox MyHeader

            MyHeader.Copy

            Set MyFooter = .Sections(1).Footers(wdHeaderFooterPrimary).Range

            MsgBox MyFooter

            Set MyDoc = Documents.Add

            ’原现有光标所在页的页面设置赋值给新文档

            With Application.Windows(ThisDocument.Name).Selection.Sections(1).PageSetup

                ActiveDocument.Sections(1).PageSetup.TopMargin = .TopMargin

                ActiveDocument.Sections(1).PageSetup.BottomMargin = .BottomMargin

                ActiveDocument.Sections(1).PageSetup.LeftMargin = .LeftMargin

                ActiveDocument.Sections(1).PageSetup.RightMargin = .RightMargin

                ActiveDocument.Sections(1).PageSetup.Orientation = .Orientation

            End With

            With ActiveDocument            ’打开页眉页脚

                .ActiveWindow.View.SeekView = wdSeekCurrentPageHeader

                With Application.Windows(MyDoc).Selection

                    .Paste          ’粘贴其中内容并删除最后一个段落标记

                    .Paragraphs(.Paragraphs.Count).Range.Delete

                End With           ’关闭页眉页脚

                .ActiveWindow.View.SeekView = wdSeekMainDocument

                .ActiveWindow.View.Type = wdPrintView

            End With

            ThisDocument.Activate

            Fn = i & ActiveDocument.Name

            If i = PageCount Then    ‘如果循环到达最后一页

                EndRange = ActiveDocument.Content.End  ‘将文档最后位置赋值于EndRange

            Else

                EndRange = .GoToNext(wdGoToPage).Start    ‘否则，将下一页的起始位置赋值于EndRange(等同于本页的最后位置)

            End If

            Set MyRange = ActiveDocument.Range(StartRange， EndRange)    ‘将本页中的内容进行复制

            MyRange.Copy

            With Application.Windows(MyDoc).Selection

                .Paste

                .Paragraphs(.Paragraphs.Count).Range.Delete

               .Find.Execute findtext：="^m"， Replacewith：=""， Replace：=wdReplaceAll

                MyDoc.SaveAs FileName：=Fn    ’保存文档名

                MyDoc.Close    ’关闭文档

            End With

        Next

    End With

End Sub

**[024]随机文档打开密码的设置**

功能简介：设置随机打开密码，并将其写在文件属性选项卡的自定义的标题中.扩展应用时，可以通过加密该密码，如在基础上加减一个任意数字，还可以通过修改WORD命令的方式，以便每次另存时为调用，当然用模板方式更好，不影响其它文档的正常运行。

提示：WORD中的文档变量随了在文档中保存外，还可以以文档变量(Variables(Item))\自动图文集\注册表和属性对话框等方式进行特定的储贮和读写。

Sub AnySaveAsWritePassword()        ’可更改为SaveAs宏(修改WORD命令)

    Dim Fd As FileDialog， R As Long

    Set Fd = Application.FileDialog(msoFileDialogSaveAs)

    If Fd.Show = -1 Then Fd.Execute Else Exit Sub

    R = Int(1000 \* Rnd()) + 500

    With ActiveDocument

        .BuiltInDocumentProperties(1) = R       ’赋于属性对话框中的标题(1)

        .Password = R       ’打开密码设为该值

        .Save ‘保存

    End With

    Application.ScreenUpdating = False

End Sub

[025]Word中的中文倒字代码

功能简介：批量转换文字方向，使其产生倒字效果(注意不是铅印的反字，效果如：白日依山尽：  白日依山尽)

Option Compare Text          ’不区分大小写

Sub DaoZi()

    Dim i As Range， Ft As String， MyRange As Range

    On Error Resume Next

    Application.ScreenUpdating = False    ‘关闭屏幕刷新

    If Selection.Type = wdSelectionIP Then    ’判断光标位置

        Set MyRange = Me.Content    ‘全文

    Else

        Set MyRange = Selection.Range    ‘所选部分

    End If

    For Each i In MyRange.Characters

        If i Like "[a-z]" = True Or i Like "[0-9]" = True Then

        Else

            Ft = i.Font.Name    ‘原来的字体

            ’中文版式/纵横混排功能

            i.HorizontalInVertical = wdHorizontalInVerticalFitInLine

            i.Font.Name = "@" & Ft    ‘原来字体的@型字体

        End If

    Next

    Application.ScreenUpdating = True    ‘恢复屏幕刷新

End Sub

**[026]返回打印设置，取得所有打印页数（张数）**

修改WORD命令，FILEPRINT，可用于统计打印机或者通过该模板打印了多少纸张

Sub FilePrint()

    Dim MyDialog As Dialog， Ps() As String， Pl() As String， PPcount As Integer， PrintSel As String

    Dim S As Integer， N As Integer， H As Integer， Upper As Integer， Lower As Integer， Cop As Integer

    Set MyDialog = Application.Dialogs(wdDialogFilePrint) ‘定义打印对话框

    With MyDialog

        If .Show = -1 Then         ’按下确定按钮

            Cop = .NumCopies         ’返回打印份数

            Select Case .Range            ’打印区域

            Case 0

                PrintSel = "您选择了打印所有页"     ‘取得文档总页数

                PPcount = ActiveDocument.Content.Information(wdNumberOfPagesInDocument)

            Case 2                 ’相当于打印光标所在页

                PPcount = 1

                PrintSel = "您选择了打印当前第" & Selection.Information(wdActiveEndPageNumber) & "页"

            Case 4 ‘选择从第几页到第几页如"1-3，5，9，10-15"

                PrintSel = "您选择了打印指定页：" & .Pages      ‘数组

                Ps = Split(.Pages， "，")

                Upper = UBound(Ps)     ’上标

                Lower = LBound(Ps)          ’下标

                For i = Lower To Upper

                    N = N + 1

                    ’如果该数组中的某个值中提取有"-"的话

                    If InStr(Ps(i)， "-") > 0 Then

                        Pl = Split(Ps(i)， "-")

                        S = Pl(1) \* 1 - Pl(0) \* 1        ’直接取得上标和下标数值之差

                        H = S + H

                    End If

                Next

                PPcount = N + H      ’打印的页数等于单页和连页数之和

            End Select

            MsgBox PrintSel & "，打印份数为：" & Cop & "，打印的页数为：" & PPcount & "张，" & vbCrLf \_

                 & "实际上产生了" & Cop \* PPcount & "张纸."， vbInformation

        End If

    End With

End Sub

**[027]在文档中插入根号的两个简洁代码**

Sub InsertReq1()    ‘方法一

    Dim Insertvalue As String

    On Error Resume Next

    With Selection

        If .End > .Start Then

            Insertvalue = IIf(Selection Like "\*" & Chr(13)， Range(.Start， .End - 1)， Selection)

            .Delete

            Application.Run "InsertFieldChars"

            .InsertAfter "Eq \r(2，" & Insertvalue & ")"

        End If

        .Fields.ToggleShowCodes

    End With

End Sub

Sub InsertReq2()    ‘方法2

    With Selection

        If .Type = wdSelectionNormal Then \_

 ： .Text = "Eq \r(2，" & IIf(InStr(Selection， Chr(13)) > 0， Range(.Start， .End - 1)， Selection) & ")" \_

           ： Application.Run "InsertFieldChars" \_

                             ： .Fields.ToggleShowCodes

    End With

End Sub

**[028]嵌套域的VBA自动插入代码**

功能简单：用于特定环境下的嵌套域的插入方法的代码

Sub Example()

    Dim Pc As Integer

    Pc = 5               ‘初始化变量，通常可来源于某些文档中的变量

    Application.Run "ViewHeader"    ‘打开页眉页脚

    ’以下为典型的嵌套域的域代码：第{ = { page }+Pc }页共{ = { numpages }+Pc }页

    With Selection

        .WholeStory    ’全选页眉中的文本

        .Text = "共页第页"    ’重新设置新文本，其实是替换了原文本

        .HomeKey    ‘移到文本之首

        .MoveRight Count：=1    ‘移到文本的第一个字符后(光标右移一个字符)

        Application.Run "InsertFieldchars"    ‘插入域标志(或称空域)

        .Text = "= page+ " & Pc    ‘插入域代码，并使其加上一个初始变量

        .Words(2).Select    ‘第二个词选定，其实是将page这个单词选定

        Application.Run "InsertFieldchars"    ‘插入域标志(已是嵌套域)注意此时是第二个域(嵌套域中的域代码为Page了)

        .EndKey    ’将光标移至文本末

        .MoveLeft Count：=1    ’光标左移一个字符

        Application.Run "InsertFieldchars"    ‘插入域标志

        .Text = "= numpages+ " & Pc    ‘插入域代码(相当于共?页，并加上初始变量值)

        .Words(2).Select    ‘第二个词选定，其实是将numpages这个单词选定

        Application.Run "InsertFieldchars"    ‘插入域标志(已是嵌套域)注意此时是第二个域(嵌套域中的域代码为numPages了)

    End With

    Application.Run "ViewHeader"    ‘关闭页眉页脚

End Sub

**[029]数字工具**

功能简介：对选定文档/全部或者表格（两者必居其一）中的数据进行简单计算、人民币符号转换和千分位设置、百分号设置以及科学记数功能

Sub StandardNumber() ‘10，000.00样式

    Dim i As Range， Acell As Cell， CR As Range， YN As String

    On Error Resume Next ‘错误忽略

    Application.ScreenUpdating = False

    If Selection.Type = 2 Then ‘为选定文本

        For Each i In Selection.Words ‘选定词中循环

            If i Like "####\*" = True Then

                If i.Next Like "." = True And i.Next(wdWord， 2) Like "#\*" = True Then

                    i.SetRange Start：=i.Start， End：=i.Next(wdWord， 2).End

                    i = Format(i， "Standard")

                Else

                    i = Format(i， "Standard")

                End If

            End If

        Next i

    ElseIf Selection.Type = 5 Then         ’表格中

        For Each Acell In Selection.Cells

            Set CR = ActiveDocument.Range(Acell.Range.Start， Acell.Range.End - 1)

            If CR Like "####\*" = True Then

                If CR Like "####.#\*" = True Then

                    YN = Format(CR， "Standard")

                    CR.Text = YN

                Else

                    YN = Format(CR， "Standard")

                    CR.Text = YN

                End If

            End If

        Next Acell

    Else

        MsgBox "您只能选定文本或者表格之一!"， vbOK + vbInformation

    End If

    Application.ScreenUpdating = True

End Sub

Sub CurrencyNumber() ‘￥1，000.00样式

    Dim i As Range， Acell As Cell， CR As Range， YN As String

    On Error Resume Next

    Application.ScreenUpdating = False

    If Selection.Type = 2 Then

        For Each i In Selection.Words

            If i Like "####\*" = True Then

                If i.Next Like "." = True And i.Next(wdWord， 2) Like "#\*" = True Then

                    i.SetRange Start：=i.Start， End：=i.Next(wdWord， 2).End

                    i = Format(i， "Currency")

                Else

                    i = Format(i， "Currency")

                End If

            End If

        Next i

    ElseIf Selection.Type = 5 Then

        For Each Acell In Selection.Cells

            Set CR = ActiveDocument.Range(Acell.Range.Start， Acell.Range.End - 1)

            If CR Like "####\*" = True Then

                If CR Like "####.#\*" = True Then

                    YN = Format(CR， "Currency")

                    CR.Text = YN

                Else

                    YN = Format(CR， "Currency")

                    CR.Text = YN

                End If

            End If

        Next Acell

    Else

        MsgBox "您只能选定文本或者表格之一!"， vbOK + vbInformation

    End If

    Application.ScreenUpdating = True

End Sub

Sub ScientificNumber() ‘科学记数法

    Dim i As Range， Acell As Cell， CR As Range， YN As String

    On Error Resume Next

    Application.ScreenUpdating = False

    If Selection.Type = 2 Then

        For Each i In Selection.Words

            If i Like "####\*" = True Then

                If i.Next Like "." = True And i.Next(wdWord， 2) Like "#\*" = True Then

                    i.SetRange Start：=i.Start， End：=i.Next(wdWord， 2).End

                    i = Format(i， "Scientific") & " "

                Else

                    i = Format(i， "Scientific") & " "

                End If

            End If

        Next i

    ElseIf Selection.Type = 5 Then

        For Each Acell In Selection.Cells

            Set CR = ActiveDocument.Range(Acell.Range.Start， Acell.Range.End - 1)

            If CR Like "####\*" = True Then

                If CR Like "####.#\*" = True Then

                    YN = Format(CR， "Scientific") & " "

                    CR.Text = YN

                Else

                    YN = Format(CR， "Scientific") & ""

                    CR.Text = YN

                End If

            End If

        Next Acell

    Else

        MsgBox "您只能选定文本或者表格之一!"， vbOK + vbInformation

    End If

    Application.ScreenUpdating = True

End Sub

Sub CalValue() ‘简单计算

    Dim MyValue As Single

    On Error Resume Next

    If Selection Like "\*" & Chr(13) Then Selection.SetRange Start：=Selection.Start， End：=Selection.End - 1

    MyValue = Selection.Calculate

    Selection.InsertAfter IIf(Abs(MyValue) < 1， "=" & Replace(MyValue， "."， "0.")， "=" & MyValue)

End Sub

Sub InsertPercent() ‘插入百分号

    Dim i As Range， MyRange As Range， ER As Long， CR As Range

    On Error Resume Next

    Application.ScreenUpdating = False

    If Selection.Type = 2 Then

        For Each i In Selection.Words

            mi = Trim(i)

            pi = Trim(i.Previous)

            ni = Trim(i.Next)

            If mi Like "\*#" = True Then

                If ni <> "." Then

                    i.InsertAfter "%"

                ElseIf pi = "." And ni <> "%" Then i.InsertAfter "%"

                End If

            End If

        Next i

    ElseIf Selection.Type = 5 Then

        For Each Acell In Selection.Cells

            Set CR = ActiveDocument.Range(Acell.Range.Start， Acell.Range.End - 1)

            If CR Like "\*#" = True Then

                If CR Like "#\*.\*#" = True Then

                    CR.InsertAfter "%"

                Else

                    CR.InsertAfter "%"

                End If

            End If

        Next Acell

    Else

        MsgBox "您只能选定文本或者表格之一!"， vbOK + vbInformation

    End If

    Application.ScreenUpdating = True

End Sub

**[030]三角函数计算**

功能简介：本代码可以完成对选定文本的三角函数计算(sin，cos，tan，cot和普通数字)的计算

Option Compare Text         ’以文本方式比较字符串

Sub SCTC()

    Dim MyCal As Double， MyValue As Double， CalValue As Double， SelVal As Single， InSin As String

    Const MyPI As Single = 3.14159265358979    ’定义一个常数(PI值)

    With Selection

        If .End = .Start Then MsgBox "请选定需要计算的文本!"： Exit Sub

        If .Text Like "sin(\*)" = True Then             ‘计算Sin值，正弦

            SelVal = CSng(Mid(Selection， 5， Len(Selection) - 5))

            MyValue = SelVal \* MyPI / 180             ‘转换为弧度值

            CalValue = Round(Sin(MyValue)， 13)           ’保留13位小数四舍五入

            If Abs(CalValue) < 1 Then

                .InsertAfter "=0" & CalValue’小于0加上0

            Else

                .InsertAfter "=" & CalValue

            End If

            Exit Sub

        End If

        If .Text Like "cos(\*)" = True Then         ’计算Cos值’余弦

            SelVal = CSng(Mid(Selection， 5， Len(Selection) - 5))

            MyValue = SelVal \* MyPI / 180

            CalValue = Round(Cos(MyValue)， 13)

            If Abs(CalValue) < 1 Then

                .InsertAfter "=0" & CalValue

            Else

                .InsertAfter "=" & CalValue

            End If

            Exit Sub

        End If

        If .Text Like "tan(\*)" = True Then       ’计算Tan值正切

            SelVal = CSng(Mid(Selection， 5， Len(Selection) - 5))

            MyValue = SelVal \* MyPI / 180

            CalValue = Round(Tan(MyValue)， 13)

            If Abs(CalValue) < 1 Then

                .InsertAfter "=0" & CalValue

            Else

                .InsertAfter "=" & CalValue

            End If

            Exit Sub

        End If

        If .Text Like "cot(\*)" = True Then         ‘计算Cot(Ctan)值余切

            SelVal = CSng(Mid(Selection， 5， Len(Selection) - 5))

            MyValue = Round(SelVal \* MyPI / 180， 13)

            MyValue = SelVal \* MyPI / 180

            CalValue = Round(1 / Tan(MyValue)， 13)

            If Abs(CalValue) < 1 Then

                .InsertAfter "=0" & CalValue

            Else

                .InsertAfter "=" & CalValue

            End If

            Exit Sub

        End If

        If .Text Like "[!A-Z]\*" = True Then .InsertAfter "=" & .Calculate： Exit Sub

    End With

End Sub

**[031]汉字拼音解决方案**

功能简介：可以完成对选定或者全部文档中的汉字进行拼音标注，拼音的结果有六种样式和两大类，一类为有声调标记，一类为无声调标记（便于小学生加注）

标准模块

Public NaFont As String， SzFont As Byte， OfValue As Byte， Cc As String， TF As Boolean

Public OB As Byte

Dim CharsEnd As Long， CharsStart As Long， SZ As Single

Sub CallBefore()

    On Error Resume Next

    ‘程序运行条件限制：不可有表格\有域，要有实际内容

    ‘判断依据，如果光标处于未选定状态，为全部文本，否则为选定内容

    If ActiveDocument.Content.Fields.Count > 0 Then  MsgBox "当前文档中包含有域，-

将会影响本程序的正确运行，请删除域" & vbCrLf \_

                  & "然后重新运行本程序!"， vbInformation + vbOKOnly，-

"汉字自动加注拼音"： Exit Sub

    If ActiveDocument.Content.End < 2 Then Exit Sub

    With Selection

        If .Type = wdSelectionIP Then

            If ActiveDocument.Tables.Count > 0 Then  MsgBox "文档中的表格将会影响拼音的生成，-

请重新选定不包含" & vbCrLf  & "表格区域的文本，-

然后重新运行本程序!"， vbInformation + vbOKOnly，-

"汉字自动加注拼音"： Exit Sub

            CharsStart = 0

            CharsEnd = ActiveDocument.Characters.Count

        ElseIf .Type = wdSelectionNormal Then

            If .Tables.Count > 0 Then \_

 MsgBox "选定区域中的表格将会影响拼音的生成，请重新选定不包含" & vbCrLf \_

      & "表格区域的文本，然后重新运行本程序!"， vbInformation + vbOKOnly，-

"汉字自动加注拼音"： Exit Sub

            CharsStart = .Start

            CharsEnd = .End

        Else

            Exit Sub

        End If

        MsgBox "Microsoft Word汉字自动加注拼音文字受文档/选字内容字符数系统配置影响，" & vbCrLf \_

             & "可能会花费一到数分钟的时间来完成每个指定项目，请耐心等待!"，-

vbOKOnly + vbInformation， "汉字自动加注拼音"

        If MsgBox("Micorsoft Word建议您在进行拼音设置之前先对选定部分/全文档进行字体格式设置!" \_

                ， vbYesNo + vbInformation， "汉字自动加注拼音") = vbYes Then Exit Sub

        SZ = Int(.Font.Size)

        UserForm1.Show

        If TF = True Then TF = False： Exit Sub

    End With

End Sub

Sub GetPinYin()         ’返回拼音

    Dim i As Long

    On Error Resume Next

    Application.ScreenUpdating = False           ’关闭屏幕刷新

    With ActiveDocument

        For i = CharsEnd To CharsStart Step -30         ’以30个字为一组进行拼音指南的调用

            If i - 30 <= CharsStart Then      ’不满30个字时说明已到达起点处

                .Range(CharsStart， i).Select          ’以此为区域为拼音指南的文本

            Else

                .Range(i - 30， i).Select    ’否则每30个字为一组进行设置

            End If

            SendKeys "{Enter}"， False ‘预置

            Application.Run "FormatPhoneticGuide"    ’运行拼音指南命令

        Next

        .ActiveWindow.View.ShowFieldCodes = True          ’显示域代码

        ‘以下按用户要求进行调整

        If NaFont <> "宋体" Then  .Content.Find.Execute findtext：="Font：宋体"，-

replacewith：="Font：" & NaFont， Replace：=wdReplaceAll

        If SzFont \* 2 <> SZ Then  .Content.Find.Execute findtext：="hps" & SZ，-

replacewith：="hps" & SzFont \* 2， Replace：=wdReplaceAll

        If Cc <> "居中" Then  .Content.Find.Execute findtext：="jc0"， replacewith：=Cc， Replace：=wdReplaceAll

        If OfValue <> "0" Then \_

 .Content.Find.Execute findtext：="up 9"， replacewith：="up " & OfValue + 9， Replace：=wdReplaceAll

        .ActiveWindow.View.ShowFieldCodes = False         ’显示域结果

        Application.ScreenUpdating = True           ’恢复屏幕更新

        Select Case MsgBox("选择Yes您可以进行去除声调标记的工作，选择NO可以跳过-

此项进行拼音样式设置，" & vbCrLf \_

                         & "选择CANCEL退出程序!"，vbYesNoCancel + vbInformation)

        Case vbYes

            Call DelYinDiao         ’删除音调程序

        Case vbNo

            If MsgBox("您即将进一步拼音进行样式设置，请按Yes，按NO取消!"，-

vbYesNo + vbInformation，"汉字自动加注拼音") = vbYes Then

                UserForm2.Show

            End If

        Case vbCancel

            Exit Sub

        End Select

    End With

    ActiveDocument.UndoClear

End Sub

Sub SetPinYin()    ‘设置拼音样式

    Dim PyRange As Range， StartRange As Long， EndRange As Long， C As Range

    On Error Resume Next

    If OB = 0 Then UserForm2.Show

    Application.ScreenUpdating = False

    With ActiveDocument

        StartRange = .GoTo(what：=wdGoToField， which：=wdGoToFirst).Start

        EndRange = .GoTo(what：=wdGoToField， which：=wdGoToLast).Start + 1

        Set PyRange = .Range(StartRange， EndRange)           ’取得第一个域(拼音)和

        ‘最后一个拼音域的Range

    End With

    PyRange.Select           ’选定该区域

    With Selection

        Select Case OB

        Case 1

            Exit Sub

        Case Else

            .InsertAfter "汉字自动加注拼音"            ’预定义一个位置，以便以后定位

            PyRange.Select

            .Copy

            .Range.PasteSpecial DataType：=wdPasteText        ’无格式文本方式粘贴

            .Find.ClearFormatting

            .Find.Execute findtext：="汉字自动加注拼音"

            .Delete

            EndRange = .Start            ’找到了选择性粘贴后的光标位置(即预定义位置)

            Set PyRange = ActiveDocument.Range(StartRange， EndRange)

            PyRange.Select          ’重新选定进行选择性粘贴后的文本

            Select Case OB

            Case 3

                For Each C In .Characters       ’将原有括号隐藏(无色)

                    If C = "(" Or C = ")" Then C.Font.Color = wdColorWhite

                Next

            Case 4

                For Each C In .Characters             ’将原有括号隐藏并加宽汉字为2倍

                    If C.Next = "(" Then

                        C.Font.Color = wdColorWhite

                        C.Font.Scaling = 200

                    End If

                Next

            Case 5

                For Each C In .Characters            ’将括号隐藏并加宽汉字为2倍

                    If C = "(" Or C = ")" Then C.Font.Color = wdColorWhite

                    If C.Next = "(" Then

                        C.Font.Color = wdColorWhite

                        C.Font.Scaling = 200

                    End If

                Next

            Case 6

                For Each C In .Characters          ’将括号隐藏并加宽汉字为2倍并拉伸为3倍行距

                    If C = "(" Or C = ")" Then C.Font.Color = wdColorWhite

                    If C.Next = "(" Then

                        C.Font.Color = wdColorWhite

                        ’                        C.Font.Scaling = 200

                    End If

                Next

                ‘设置为三位行距

                .Paragraphs.LineSpacingRule = wdLineSpaceMultiple

                .Paragraphs.LineSpacing = LinesToPoints(3)

                ’恢复字体字号

                .Font.Name = NaFont

                .Font.Size = SzFont

                With .Find       ’将文档中的隐藏文字(白色字体)替换为空格

                    .ClearFormatting

                    .Font.Color = wdColorWhite

                    .Text = ""

                    With .Replacement

                        .ClearFormatting

                        .Text = " "

                    End With

                    .Execute Format：=True， Replace：=wdReplaceAll

                End With

            End Select

        End Select

    End With

    ActiveDocument.UndoClear        ’不允许用户撤消操作，以清空部分内存

    OB = 0 ‘恢复OB值，以便下一次调用与判断

    Application.ScreenUpdating = True       ’恢复屏幕更新

End Sub

Sub DelYinDiao()    ‘去除声调标记(适宜于手动加注声调)

    Dim YD As String， i As Byte， SD As String \* 1， RW As String \* 1

    If ActiveDocument.Fields.Count <= 1 Then MsgBox "Microsoft Word未发现可以去除音调的拼音域！"，-

vbOKOnly + vbInformation，"汉字自动加注拼音"： Exit Sub

    Application.ScreenUpdating = False

    ’显示域代码

    ActiveDocument.ActiveWindow.View.ShowFieldCodes = True

    ‘如果域代码中有以下字符，逐个替换

    YD = "āáǎàōóǒòēéěèīíǐìūúǔùǖǘǚǜ"

    For i = 1 To 24

        SD = Mid(YD， i， 1)

        Select Case i

        Case Is <= 4

            RW = "a"

        Case Is <= 8

            RW = "o"

        Case Is <= 12

            RW = "e"

        Case Is <= 16

            RW = "i"

        Case Is <= 20

            RW = "u"

        Case Is <= 24

            RW = "ü"

        End Select

        ActiveDocument.Content.Find.Execute findtext：=SD， replacewith：=RW， Replace：=wdReplaceAll

    Next

    ’如果用户没有MSPY3.0以上版本，则将域代码中的声调标记(1~5)逐个替换

    For i = 1 To 5

        With ActiveDocument.Content.Find

            .Text = i & ")，"

            .Replacement.Text = ")，"

            .Execute Replace：=wdReplaceAll

        End With

    Next

    ‘显示域结果

    ActiveDocument.ActiveWindow.View.ShowFieldCodes = False

    Application.ScreenUpdating = True

    If MsgBox("您即将进一步拼音进行样式设置，请按Yes，按NO取消!"， vbYesNo + vbInformation， \_

              "汉字自动加注拼音") = vbYes Then UserForm2.Show

End Sub

用户窗体

Private Sub CommandButton1\_Click()

‘设置拼音对齐方式

    Select Case Me.ComboBox1.Value

    Case "居中"

        Cc = "jc0"

    Case "0-1-0"

        Cc = "jc1"

    Case "左对齐"

        Cc = "jc3"

    Case "右对齐"

        Cc = "jc4"

    Case "1-2-1"

        Cc = "jc2"

    End Select

    NaFont = Me.ComboBox2.Value       ’返回用户字体设置

    SzFont = Me.ComboBox4.Value       ’返回用户字号设置

    OfValue = Me.ComboBox3.Value          ‘返加用户偏移量设置

    Unload Me         ’从内存中释放

    Call GetPinYin        ’调用GetPinYin过程

End Sub

Private Sub CommandButton2\_Click()

    TF = True

    End

End Sub

Private Sub UserForm\_Initialize()

    Dim Fn， Fs As Byte

    ’初始化对话框，包括标题，字体字号偏移量和对齐

    Me.Caption = "拼音设置-汉字自动加注拼音"

    For Each Fn In Application.FontNames

        Me.ComboBox2.AddItem Fn

    Next

    Me.ComboBox2.Value = "宋体"

    For Fs = 4 To 72

        Me.ComboBox4.AddItem Fs

        Me.ComboBox3.AddItem Fs - 4

    Next

    Me.ComboBox3.Value = 0

    Me.ComboBox4.Value = 10

    With Me.ComboBox1

        .AddItem "居中"

        .AddItem "0-1-0"

        .AddItem "1-2-1"

        .AddItem "左对齐"

        .AddItem "右对齐"

        .Value = "居中"

    End With

    Me.CommandButton1.Default = True

End Sub

用户窗体：’根据用户定义结果进行返回以使程序识别

Private Sub CommandButton1\_Click()

    If OB = 1 Then End

    Unload Me

    Call SetPinYin

End Sub

Private Sub CommandButton2\_Click()

    OB = 1

    End

End Sub

Private Sub OptionButton1\_Click()

    OB = 1

End Sub

Private Sub OptionButton2\_Click()

    OB = 2

End Sub

Private Sub OptionButton3\_Click()

    OB = 3

End Sub

Private Sub OptionButton4\_Click()

    OB = 4

End Sub

Private Sub OptionButton5\_Click()

    OB = 5

End Sub

Private Sub OptionButton6\_Click()

    OB = 6

End Sub

Private Sub UserForm\_Initialize()

‘初始化对话框

    Me.Caption = "拼音样式设置-汉字自动加注拼音"

    Me.OptionButton1.Value = True

    Me.CommandButton1.Default = True

End Sub

**[032]WORD文档的VBProject的引用列表与示例**

功能简介：在对于需要调用工程引用的文档而言，我们在引用之前，必须要取得指定引用的路径，以及了解相应的NAME、GUID、FULLPATHT和属性描述（Description），仅运行GetReferencesName便可在文档中直接列出已选取引用的全路径等四个属性。

Sub GetReferencesName()

    Dim Ref， N As Integer， StrRef As String， MyStr As String

    With Me.PageSetup    ‘进行页面设置

        .Orientation = wdOrientLandscape    ‘横向页面

        .LeftMargin = CentimetersToPoints(1.5)    ‘左边距为1.5CM

        .RightMargin = CentimetersToPoints(1.5)    ’右边距为1.5CM

    End With

    Me.Content.Delete    ’清空全部内容

    With Selection

        .InsertAfter "序"    ‘插入序

        ‘设置一个制表位(0.5)

        .ParagraphFormat.TabStops.Add Position：=CentimetersToPoints(0.5)

        ’发送一个TAB键

        .InsertAfter Chr(9)

        .InsertAfter "引用名称"    ‘插入引用名称

        ‘设置一个制表位(2)

        .ParagraphFormat.TabStops.Add Position：=CentimetersToPoints(2)

        ‘发送一个TAB键

        .InsertAfter Chr(9)

        .InsertAfter "GUID"    ‘插入GUID

        ‘设置一个制表位(9)

        .ParagraphFormat.TabStops.Add Position：=CentimetersToPoints(9)

        ‘发送一个TAB键

        .InsertAfter Chr(9)

        .InsertAfter "引用路径"    ‘插入引用路径

        ‘设置一个制表位(20.5)

        .ParagraphFormat.TabStops.Add Position：=CentimetersToPoints(20.5)

        ‘发送一个TAB键

        .InsertAfter Chr(9)

        .InsertAfter "引用描述"    ‘插入引用描述

        .EndKey Unit：=wdLine    ‘将光标移到最后

        .TypeParagraph    ’发送一个回车

        For Each Ref In Me.VBProject.References    ‘在当前文档的VBProject的引用中循环

            N = N + 1    ‘计数

            With Ref    ‘以下为取得该引用的Name，GUID，Fullpath和Description四个属性

                ’GUID：返回指定 COMAddIn 对象的全局唯一类标识符（GUID）

                ‘Description：返回或设置一个字符串表达式，包含与对象相关联的描述性字符串

                StrRef = N & Chr(9) & .Name & Chr(9) & .GUID & Chr(9) & .Fullpath & Chr(9) & .Description & Chr(13)

                MyStr = MyStr & StrRef

            End With

        Next

        .InsertAfter MyStr    ‘插入所得文本

        .Font.Color = wdColorBlue    ’设置为兰色字体

        .WholeStory    ‘全选

        .Font.Name = "Arial"    ‘设置字体

        .Font.Size = 9    ‘字体为9磅

        .Paragraphs(1).Range.Font.Bold = True    ‘第一个段落为粗体

        .Paragraphs(1).Range.Font.Size = 10    ‘第一个段落为10字体

        Me.Paragraphs(Me.Paragraphs.Count).Range.Delete    ‘最后一个光标所在段落删除

    End With

End Sub

‘以下为示例如何进行打开文档时引用指定引用和退出时删除该引用

Private Sub Document\_Close()

    On Error Resume Next

    With Me.VBProject     ’删除指定引用

        Set Ref = .References("ADODB")

        If Not Ref Is Nothing Then

            .References.Remove Ref

        End If

    End With

    Me.Save

End Sub

Private Sub Document\_Open()

    On Error Resume Next       ’忽略错误主要针对已引用该指定的引用

    ‘通常我们使用AddFromFile REF.FULLPATH为主要手段，因此必须在引用前知道该引用的路径

    ‘本示例是打开时引用ADODB(Microsoft ActiveX Data Objects 2.5 Library)

    Me.VBProject.References.AddFromFile "C：\Program Files\Common Files\system\ado\msado15.dll"

End Sub

**[033]制作动态链接库(\*.dll)文件和WORD中引用动态链接库**

目的：通过动态链接库的制作，可以将WORD中的VBA代码进行封装，达到：一，保护代码的目的，避免他人通过简单方法(解密)就能窥知过程代码，进一步保护源作者的代码;二，将较复杂的代码，通过制作成动态链接库，还可以加快代码的运行速度(特别是各程序间的协同作业);三，简单化调用过程，使用活动文档中的代码数量大大降低，有利于初学者进行使用.以下是在WORD的五个宏(五个过程)，分别是：

StandardNumber功能，对所选数据(文本中或者是表格中的两者之一)，进行自动格式化为千分位数据。

CurrencyNumber功能，对所选数据(文本中或者是表格中的两者之一)，进行自动格式化为人民币的货币格式数据

ScientificNumber功能，对所选数据(文本中或者是表格中的两者之一)，进行自动格式化为科学记数法。

CalValue对所选数据进行简单四则混合运算等功能

InsertPercent对所选数据(文本中或者是表格中的两者之一)，进行自动补加百分号。

请记住以上五个过程名(宏名)

VBE中代码见：[029]数字工具。我们开始制作动态链接库，打开Microsoft Visual Basic 6.0 中文版程序，在"文件"菜单下点"新建工程"，出现是否保存对"工程1"的对话框，选"否"。我们将上述代码原原本本粘贴于VB工程1的代码窗口中，并将其右则的属性对话框中的类的名称更改为"MyDll"，然后，在"工程"菜单下点引用：在出现的引用对话框中选取"Microsoft Word 10.0 Object Library"，这是关键步骤!

然后，我们需要对该工程的属性和文件进行定义，点"工程"菜单下的："工程1属性" ，输入后确认。接下来，是生成DLL文件，点"文件"菜单下的"生成"WdNumberFormated.dll(K)"，出现一个保存对话框，选择并记住这个路径，本例中的DLL文件放在"D：\Test"文件夹下。

我们可以退出VB编辑器，它会询问是否保存下列文件的更改对话框，如果你觉得有必要，可以保存，也可以不保存。

回到WORD中，我们新建一个空白文档，并保存为"DLLTEST.DOC"文件，进入 VBE中，在左侧的资源管理器中，双击PROJECT(DLLTEST)下的"THISDOCUMENT"，出现代码窗口，"，点"工具"菜单下的"引用"命令.引用对话框中自动引用了该动态链接库.确定后退出引用对话框。

现在，我们进入代码调用阶段。在DLLTEST工程的THISDOCUMENT代码窗口中(也可以在该工程的标准模块中进行，即插入/模块)

写下一个(或者多个)过程 ，我们首先声明一个变量(如MyDlls为一个新的动态链接库(WdNumberFormated)的一个类模块(.MyDll)，即此句： Dim MyDlls As New WdNumberFormated.MyDll

全部代码如下：

Sub Dlltest1()

    Dim MyDlls As New WdNumberFormated.MyDll

    MyDlls.CalValue

End Sub

Sub Dlltest2()

    Dim MyDlls As New WdNumberFormated.MyDll

    MyDlls.CurrencyNumber

End Sub

Sub Dlltest3()

    Dim MyDlls As New WdNumberFormated.MyDll

    MyDlls.InsertPercent

End Sub

Sub Dlltest4()

    Dim MyDlls As New WdNumberFormated.MyDll

    MyDlls.ScientificNumber

End Sub

Sub Dlltest5()

    Dim MyDlls As New WdNumberFormated.MyDll

    MyDlls.StandardNumber

End Sub

回到文档中，随便写一个或几个数据，运行其中的一个宏，也可以指定一下，也可以使用ALT+F8，也可以回到VBE中，直接运行相应的宏名(光标定于过程中，按下F5)等等.此处从略。我们已经完成了对动态链接库的制作包装和调用过程.受OFFICE和VB程序版本不同，请读者在实际操作过程中注意相应版本问题。如果我们知道该\*.DLL的确切位置，比如以打包形式的，我们可以免去手动引用，可以以代码形式引用，如：

Private Sub Document\_Open()

    On Error Resume Next

    Me.VBProject.References.AddFromFile "D：\Test\WdNumberFormated.dll"

End Sub

**[034]语音朗读**

适当调用OFFICE各成员间的功能为WORD所用，取长补短.WORD中有语音录入功能，而EXCEL中有语音朗读功能，互为弥补，可至大全.(如要中止，可按下CTRL+BREAK)

Sub SpeakText()

    Dim Sp As Excel.Application

    Set Sp = New Excel.Application

    Sp.Speech.Speak ActiveDocument.Content

    Set Sp = Nothing

End Sub

Private Sub Document\_Open()

    On Error Resume Next

    ‘以下引用EXCEL.EXE

    ActiveDocument.VBProject.References.AddFromFile \_

    "C：\Program Files\Microsoft Office\Office" & Mid(Application.Version， 1， 2) & "\Excel.exe"

    SpeakText

End Sub

**[035]VBE中文代码复制器**

功能简介：由于中文代码字符在VBE中的长度与应用程序中不一致，而导致复制粘贴过程中成为乱码，本程序可安装于WORD/STARTUP启动文件夹下，则其在启动WORD时自动装载，该加载项的自动宏作用，进行事件触发，当卸载时自动禁用。并自动对代码的制作附一简单说明，如版本号，WINDOWS系统版本和作者等代码头，同时为适宜于EXCELHOME论坛中的粘贴，使用了手动换行符替代段落标记。

Public WithEvents MyProject As VBIDE.CommandBarEvents

Private Sub MyProject\_Click(ByVal CommandBarControl As Object， handled As Boolean， CancelDefault As Boolean)

    Dim i As Long， Avs As Object， strSub As String， ComType As String

    Dim RowStart As Long， ColStart As Long， RowEnd As Long， ColEnd As Long

    With Application

        .ScreenUpdating = False

        Set Avs = .VBE.SelectedVBComponent

        .VBE.ActiveCodePane.GetSelection RowStart， ColStart， RowEnd， ColEnd

        If RowStart = RowEnd And ColStart = ColEnd Then

            m = 1： n = Avs.CodeModule.CountOfLines

        ElseIf ColEnd = 1 Then

            m = RowStart： n = RowEnd - 1

        Else

            m = RowStart： n = RowEnd

        End If

        Select Case Avs.Type

        Case 1

            ComType = "标准模块"

        Case 2

            ComType = "类模块"

        Case 3

            ComType = "用户窗体"

        Case 100

            ComType = "ThisDocument"

        Case Else

            ComType = "未知模块"

        End Select

        With Selection

            .Collapse Direction：=wdCollapseEnd

            .InsertAfter "‘\* +++++++++++++++++++++++++++++++++++++++" & vbCrLf \_

                       & "‘\* Created By " & Application.UserName & "@ExcelHome " & Date & " " & Time & vbCrLf \_

                          & "‘仅测试于System： " & System.OperatingSystem & " Word：-

" & Application.Version & " Language： " \_

                                  & Application.Language & vbCrLf \_

                                      & "‘^The Code CopyIn [" & ComType & "-" & Avs.Name & "]^’" & vbCrLf \_

                       & "‘\* --------------------------------------------------------------------------" & vbCrLf

            .Font.Bold = True

            For i = m To n

                strSub = Avs.CodeModule.Lines(i， 1)

                If strSub Like "End Sub\*" = True Or strSub Like "End Type\*" = True Or \_

                   strSub Like "End Function\*" = True Then strSub = strSub & Chr(11) & ""

                .InsertAfter strSub & Chr(11)

            Next

            .Font.Name = "Tahoma"

            .Font.Size = 11

            .Font.Color = wdColorBlue

            ActiveDocument.Range(.Paragraphs(1).Range.Start， .Paragraphs(5).Range.End).Font.Color = wdColorRed

            .Cut

        End With

        .ScreenUpdating = True

    End With

End Sub

The Code CopyIn [ThisDocument-ThisDocument]

Dim MyVbeProject As MyClass

Sub AutoExec()

    On Error Resume Next

    ‘加载时自动加载"VBIDE"库文件和运行AddMybar的过程

    Me.VBProject.References.AddFromFile "C：\Program Files\Common Files\Microsoft Shared\VBA\VBA6\VBE6EXT.OLB"

    ‘引用VBIDE库文件，引用名为"Microsoft Visual Basic for Applications Extensibility 5.3"，版本不同可能略有差异

    AddMyBar

End Sub

Sub AutoExit()

    On Error Resume Next

    ‘卸载时自动去除"VBIDE"引用

    With Me.VBProject

        Set ref = .References("VBIDE")

        If Not ref Is Nothing Then

            .References.Remove ref

        End If

    End With

    DelMyBar

End Sub

Sub AddMyBar()

    Dim MyBar As CommandBarControl

    On Error Resume Next

    DelMyBar    ‘先删除后增加

    Set MyBar = Application.VBE.CommandBars("Code Window").Controls.Add

    With MyBar

        .Caption = "GetCopy"

        .FaceId = 19

        Set MyVbeProject = New MyClass    ‘将MyVbeProject定义为新MyClass类

        Set MyVbeProject.MyProject = Application.VBE.Events.CommandBarEvents(MyBar)

        ‘此句代码意为该MyvbeProject类中的MyProject过程是指向(响应)命令MyBar的单击事件

        ‘MyBar是该事件的源对象，此事件相当于CommandControl的OnAction属性 End With

    End With

End Sub

Sub DelMyBar()

    On Error Resume Next

    ’删除VBE右键中的一个"GetCopy"的命令按钮，使其还原

    Application.VBE.CommandBars("Code Window").Controls("GetCopy").Delete

End Sub

**[036]自动图文集与自选图形-自动插入带编号的小旗**

功能简介：本过程可以自动插入一个带编号的小旗（由直线与“星与旗帜”的组合图形），它需要在指定的模板中进行创建自动图文集，为加快生成速度，我们可以指定快捷键为CTRL+1

Sub AutoInsertShapes()

    Dim i As Integer

    On Error Resume Next    ‘错误继续

    With ActiveDocument

        .Range(0， 0).Select    ‘文档起点选定

        If .Shapes.Count = 0 Then    ‘如果没有自选图形

            i = 1    ‘编号从1开始

        Else

            i = .Shapes.Count    ‘等于自选图形数目，此处假定为1个自选图形

        End If

        ‘插入关联模板中的自动图文集，用户可以对自动图文集修改并保存为原名，注意保存模板名称

        ‘通过修改该并保存自动图文集，可将其格式自动应用

        .AttachedTemplate.AutoTextEntries("小红旗").Insert Where：=Selection.Range， RichText：=True

        ‘修改编号

        .Shapes(.Shapes.Count).GroupItems(1).TextFrame.TextRange.Text = i

        ‘选定该组合图形

        .Shapes(.Shapes.Count).Select

        ‘设置原始位置，可适当更改，此处从略，由用户自行移动至适当位置

        Selection.ShapeRange.Top = 200

        Selection.ShapeRange.Left = 300

    End With

End Sub

**[037]图片编辑器**

对于大量图片的编辑，设置比较统一的图（照）片格式（可任意设置），并配置相关的说明性文字和统一编号（可任意设置），在同一页面中做到类似于即点即输入功能的图片编辑程序。

功能与用途：指定光标处插入指定的图片，并能统一和分批编号。

必须：将工具/宏/安全性级别设置为低，假如不为低，请设置为低后退出并重启WORD。将工具/选项/编辑选项卡中的插入/粘贴图片的方式调整为四周型，否则将不能调整图片大小；注意：每次开启该文档时，先运行“指定高宽”命令，如果没有设置，当点击“插入照片”时，会自动出现设置高宽对话框（相当于调用该“指定高宽”命令）。提示：当每次需要相同尺寸的照片时，无须再进行“指定高宽”和"名称"的设置，程序会自动记忆；只有需要设置不同尺寸时，再行设置高宽或名称。

提示：插入/图片/来自文件命令同新菜单（帮助菜单）右侧新菜单（照片编辑/光标处插入照片）命令和右键菜单/光标处插入照片命令等同，但应先定位，再点击任一命令。

注意：高度和宽度的度量单位为厘米，先高度再宽度，输入对话框中的输入数据形式如：“4\*5”，或者“5.26\*3.17”等，必须用“\*”（星号）作为分隔符，小数点应该使用英文状态下的标点符号，代码程序不支持无效数据的输入。（有提示）。

操作方法：先定位，即在需要插入照片的页面位置，双击鼠标，使光标处于即点即输入位置，然后右击，在右键快捷菜单中出现：“光标处插入照片”，点击该命令，即可在此处进行指定照片的插入。

编号与照片已进行了组合，除非特殊需要，可以取消组合。在组合的情况下，编号栏文本框中可直接进行编辑。

设置高宽和名称：每次需要改变原来的照片尺寸和名称，可通过此命令进行操作.可将照片的编号重新设置为指定的开始编号，注意此数据即使文档退出（保存）后，下次仍然有效。假如用户上次编辑到“照片10”，则重新开启文档后将自动从11开始编号。

错误重启：受不可预知因素影响，可能使插入照片的位置处于非正常状态(位于页面左上角时，您需要使用该命令。

Public WithEvents App As Word.Application ‘定义一个App为WROD程序

Private Sub App\_WindowBeforeRightClick(ByVal Sel As Selection， Cancel As Boolean)

    SLT = Sel.Information(wdHorizontalPositionRelativeToPage)       ’获得光标的LEFT位置

    STP = Sel.Information(wdVerticalPositionRelativeToPage)          ’获得光标的TOP位置

End Sub

Private Sub App\_WindowSelectionChange(ByVal Sel As Selection)

    Dim SelShape As Shape， W As Single， H As Single， Hp As Single， Ht As Single

    On Error Resume Next       ’错误忽略

    If Selection.Type = wdSelectionShape Then    ’如果选定的是图形并且为组合图形其数量为1

        If Sel.ShapeRange.Type = 6 And Sel.ShapeRange.Count = 1 Then

            Set SelShape = Sel.ShapeRange(1)           ’定义对象

            With SelShape            ’取得该对象的宽高和图片总高度

                W = Round(PointsToCentimeters(.Width)， 2)

                H = Round(PointsToCentimeters(.Height)， 2)

                Hp = H - Round(PointsToCentimeters(25)， 2)

                Ht = Round(PointsToCentimeters(25)， 2)

            End With

            Application.StatusBar = "照片宽：" & W & "厘米，" & "高：" & Hp & "厘米;文本框高：0" \_

                                  & Ht & "厘米;图片总高：" & H & "厘米"

        End If

    End If

End Sub

Private Sub Document\_Close()

    On Error Resume Next          ’忽略错误

    ‘删除右键菜单

    Application.CommandBars("Text").Controls("光标处插入照片").Delete

End Sub

Private Sub Document\_Open()

    Dim NewButton As CommandBarButton

    Call ErrReset         ’触发类模块

    On Error Resume Next

    Set NewButton = Application.CommandBars("text").Controls.Add(Type：=msoControlButton)

    With NewButton         ’修改TEXT的右键菜单

        .Caption = "光标处插入照片"

        .OnAction = "InsertPicture"

        .FaceId = 100

        .Visible = True

    End With

End Sub

Sub ResetControls()     ’恢复右键菜单(调试用)

    Application.CommandBars("Text").Reset

End Sub

Public SLT As Single， STP As Single， PH As Single， PW As Single， PicName As String

Sub InsertPicture()

    Dim Mydialog As FileDialog， MyPicture As Shape， MyText As Shape

    Dim PL As Single， PT As Single， Pcount As Integer， strBmp As String

    On Error Resume Next

    Application.ScreenUpdating = False

        If SLT = -1 Or STP = -1 Or Selection.Type <> wdSelectionIP \_

        Then MsgBox "请将光标定位于页面中或者错误的光标选定项目"，-

vbOKOnly + vbCritical， "Microsoft Word"： Exit Sub

     ’MsgBox SLT & STP

    If PH \* PW = 0 Then SetHW

    PicName = ActiveDocument.Variables("PicName").Value

    Set Mydialog = Application.FileDialog(msoFileDialogOpen)

    ’定义一个对话框对象，其文件筛选器格式为以下四种图片格式

    With Mydialog

        .Filters.Clear

        .Filters.Add "Images"， "\*.Bmp; \*.Gif; \*.Jpg; \*.Jpeg"， 1

        .AllowMultiSelect = False          ’只可单选

        If .Show = -1 Then

            strBmp = .SelectedItems(1)      ’取得选中的路径

        Else

            Exit Sub

        End If

        With ActiveDocument

            Pcount = .Variables("Pcount").Value        ’返回文档变量值

            Pcount = Pcount + 1

            .Variables("Pcount").Value = Pcount        ’重新设置文档变量值

            ‘设置图片的属性值

            Set MyPicture = .Shapes.AddPicture(FileName：=strBmp， \_

                                               Left：=SLT， Top：=STP， Width：=PW， Height：=PH)

            With MyPicture

                .Name = "Pone" & Pcount

                .LockAnchor = False

                .WrapFormat.Side = wdWrapBoth

            End With

            Set MyText = .Shapes.AddTextbox(msoTextOrientationHorizontal， SLT， STP + PH， PW， 25)

            With MyText

                .Name = "Ptwo" & Pcount

                .Line.Visible = msoFalse

                .TextFrame.TextRange.Text = PicName & Pcount       ’’对文本框进行编号

                .TextFrame.TextRange.ParagraphFormat.Alignment = wdAlignParagraphCenter

            End With            ‘图形组合并设置为不允许重叠

            .Shapes.Range(Array("Pone" & Pcount， "Ptwo" & Pcount)).Group.Name = "Pthree" & Pcount

            .Shapes("Pthree" & Pcount).WrapFormat.AllowOverlap = False

        End With

    End With

    Application.ScreenUpdating = True

End Sub

Sub SetHW()

    UserForm1.Show ‘运行窗体1

End Sub

Sub SetRestore()    ’编号重置

    Dim Y As String

    Y = InputBox("请在此输入重新开始的编号值"， "Microsoft Word 编号重置")

    If Y = "" Then

        ActiveDocument.Variables("Pcount").Value = 0

    Else

        ActiveDocument.Variables("Pcount").Value = CInt(Y) - 1

    End If

End Sub

Sub test() ‘调试过程预定义文档变量名和变量值

    ActiveDocument.Variables.Add Name：="Pcount"， Value：=0

    ActiveDocument.Variables.Add Name：="PicName"， Value：="照片"

End Sub

Sub GetTest()

    MsgBox ActiveDocument.Variables(1)

    MsgBox ActiveDocument.Variables(2)

End Sub

Sub ErrReset() ‘错误重启(类模块事件当遇到不可预测性错误时将终止)，以此强制运行

     Register\_Event\_Handler

End Sub

标准模块-模块2

Dim X As New EventClassModule      ’定义X为新类EventClassModule

Sub Register\_Event\_Handler()      ’将X类的APP事件指向WORD.APPLICATION

    Set X.App = Word.Application

End Sub

The Code CopyIn [用户窗体-UserForm1]^’

Private Sub CommandButton1\_Click()

    Dim MyValue As String， L As Byte

    On Error GoTo Errhandle      ’错误处理行

    MyValue = Me.TextBox1

    If MyValue = "" Then Exit Sub

    L = InStr(MyValue， "\*")

    If L = 0 Then

        GoTo Errhandle

    Else      ’传递图片高宽

        PH = CentimetersToPoints(CSng(Mid(MyValue， 1， L - 1)))

        PW = CentimetersToPoints(CSng(Mid(MyValue， L + 1， Len(MyValue) - L)))

    End If

    PicName = Me.TextBox2

    If PicName <> "" Then      ’刷新该文档变量值

        ActiveDocument.Variables("PicName").Value = PicName

    Else

        PicName = ActiveDocument.Variables("PicName").Value

    End If

    Me.Hide

    Exit Sub

Errhandle：

    MsgBox "无效数据，请重新正确输入!"， vbOKOnly + vbInformation

    If PH \* PW <> 0 Then

        Me.TextBox1 = PointsToCentimeters(PH) & "\*" & PointsToCentimeters(PW)

    Else

        Me.TextBox1 = "2\*3"

    End If

    Me.TextBox1.SetFocus            ’光标焦点移到TextBox1中

End Sub

Private Sub UserForm\_Initialize()           ’预定义对话框(用户窗体)的属性和值

    Me.Caption = "Microsoft Word 照片尺寸/名称设置"

    If PH \* PW <> 0 Then

        Me.TextBox1 = PointsToCentimeters(PH) & "\*" & PointsToCentimeters(PW)

    Else

        Me.TextBox1 = "2\*3"

    End If

    Me.TextBox2 = ActiveDocument.Variables("PicName")

    Me.TextBox1.SetFocus

    Me.CommandButton1.Default = True

End Sub

**[038]WORD表格中公式代码自动填充**

功能简介：类似于EXCEL中的单元格公式填充(拖曳)

Option Compare Text        ’以文本方式比较

Sub AutoFormula()

    Dim aCell As Cell， Fct As String， Rfct As String， StartRow As Integer， EndRow As Integer

    Dim StartCol As Byte， EndCol As Byte， i As Byte

    On Error Resume Next    ‘错误处理(忽略错误)

    Application.ScreenUpdating = False    ‘关闭屏幕刷新

    With Selection

        If .Information(wdWithInTable) = False Then MsgBox "光标未处于Word表格中!"： GoTo 10

 ’检测选定部分或者单元格是否处于表格中

        StartRow = .Cells(1).RowIndex    ’选定单元格的开始行号

        EndRow = .Cells(.Cells.Count).RowIndex    ’选定单元格的开始列号

        StartCol = .Cells(1).ColumnIndex    ’选定单元格的结束行号

        EndCol = .Cells(.Cells.Count).ColumnIndex    ’选定单元格的结束列号

        Fct = InputBox("请输入选定单元格中首个单元格的公式，以=开头!-

注意引用单元格的行(列)号与公式中的引用相一致!")

        ‘初步判断公式录入是否正确，如果不正确转入行标签为10的语句

        If Fct Like "=[a-z]#\*" = False Or Fct = "" Then MsgBox "无效公式!"： GoTo 10

        If StartCol = EndCol Then    ‘判断是否为同一行中的选定单元格

            For Each aCell In .Cells

                If aCell.RowIndex = StartRow Then

                    aCell.Formula Formula：=Fct    ‘填充第一个公式

                Else

                    Rfct = Replace(Fct， StartRow， aCell.RowIndex)

                    aCell.Formula Formula：=Rfct    ‘根据列号循环填充公式

                End If

            Next

        ElseIf StartRow = EndRow Then    ‘判断是否为同一列中的选定单元格

            .Tables(1).Cell(StartRow， StartCol).Select

            .InsertFormula Formula：=Fct    ‘填充第一个单元格公式

            For i = StartCol + 1 To EndCol

                Rfct = Replace(Fct， Chr(StartCol + 96)， Chr(i + 96))

                .MoveRight unit：=wdCell

                .InsertFormula Formula：=Rfct    ‘循环填充公式(将行号与字母转换)

            Next

        Else

            MsgBox "多行多列的单元格选定区域，Word不予支持!"

        End If

    End With

10：     Exit Sub

    Application.ScreenUpdating = True    ’恢复屏幕刷新

End Sub

http：//club.excelhome.net/dispbbs.asp?BoardID=23&ID=62219

带公式的单元格计算填充：注意：不支持嵌套函数!

适用函数： Sum，Abs，Average，Count，Int，Max，Min，Round

Option Compare Text    ‘以文本方式比较

Sub AutoFormula()

    Dim FFct As String， aCell As Cell， Fun As String， Fct As String， Rfct As String， StartRow As Integer

    Dim EndRow As Integer， StartCol As Byte， EndCol As Byte， i As Byte， Fend As Byte

    On Error Resume Next    ‘错误处理(忽略错误)

    Application.ScreenUpdating = False    ‘关闭屏幕刷新

    With Selection

        If .Information(wdWithInTable) = False Then MsgBox "光标未处于Word表格中!"： GoTo 10

‘检测选定部分或者单元格是否处于表格中

        StartRow = .Cells(1).RowIndex    ‘选定单元格的开始行号

        EndRow = .Cells(.Cells.Count).RowIndex    ‘选定单元格的开始列号

        StartCol = .Cells(1).ColumnIndex    ’选定单元格的结束行号

        EndCol = .Cells(.Cells.Count).ColumnIndex    ‘选定单元格的结束列号

        FFct = InputBox("请输入选定单元格中首个单元格的公式，以=开头!-

注意引用单元格的行(列)号与公式中的引用相一致!" & Chr(13) & "函数内部必须带有小括号()")

        ‘初步判断公式录入是否正确，如果不正确转入行标签为10的语句

        If FFct = "" Then Exit Sub    ’如果用户按下取消则退出运行

        Fend = InStr(FFct， "(")    ‘得到"("的位置

        If Fend = 0 Then MsgBox "无论什么算式，必须有配对包括!"， vbOKOnly + vbInformation： Exit Sub

        Fun = Mid(FFct， 2， Fend - 2)    ‘取得函数

        If Fun <> "" Then    ’如果非空

            ‘检查函数是否正确

            If InStr("Sum，Abs，Average，Count，Int，Max，Min，Round"， Fun & "，") = 0 Then \_

 MsgBox "对不起，本程序不支持该公式!本程序支持的公式为：" & Chr(13) & -

"Sum，Abs，Average，Count，Int，Max，Min，Round"， vbOKOnly \_

                  + vbInformation： Exit Sub

        End If

        Fct = Mid(FFct， Fend， Len(FFct) - Fend + 1)    ‘提取需要填充的单元格数据

           MsgBox Fun

           MsgBox Fct

        If Fct Like "([a-z]#\*)" = False Or Fct = "" Then MsgBox "无效运算式!"， vbOKOnly + vbInformation： GoTo 10

        If StartCol = EndCol Then    ‘判断是否为同一行中的选定单元格

            For Each aCell In .Cells

                If aCell.RowIndex = StartRow Then

                    aCell.Formula Formula：="=" & Fun & Fct   ‘填充第一个公式

                Else

                    Rfct = Replace(Fct， StartRow， aCell.RowIndex)

                    aCell.Formula Formula：="=" & Fun & Rfct    ’根据列号循环填充公式

                End If

            Next

        ElseIf StartRow = EndRow Then    ’判断是否为同一列中的选定单元格

            .Tables(1).Cell(StartRow， StartCol).Select

            .InsertFormula Formula：="=" & Fun & Fct    ‘填充第一个单元格公式

            For i = StartCol + 1 To EndCol

                Rfct = Replace(Fct， Chr(StartCol + 96)， Chr(i + 96))

                .MoveRight unit：=wdCell

                .InsertFormula Formula：="=" & Fun & Rfct    ‘循环填充公式(将行号与字母转换)

            Next

        Else

            MsgBox "多行多列的单元格选定区域，Word不予支持!"

        End If

    End With

10：     Exit Sub

    Application.ScreenUpdating = True    ’恢复屏幕刷新

End Sub

**[039]取得汉字笔画数(WORD版)**

注意此版必须结合EXCEL的汉字笔画数据库("HzBhJsBiao.xls")，否则不能单独运行。

功能简介：一次性取得任意选定数量的简体汉字（GB2312字符集6763个）的笔画数。

Sub GetBhshu()

    Dim xlObj As Excel.Application， Wk As Excel.Workbook， C As Excel.Range

    Dim i As Range， Chars As String

    On Error Resume Next

    Application.ScreenUpdating = False

    ’检测任务栏中是否有EXCEL程序

    If Tasks.Exists("Microsoft Excel") = True Then

        ’引用原有EXCEL程序

        Set xlObj = GetObject(， "Excel.Application")

        ’创建EXCEL程序

    Else

        Set xlObj = CreateObject("Excel.Application")

    End If

    ‘打开同一路径下的EXCEL笔画数据库

    Set Wk = xlObj.Workbooks.Open(ThisDocument.Path & "\HzBhJsBiao.xls")

    Set C = Wk.Sheets(1).Range("A1：B6764")

    ’在当前文档中的字符集中循环

    For Each i In Selection.Characters

        Cr = xlObj.WorksheetFunction.VLookup(i， C， 2， False)

        ‘设置错误陷阱

        ‘如果错误则原有字符不变

        If Err.Number <> 0 Then

            Err.Clear

            Cr = i

        Else    ‘反之则用字符和该字符的笔画来取代，注意保存在内存中

            Cr = i & "(" & Cr & ")"

        End If

        Chars = Chars & Cr    ’累加器

    Next

    Selection.Text = Chars    ‘将原有文本替换为新带笔画数的文本

    Wk.Close False    ‘退出笔画数据库工作薄

    Application.ScreenUpdating = True

End Sub

Private Sub Document\_Open()

    On Error Resume Next

    ‘引用EXCEL.APPLICATION

    ActiveDocument.VBProject.References.AddFromFile "C：\Program Files\Microsoft Office\Office" & Mid(Application.Version， 1， 2) & "\Excel.exe"

End Sub

**[040]后台解除已知密码的VBA工程的代码**

功能简介：用于后台解除VBA工程代码并完成相应修改的代码，亦可用于EXCEL中

Sub UnProtectPassWord()

    Dim MyPw As String

    ‘最新修改时间：2004-12-11 16：15：04

    MyPw = "123"         ‘假设密码为123，可修改

    Application.ScreenUpdating = False

‘打开VBE/工具/Project属性对话框

    Application.VBE.CommandBars.FindControl(ID：=2578).Execute

‘发送密码和回车，第二次回车为确定属性对话框框

    SendKeys MyPw & "{Enter 2}"， True

    Call ReWork

    Application.ScreenUpdating = True

End Sub

Sub ReWork()

‘测试用于修改VBA代码的代码，注意宏安全性的可靠性来源中勾选信’任对于VB项目的访问

Me.VBProject.VBComponents(1).CodeModule.ReplaceLine 3， "‘最新修改时间：" & Now         ‘将当前时间写在代码中

End Sub

**[041]画直角坐标系**

功能简介：本程序可以实现WORD中绘制直角坐标系，原点以页面左上角为绝对位置，根据用户需要进行定位，并可实现无刻度、二分度、和八分度

Public BeforeShapes As Integer

Sub 画坐标系()

    UserForm1.Show

End Sub

Sub SelAllShapes()

    Dim AllShapes()， ShapeCount As Integer， N As Shape， Y As Integer

    ShapeCount = ActiveDocument.Shapes.Count

    Y = 0

    ’定义一维上标可变数组，从0开始

    ReDim AllShapes(ShapeCount - BeforeShapes - 1)

    With ActiveDocument

        For Each N In .Shapes

            If N.Name Like "已有图形\*" = False Then

                AllShapes(Y) = N.Name

                Y = Y + 1

            End If

        Next N

        With .Shapes.Range(AllShapes).Group

            .ZOrder msoSendToBack

            .Select

            .Name = "坐标系"

        End With

    End With

End Sub

用户窗体-UserForm1

Private Sub CommandButton1\_Click()

    Dim XLeft As Single， XTop As Single， YLeft As Single， YTop As Single， XLong As Single

    Dim YTtop As Single， YHight As Single， XLine As Shape， YLine As Shape， i As Single

    Dim M As Byte， MyTextbox As Shape， MyValue As Single， ModValue As Byte

    On Error Resume Next    ‘忽略错误

    ’必要数据判断

    If Me.TextBox1 = "" Or Int(Me.TextBox1) <> Me.TextBox1 \* 1 Then MsgBox "无效数据!"， \_

                vbInformation： Exit Sub

    If Me.TextBox2 = "" Or Int(Me.TextBox2) <> Me.TextBox2 \* 1 Then MsgBox "无效数据!"， \_

                vbInformation： Exit Sub

    If Me.TextBox3 = "" Or Int(Me.TextBox3) <> Me.TextBox3 \* 1 Then MsgBox "无效数据!"， \_

                vbInformation： Exit Sub

    If Me.TextBox3 \* 1 > Me.TextBox1 \* 1 Or Me.TextBox3 \* 1 > Me.TextBox2 \* 1 Then \_

 MsgBox "无效数据!"， vbInformation：Exit Sub

    ’TextBox1为原点横坐标 ， TextBox2为原点纵坐标

    Application.ScreenUpdating = False

    XLeft = CentimetersToPoints(Me.TextBox1 - Me.TextBox3 / 2)

    XLong = CentimetersToPoints(Me.TextBox3 + 0.5)

    XTop = CentimetersToPoints(Me.TextBox2)

    YLeft = CentimetersToPoints(Me.TextBox1)    ‘左边距

    ’顶部距离为原点纵坐标+高度/2，从下至上.则上部顶点为原点纵坐标-TextBox3/2-0.5

    YTop = CentimetersToPoints(Me.TextBox2 + Me.TextBox3 / 2)

    YTtop = CentimetersToPoints(Me.TextBox2 - Me.TextBox3 / 2 - 0.5)

    YHight = CentimetersToPoints(Me.TextBox3)

    With ActiveDocument

        BeforeShapes = .Shapes.Count    ’获取工作之前的图形总数

        If BeforeShapes >= 1 Then

            For i = 1 To BeforeShapes

                .Shapes(i).Name = "已有图形" & BeforeShapes & i    ’避免重复命名值出错

            Next

        End If

        ‘    If BeforeShapes >= 1 Then MsgBox "非完全版，请删除其它图形或者在另一文档中重新建立坐标系！" \_

             ： Exit Sub

        Set XLine = .Shapes.AddLine(XLeft， XTop， XLeft + XLong， XTop)

        Set MyTextbox = .Shapes.AddTextbox(msoTextOrientationHorizontal， XLeft + XLong - 5， XTop + 5， 20， 15)

        With MyTextbox    ‘设置X轴文本框

            .Line.Visible = msoFalse

            .TextFrame.MarginBottom = 0

            .TextFrame.MarginLeft = 0

            .TextFrame.MarginRight = 0

            .TextFrame.MarginTop = 0

            .TextFrame.TextRange.Font.Name = "Arial"

            .TextFrame.TextRange.Font.Size = 10

            .TextFrame.TextRange = "X"

        End With

        With XLine    ’设置箭头形状

            .Line.EndArrowheadStyle = msoArrowheadTriangle

            .Line.EndArrowheadLength = msoArrowheadLengthMedium

            .Line.EndArrowheadWidth = msoArrowheadWidthMedium

        End With

        Set YLine = .Shapes.AddLine(YLeft， YTop， YLeft， YTtop)

        Set MyTextbox = .Shapes.AddTextbox(msoTextOrientationHorizontal， YLeft - 20， YTtop， 15， 15)

        With MyTextbox    ‘设置Y轴文本框

            .Line.Visible = msoFalse

            .TextFrame.MarginBottom = 0

            .TextFrame.MarginLeft = 0

            .TextFrame.MarginRight = 0

            .TextFrame.MarginTop = 0

            .TextFrame.TextRange.Font.Name = "Arial"

            .TextFrame.TextRange.Font.Size = 10

            .TextFrame.TextRange = "Y"

        End With

        With YLine    ‘设置箭头形状

            .Line.EndArrowheadStyle = msoArrowheadTriangle

            .Line.EndArrowheadLength = msoArrowheadLengthMedium

            .Line.EndArrowheadWidth = msoArrowheadWidthMedium

        End With

        Set MyTextbox = .Shapes.AddTextbox(msoTextOrientationHorizontal， \_

                                           CentimetersToPoints(Me.TextBox1) - 10， CentimetersToPoints(Me.TextBox2) - 1， 15， 15)

        With MyTextbox    ‘设置原点O文本框

            .Line.Visible = msoFalse

            .TextFrame.MarginBottom = 0

            .TextFrame.MarginLeft = 0

            .TextFrame.MarginRight = 0

            .TextFrame.MarginTop = 0

            .TextFrame.TextRange.Font.Name = "Arial"

            .TextFrame.TextRange.Font.Size = 8

            .TextFrame.TextRange = "O"

            .ZOrder msoSendToBack

        End With

        If Me.OptionButton1.Value = True Then Call SelAllShapes： Exit Sub   ‘未选刻度值退出

        If Me.OptionButton2.Value = True Then MyValue = 0.5： ModValue = 2

        If Me.OptionButton3.Value = True Then MyValue = 0.125： ModValue = 5

        For i = 0 To Me.TextBox3 \* 1 Step MyValue

            M = VBA.IIf(VBA.IIf(MyValue = 0.5， i \* 10 Mod 10 = 0， i \* 10 Mod 5 = 0)， 10， 5)

            .Shapes.AddLine CentimetersToPoints(i + Me.TextBox1 - Me.TextBox3 / 2)， XTop - M， \_

                            CentimetersToPoints(i + Me.TextBox1 - Me.TextBox3 / 2)， XTop

            .Shapes.AddLine YLeft， CentimetersToPoints(Me.TextBox2 + Me.TextBox3 / 2 - i)， \_

                            YLeft + M， CentimetersToPoints(Me.TextBox2 + Me.TextBox3 / 2 - i)

            If M = 10 And i - Me.TextBox3 / 2 <> 0 Then    ‘逢0.5和1标识数值，忽略0值(与零点合)

                ‘对X轴刻度

                Set MyTextbox = .Shapes.AddTextbox(msoTextOrientationHorizontal， \_

                       CentimetersToPoints(i + Me.TextBox1 - Me.TextBox3 / 2) - 3， XTop + 3， 10， 10)

                With MyTextbox    ‘设置刻度文本框及值

                    .Line.Visible = msoFalse

                    .TextFrame.MarginBottom = 0

                    .TextFrame.MarginLeft = 0

                    .TextFrame.MarginRight = 0

                    .TextFrame.MarginTop = 0

                    .TextFrame.TextRange.Font.Name = "Arial"

                    .TextFrame.TextRange.Font.Size = 5

                    .TextFrame.TextRange = i - Me.TextBox3 / 2

                    .ZOrder msoSendToBack

                End With

                ‘对Y轴刻度

                Set MyTextbox = .Shapes.AddTextbox(msoTextOrientationHorizontal， YLeft + 12， \_

                     CentimetersToPoints(Me.TextBox2 + Me.TextBox3 / 2 - i) - 8， 10， 10)

                With MyTextbox    ’设置刻度文本框及值

                    .Line.Visible = msoFalse

                    .TextFrame.MarginBottom = 0

                    .TextFrame.MarginLeft = 0

                    .TextFrame.MarginRight = 0

                    .TextFrame.MarginTop = 0

                    .TextFrame.TextRange.Font.Name = "Arial"

                    .TextFrame.TextRange.Font.Size = 5

                    .TextFrame.TextRange = i - Me.TextBox3 / 2

                    .ZOrder msoSendToBack

                End With

            End If

        Next

        Call SelAllShapes    ’全选图形宏(SelAllShapes)

    End With

    Application.ScreenUpdating = True

End Sub

Private Sub CommandButton2\_Click()

    Me.TextBox1 = ""

    Me.TextBox2 = ""

    Me.TextBox3 = ""

End Sub

Private Sub CommandButton3\_Click()

End

End Sub

Private Sub UserForm\_Activate()

    Me.TextBox3.SetFocus

    Me.CommandButton1.Default = True

End Sub

http：//club.excelhome.net/dispbbs.asp?boardID=23&ID=73111&page=2(第六楼)

**[042]Word绘图中的交点自动绘制**

功能简介：自动绘制两个线条的交点，只适用于水平线条与垂直线条的相交

Sub SetIntersect()

    Dim Xo As Single， X1 As Single， X As Single， Lw As Single， ShType As MsoAutoShapeType， InStr1 As String

    Dim Yo As Single， Y1 As Single， Y As Single， Interor As Shape， FoColor As WdColor， InStr2 As String

    Dim H As Single， W As Single， M As Single， N As Single

    On Error Resume Next

    With ActiveDocument.PageSetup

        Xo = .LeftMargin    ‘文档页面左边距

        Yo = .TopMargin    ‘文档页面上边距

    End With

    With Selection

        If .ShapeRange.Count = 2 Then    ’判断选定图形的数量

            ’判断是否为水平线条或者是垂直线条

            If .ShapeRange(1).Height <> 0 And .ShapeRange(1).Width <> 0 Or .ShapeRange(2).Height-

 <> 0 And .ShapeRange(2).Width <> 0 Then MsgBox "非水平/垂直线条，无法定位!"，-

vbOKOnly + vbInformation： Exit Sub

            Lw=IIf(.ShapeRange(1).Line.Weight .ShapeRange(2).Line.Weight，-

.ShapeRange(1).Line.Weight， .ShapeRange(2).Line.Weight)

            If .ShapeRange(1).Height = 0 And .ShapeRange(2).Width = 0 Then

                ‘如果线条一的高度为0(水平线条)并且线条二的宽度为0(垂直线条)

                Y1 = .ShapeRange(1).Top

                W = .ShapeRange(1).Width

                X1 = .ShapeRange(2).Left

                H = .ShapeRange(2).Height

            ElseIf .ShapeRange(1).Width = 0 And .ShapeRange(2).Height = 0 Then

                ‘如果线条一为垂直线条线条二为水平线条

                X1 = .ShapeRange(1).Left

                H = .ShapeRange(1).Height

                Y1 = .ShapeRange(2).Top

                W = .ShapeRange(2).Width

            Else

                MsgBox "平行线条，无法找到交点!"， vbOKOnly + vbInformation

                Exit Sub

            End If

            ‘尽管符合一个为水平线条一个为垂直线条，但如果组合的图形的高宽与它们不等

            ‘则说明不在没有相交点

            .ShapeRange.Group.Select

            M = .ShapeRange.Width

            N = .ShapeRange.Height

            If M <> W Or N <> H Then MsgBox "Word未能找到相交点!"， vbOKOnly + vbInformation： .ShapeRange.Ungroup.Select： Exit Sub

            .ShapeRange.Ungroup.Select

        Else

            MsgBox "图形选定数目限于二个，无法继续!"， vbOKOnly + vbInformation

            Exit Sub

        End If

        ‘进行交点图形设置

        InStr1 = InputBox("请设置交点图形，1(默认)为圆形，2 为正方形，3为菱形")

        Select Case InStr1

        Case 2    ‘正方形

            ShType = msoShapeRectangle

        Case 3    ‘菱形

            ShType = msoShapeDiamond

        Case Else    ‘其它则为圆形交点

            ShType = msoShapeOval

        End Select

        ‘设置交点的填充色

        InStr2 = InputBox("请设置交点的填充色，1(默认)为黑色，2为红色，3为绿色，4为黄色，5为蓝色")

        Select Case InStr2

        Case 2    ‘红色

            FoColor = wdColorRed

        Case 3    ‘绿色

            FoColor = wdColorGreen

        Case 4    ‘黄色

            FoColor = wdColorYellow

        Case 5    ‘蓝色

            FoColor = wdColorBlue

        Case Else    ‘黑色(默认)

            FoColor = wdColorBlack

        End Select

        ‘设置交点的图标位置

        X = Xo + X1 - (Lw + 3.25) / 2

        Y = Yo + Y1 - (Lw + 3.25) / 2

        ‘插入交点

        Set Interor = ActiveDocument.Shapes.AddShape(ShType， X， Y， Lw + 3.25， Lw + 3.25)

        With Interor    ‘设置交点的属性

            .Line.ForeColor = FoColor

            .Fill.ForeColor = FoColor

            .Line.Weight = Lw

            .ZOrder msoBringToFront

            .Select

        End With

    End With

End Sub

**[044]Word 中的AutoCad功能**

功能简介：可能设置原点坐标，并随时更改，通过定位两个点的相对位置，自动连成线条，并自动标注尺寸。

Public WithEvents App As Word.Application

Private Sub App\_WindowSelectionChange(ByVal Sel As Selection)

    On Error Resume Next

    With Selection

        If .ShapeRange.Count = 1 And .ShapeRange.AutoShapeType = msoShapeOval Then

            UserForm1.CommandButton4.Enabled = True

            If .ShapeRange.Name = "原点" Then

                UserForm1.CommandButton4.Enabled = False

                Application.StatusBar = "X轴相对坐标：X=0" & ";Y轴相对坐标：Y=0" \_

                    & ";X轴绝对坐标：X=" & Format(PointsToMillimeters(X0 + X1)，-

"0.0") & ";Y轴绝对坐标：Y=" & Format(PointsToMillimeters(Y0 + Y1)， "0.0") \_

                            & ";Left：=" & .ShapeRange.Left & ";Top：=" & .ShapeRange.Top

            Else

                Application.StatusBar = "X轴相对坐标：X=" & Format(PointsToMillimeters(.ShapeRange.Left - X1)，-

"0.0") & ";Y轴相对坐标：Y=" &

                         Format(PointsToMillimeters(.ShapeRange.Top - Y1)， "0.0") &-

";X轴绝对坐标：X=" & Format(PointsToMillimeters(.ShapeRange.Left + X0)， "0.0") \_

                               & ";Y轴绝对坐标：Y=" & Format(PointsToMillimeters(.ShapeRange.Top + Y0)， "0.0") \_

                                 & ";Left：=" & .ShapeRange.Left & ";Top：=" & .ShapeRange.Top

            End If

        Else

            UserForm1.CommandButton4.Enabled = False

        End If

    End With

End Sub

标准模块-模块1

Public X1 As Single， Y1 As Single， X0 As Single， Y0 As Single， Sp0 As Shape

Dim X As New EventClassModule

Sub Register\_Event\_Handler()

    Set X.App = Word.Application

End Sub

Sub ConLine()

    Dim L1 As Single， T1 As Single， L2 As Single， T2 As Single， Sp1 As Shape， Sp2 As Shape

    Dim LineLenth As Single， MyLength As Shape， ConLin As Shape， CenterT As Single， CenterL As Single

    On Error GoTo ErrorHandle

    With Selection

        If .ShapeRange.Count = 2 Then

            Set Sp1 = .ShapeRange(1)

            Set Sp2 = .ShapeRange(2)

            L1 = Sp1.Left + X0 + 0.5

            L2 = Sp2.Left + X0 + 0.5

            T1 = Sp1.Top + Y0 + 0.5

            T2 = Sp2.Top + Y0 + 0.5

            Set ConLin = ActiveDocument.Shapes.AddConnector(msoConnectorStraight， L1， T1， L2 - L1， T2 - T1)

            If Sp1.Left = Sp2.Left Then

                LineLenth = Format(PointsToMillimeters(Abs(Sp1.Top - Sp2.Top))， "0.0")

                CenterT = Abs((T2 - T1) / 2) + IIf(T2 > T1， T1， T2)

                Set MyLength = ActiveDocument.Shapes.AddShape(msoShapeRectangle， L1 - 12， CenterT， 20， 10)

            ElseIf Sp1.Top = Sp2.Top Then

                LineLenth = Format(PointsToMillimeters(Abs(Sp1.Left - Sp2.Left))， "0.0")

                CenterL = Abs((L1 - L2) / 2) + IIf(L2 > L1， L1， L2)

                Set MyLength = ActiveDocument.Shapes.AddShape(msoShapeRectangle， CenterL， T1 - 12， 20， 10)

            Else

                LineLenth = Format(PointsToMillimeters(Sqr((Sp1.Left - Sp2.Left) ^ 2 + (Sp1.Top - Sp2.Top) ^ 2))， "0.0")

                CenterT = Abs((T2 - T1) / 2) + IIf(T2 > T1， T1， T2)

                CenterL = Abs((L1 - L2) / 2) + IIf(L2 > L1， L1， L2)

                Set MyLength = ActiveDocument.Shapes.AddShape(msoShapeRectangle， CenterL， CenterT， 20， 10)

            End If

        End If

    End With

    ‘    MsgBox LineLenth

    With MyLength

        .Select

        .Line.Visible = msoFalse

        .ZOrder msoBringToFront

        With Selection

            .ShapeRange.TextFrame.TextRange.Select

            .TypeText Text：=LineLenth

            .ShapeRange.TextFrame.MarginBottom = 0

            .ShapeRange.TextFrame.MarginLeft = 0

            .ShapeRange.TextFrame.MarginRight = 0

            .ShapeRange.TextFrame.MarginTop = 0

            .WholeStory

            .Font.Name = "Arial"

            .Font.Size = 6

        End With

    End With

    ConLin.ZOrder msoBringToFront

    Sp1.ZOrder msoBringToFront

    Sp2.ZOrder msoBringToFront

    ConLin.Select

    Exit Sub

ErrorHandle：

    MsgBox "Word 无法作出判断，可能您选定的点不满二个或者超过了二个!"， vbOKOnly + vbInformation

End Sub

Sub ShowMe()

    UserForm1.Show (0)

End Sub

Sub td()

    MsgBox Selection.ShapeRange.Top

End Sub

Private Sub Document\_Open()

    Register\_Event\_Handler

    With ActiveDocument.PageSetup

        X0 = .LeftMargin

        Y0 = .TopMargin

    End With

    On Error Resume Next

    ActiveDocument.Shapes("原点").Select

    X1 = Selection.ShapeRange.Left

    Y1 = Selection.ShapeRange.Top

End Sub

用户窗体-UserForm1

Private Sub CommandButton1\_Click()

    Dim Sp1 As Shape， X2 As Single， Y2 As Single

    On Error Resume Next

    With ActiveDocument.PageSetup

        X0 = .LeftMargin

        Y0 = .TopMargin

    End With

    If Me.Caption = "设置原点坐标" Then

        Err.Number = 0

        ActiveDocument.Shapes("原点").Select

        If Err.Number <> 0 Then

            Err.Clear

            If Me.TextBox1 <> "" And Me.TextBox2 <> "" Then

                X1 = MillimetersToPoints(Me.TextBox1)

                Y1 = MillimetersToPoints(Me.TextBox2)

            End If

        End If

        Set Sp0 = ActiveDocument.Shapes.AddShape(msoShapeOval， X0 + X1， Y0 + Y1， 1， 1)

        Sp0.Select

        Sp0.Name = "原点"

        Sp0.ZOrder msoBringToFront

        Me.Hide

        Exit Sub

    ElseIf Me.Caption = "相对于原点坐标" Then

        If Me.TextBox1 <> "" And Me.TextBox2 <> "" Then

            X2 = MillimetersToPoints(Me.TextBox1)

            Y2 = MillimetersToPoints(Me.TextBox2)

            Set Sp1 = ActiveDocument.Shapes.AddShape(msoShapeOval， X0 + X1 + X2， Y0 + Y1 + Y2， 1， 1)

            Sp1.Select

        End If

        Me.TextBox1 = ""

        Me.TextBox2 = ""

        Me.TextBox1.SetFocus

    End If

End Sub

Private Sub CommandButton2\_Click()

    Me.Hide

End Sub

Private Sub CommandButton3\_Click()

    On Error Resume Next

    Me.Caption = "设置原点坐标"

    ActiveDocument.Shapes("原点").Delete

    Me.TextBox1 = ""

    Me.TextBox2 = ""

    Me.TextBox1.SetFocus

End Sub

Private Sub CommandButton4\_Click()

    Dim Sp1 As Shape， Sp2 As Shape

    On Error Resume Next

    Set Sp2 = Selection.ShapeRange(1)

    ActiveDocument.Shapes("原点").Delete

    Set Sp1 = ActiveDocument.Shapes.AddShape(msoShapeOval， X0 + X1 + X2， Y0 + Y1 + Y2， 1， 1)

    Sp1.ZOrder msoBringToFront

    With Sp2

        X1 = .Left

        Y1 = .Top

        .Name = "原点"

        .ZOrder msoBringToFront

    End With

    Me.Caption = "相对于原点坐标"

    Me.CommandButton4.Enabled = False

End Sub

Private Sub TextBox1\_Change()

    Me.CommandButton1.Default = True

End Sub

Private Sub UserForm\_Activate()

    On Error Resume Next

    Err.Number = 0

    ActiveDocument.Shapes("原点").Select

    If Err.Number <> 0 Then

        Err.Clear

        Me.Caption = "设置原点坐标"

    Else

        Me.Caption = "相对于原点坐标"

    End If

    Me.TextBox1 = ""

    Me.TextBox2 = ""

    Me.TextBox1.SetFocus

End Sub

**[044]乾坤大挪移**

对正常方向字体进行挪移，并可设置框线类型及文本从右到左或者从左到右，从上到下或者从下到上，对竖排字体（适用一种并受WORD表格限制，仅在字数300~500字左右进行装裱可达到类似书法贴或古籍效果，可进一步完善），横排字数不限。

Public Sz As Byte， Bor As Byte， Rl As Byte， Ud As Byte

Sub SetUnderline()

    Dim i As Integer， FilName As String， FilPath As String， LisValue As String， LineOf As Integer， Orient As Byte

    Dim NewDoc As Document， NewTable As Table， n As Integer， X As Long， Y As Long， MyText As String

    On Error GoTo ErrorHandle

    Application.ScreenUpdating = False

    With ActiveDocument

        .Content.Font.Size = Sz \* 1.1

        FilPath = .Path

        FilName = .Name

        Orient = .Content.Orientation

        CommandBars("Word Count").Visible = True

        CommandBars("Word Count").Controls(2).Execute

        LisValue = CommandBars("Word Count").Controls(1).List(6)

        CommandBars("Word Count").Visible = False

        LineOf = Int(Mid(LisValue， 1， Len(LisValue) - 1))

    End With

    Set NewDoc = Documents.Add

    With NewDoc

        .SaveAs FileName：=FilPath & "\U" & FilName

        Set NewTable = .Tables.Add(Range：=Selection.Range， NumRows：=IIf(Orient = 0， LineOf， 1)， NumColumns：=IIf(Orient = 0， 1， LineOf))

    End With

    Documents(FilName).Activate

    With ActiveDocument

        .Range(0， 0).Select

        For n = 1 To LineOf

            Selection.EndKey unit：=wdLine

            Selection.HomeKey unit：=wdLine， Extend：=wdExtend

            MyText = IIf(Rl = 0， Selection， StrReverse(Selection))

            NewTable.Cell(IIf(Orient = 0， IIf(Ud = 0， n， LineOf - n + 1)， 1)， IIf(Orient = 0， 1， IIf(Ud = 0， n， LineOf - n + 1))).Range.Text = MyText

            Selection.MoveDown unit：=wdLine， Count：=1

        Next

    End With

    With NewDoc

        .Activate

        .Tables(1).Select

        .PageSetup.Orientation = IIf(Orient = 1， wdOrientLandscape， wdOrientPortrait)

        With Options

            .DefaultBorderLineStyle = wdLineStyleSingle

            .DefaultBorderLineWidth = wdLineWidth050pt

            .DefaultBorderColor = wdColorRed

        End With

        Select Case Bor

        Case 0

            Application.Run "BorderBottom"

            Application.Run "BorderHoriz"

        Case 1

            Application.Run "BorderAll"

        End Select

        .Content.Font.Size = Sz

    End With

    Documents(FilName).Content.Font.Size = Sz

    Application.ScreenUpdating = True

    Exit Sub

ErrorHandle：

    MsgBox "Word遇到不可预测性错误，本程序将不能正确执行，请检查后再运行！"

    Exit Sub

End Sub

Sub ShowMe()

    UserForm1.Show

End Sub

用户窗体-UserForm1

Private Sub CommandButton1\_Click()

    On Error Resume Next

    Me.Hide

    Sz = Me.ComboBox1.ListIndex + 5

    Bor = Me.ComboBox2.ListIndex

    Rl = Me.ComboBox3.ListIndex

    Ud = Me.ComboBox4.ListIndex

    If Me.ComboBox2.Value = "More" Then MsgBox \_

    "Word注意到：您选取的框线为More，更多框线设置请在完成本功能后在目标文件的格式/边框和底纹中进行!"

    Call SetUnderline

End Sub

Private Sub UserForm\_Activate()

    On Error Resume Next

    Me.ComboBox1.ListIndex = 7

    Me.ComboBox2.ListIndex = 0

    Me.ComboBox3.ListIndex = 0

    Me.ComboBox4.ListIndex = 0

    Me.CommandButton1.Default = True

End Sub

Private Sub UserForm\_Initialize()

    Dim i As Byte

    On Error Resume Next

    With Me.ComboBox1

        For i = 5 To 30

            .AddItem i

        Next

    End With

    With Me.ComboBox2

        .AddItem "下框线"

        .AddItem "全框线"

        .AddItem "More"

    End With

    With Me.ComboBox3

        .AddItem "从左至右"

        .AddItem "从右向左"

    End With

    With Me.ComboBox4

        .AddItem "从上至下"

        .AddItem "从下向上"

    End With

End Sub

Private Sub UserForm\_QueryClose(Cancel As Integer， CloseMode As Integer)

    On Error Resume Next

    Cancel = True’关闭无效

End Sub

**[045]遍历文件夹之一**

Sub Example1()    ‘本示例通过DIR来遍历指定文件夹中的所有

‘WORD文档并且加上指定密码后保存并退出

    Dim Adoc As String， PsDoc As Document

    On Error Resume Next

    ChDrive "C"    ‘设置当前驱动器盘符

    ChDir "C：\Documents and Settings\My Documents\Temp"    ‘进入指定目录

    Adoc = Dir("\*.doc")

    Application.ScreenUpdating = False

    Do While Adoc <> ""    ‘如果是文件夹，或者没有此文件，则会返回""

        ‘        MsgBox Adoc

        Set PsDoc = Documents.Open(Adoc)    ‘打开指定文档

        PsDoc.Protect Type：=wdAllowOnlyFormFields， Password：="Password"

        PsDoc.Close True

        Adoc = Dir()

    Loop

    Application.ScreenUpdating = True

End Sub

**[046]遍历文件夹之二**

Sub Example2()’此代码功能为列出指定文件夹中所有选取的WORD文件全路径名

    Dim MyDialog As FileDialog， GetStr As String

    On Error Resume Next ‘忽略错误

    ‘定义一个文件夹选取对话框

    Set MyDialog = Application.FileDialog(msoFileDialogFilePicker)

    With MyDialog

        .Filters.Clear    ‘清除所有文件筛选器中的项目

        .Filters.Add "所有 WORD 文件"， "\*.doc"， 1    ‘增加筛选器的项目为所有WORD文件

        .AllowMultiSelect = True    ‘允许多项选择

        If .Show = -1 Then    ‘确定

            For Each vrtSelectedItem In .SelectedItems    ‘在所有选取项目中循环

                GetStr = GetStr & vbCrLf & vrtSelectedItem

            Next vrtSelectedItem

            ‘            MsgBox GetStr

            Selection.InsertAfter GetStr    ‘列出所有文件名

        End If

    End With

End Sub

**[047]遍历文件夹之三**

Sub Example3()    ‘本代码用于指定文件夹中随机返回一个文件名

    Dim MyFile As Object， MyFolder As Object， MyFolders As Object

    Dim i As Integer， RndNumber As Integer， n As Integer

    Set MyFolders = CreateObject("Scripting.FileSystemObject")    ‘创建系统文件

    ’获得指定系统文件下的文件夹对象

    Set MyFolder = MyFolders.GetFolder("C：\My Documents\AppliOffice\AppliWord")

    ‘获得指定文件夹下的文件总数

    i = MyFolder.Files.Count

    VBA.Randomize    ’初始化随机数生成器

    RndNumber = Int(i \* Rnd + 1)    ’取得一个从1到文件总数间的一个随机数

    ‘    MsgBox RndNumber

    For Each MyFile In MyFolder.Files

        n = n + 1    ‘设置循环条件计数

        ’如果满足要求则返回文件名并退出程序

        If n = RndNumber Then MsgBox MyFile.Path： Exit Sub

    Next

End Sub

**[048]遍历文件夹之四**

通过遍历所有驱动器和所有该驱动器下的所有文件夹中查找符合搜索条件的文件，注意，如果我们在找到该文件路径时，加上KILL或者Delete方法，则会删除所有符合搜索条件的文件。

Sub ListDrivesFiles()

    Dim Fs As Object， Dr As Object， Dc As Object， DrName As String， i As Long， N As Long

    Dim Mysearch As Office.FileSearch， StrFile As String， AllFile As String

    On Error Resume Next

    Set Mysearch = Application.FileSearch    ‘定义一个Application.FileSearch

    Set Fs = CreateObject("Scripting.FileSystemObject")    ‘定义一个系统文件夹对象

    Set Dc = Fs.Drives    ‘定义一个系统文件夹下的驱动器集合

    For Each Dr In Dc    ‘在驱动器集合下循环，遍历驱动器

        DrName = Dr.driveletter & "：\"    ’获得驱动器名

        With Mysearch

            .NewSearch    ‘设置一个新搜索

            .LookIn = DrName    ’在该驱动器盘符下

            .SearchSubFolders = True    ‘搜索子文件夹

            ‘    .FileType = msoFileTypeWordDocuments           ‘以此可以定义文件类型

            .FileName = "\*.DOT"    ‘搜索一个指定文件，此处为任意WORD模板文件

            If .Execute() > 0 Then    ’开始并搜索成功

                For i = 1 To .FoundFiles.Count

                    N = N + 1    ’计数

                    StrFile = N & vbTab & .FoundFiles(i) & vbCrLf

                    AllFile = AllFile & StrFile    ‘内存中累计

                Next i

            End If

        End With

    Next

    Selection.InsertAfter AllFile    ’在文档中插入

End Sub

**[049]批量重命名文件**

Sub FileNewName()

    Dim FSO As Object， FDR As Object， F As Object， i As Variant， OldName As String， NewName As String

    On Error Resume Next            ’忽略错误

    Set FSO = CreateObject("Scripting.FileSystemObject")        ’创建计算机文件系统以向其访问

    Set FDR = FSO.GetFolder("D：\Test")             ’指定其中访问的文件夹对象

    Set F = FDR.Files            ’定义该文件夹中的所有文件集合

    For Each i In F              ’在指定文件下的文件中循环

        OldName = FDR & "\" & i.Name

        NewName = FDR & "\" & Mid(i.Name， 3)    ‘去掉前两个字符

        Name OldName As NewName

    Next i

End Sub

**[050]拖曳ActiveX控件**

说明：通常ActiveX控件在设计时间定位，以下代码解决了在运行时间控件根据鼠标方向移动到指定位置，当然您需要插入一个ActiveX控件-Image。

Private Sub Image1\_MouseUp(ByVal Button As Integer， ByVal Shift As Integer， ByVal X As Single， ByVal Y As Single)

    Me.Image1.Left = X + Me.Image1.Left           ’释放鼠标时鼠标相对于Image的左上角的LEFT位置

    Me.Image1.Top = Y + Me.Image1.Top            ’释放鼠标时鼠标相对于Image的左上角的TOP位置

End Sub

**[051]打字游戏**

功能简介： 这是一款结合WORD实际与用户窗体制作的打字练习程序，注意请在同一目录下安装有"练习文章.doc"

标准模块-模块1

Public StTime As Date， EnTime As Date， Rn As Integer， Sd As Integer， CountText As Integer， YText As Range

Public TF As Boolean， Ys As Integer， Temp As Byte， ST As Single， Nst As Single

Sub MoveFont()

    Dim Er As Integer

    On Error Resume Next

    If TF = False Then Exit Sub

    CountText = Application.Selection.End

    Call UserForm\_Layout

    If CountText > 0 Then

        EnTime = Now

        Er = Rn + CountText - 1

        If Er = YText.End - 1 Then MsgBox "对不起，已到文章末，请重新再来!"： UserForm1.CommandButton1.Value = True： Exit Sub

        DoEvents

        UserForm1.TextBox1 = Mid(YText， Er， 25)

        UserForm1.TextBox3.Value = Format(CDate(EnTime - StTime)， "H：MM：ss")

    End If

    WaitTime

End Sub

Sub WaitTime()

    If TF = False Then Exit Sub

    Application.OnTime When：=Now + TimeValue("00：00：01")， Name：="MoveFont"

    Nst = Selection.Information(wdVerticalPositionRelativeToPage)

End Sub

Sub Starting()

    MySub

End Sub

Sub MySub()

    On Error Resume Next

    UserForm1.Show (0)

    Call UserForm\_Layout

    UserForm1.CommandButton1.Value = True

End Sub

Private Sub UserForm\_Layout()

   With UserForm1 ‘定位窗体位置

        .Left = (Application.Width - .Width) / 2       ’水平居中

       .Top = 0 + (Nst - ST)        ’随光标移动而移动

    End With

End Sub

Private Sub Document\_Close()

    On Error Resume Next

    Application.CommandBars("text").Reset

    Options.SaveInterval = Temp         ’还原原有定时保存调协

    TF = False

    ThisDocument.Close False           ’关闭并不保存

End Sub

Private Sub Document\_Open()

    On Error Resume Next

    Temp = Options.SaveInterval        ’返回定时保存时间间隔

    If Temp <> 0 Then Options.SaveInterval = 0     ’不允许定时保存(考虑可能会对计时器产生影响)

    ‘以下为修改右键菜单

    Set MyControl = Application.CommandBars("text").Controls.Add

    With MyControl

        .FaceId = 102

        .Caption = "BeginOrEnd"

        .Visible = True

        .OnAction = "MySub"

    End With

    For i = 1 To Application.CommandBars("text").Controls.Count - 1

        Application.CommandBars("text").Controls(i).Visible = False

    Next

    MySub

End Sub

Sub ComReset()    ’恢复默认值

    Application.CommandBars("text").Reset

End Sub

用户窗体-UserForm1

Private Sub CommandButton1\_Click()

    Dim MyRange As Range， YwRange As String， i As Range， n As Range， Ect As Integer

    Dim Xct As Integer， Zz As String

    On Error GoTo ErrHandle    ‘进入错误处理程序

    If Me.CommandButton1.Caption = "Start" Then    ’如果按钮名为"Start"

        Application.ScreenUpdating = False    ‘关闭屏幕更新

        TF = True    ‘将TF赋值为True

        ‘打开同目录下的该文档

        Documents.Open FileName：=ThisDocument.Path & "\练习文章.doc"

        ‘该文档窗口隐藏

        Application.Windows("练习文章.doc").Visible = False

        ‘开始时间

        StTime = Now

        ‘将命令1的名称修改为"End"

        Me.CommandButton1.Caption = "End"

        ‘产生一个从1-3000之间的随机数

        Rn = Int(Rnd() \* 3000 + 1)

        ’固定该随机数

        Ys = Rn

        ‘取得"练习文章"文档的所有文本内容

        Set YText = Documents("练习文章.doc").Content

        ’将光标移入TextBox1中

        Me.TextBox1.SetFocus

        ‘该文本框中放入从Ys开始的30个字

        Me.TextBox1 = Mid(YText， Ys， 30)

        ‘提示一下第一个字符(可能不完全显示出来)

        MsgBox "您准备好了吗?第一个文字为（" & Mid(YText， Ys， 1) & "）开始!"

        ‘激活当前文档

        ThisDocument.Activate

        ‘活动文档中的所有内容删除

        ActiveDocument.Content.Delete

        ‘取得0位置(range(0，0))下的光标的TOP位置

        ST = Selection.Information(wdVerticalPositionRelativeToPage)

        ’起点位置

        CountText = Selection.Start

        ‘以下三句清空三个文本框中的数据

        Me.TextBox2 = ""

        Me.TextBox3 = ""

        Me.TextBox4 = ""

        ‘启动定时器

        WaitTime

    Else

        ’将命令1的名称设置为"Start"，相当于按下了"End"，结束游戏

        Me.CommandButton1.Caption = "Start"

        ‘取得结束的当前时间值

        EnTime = Now

        ’将TF赋值为False

        TF = False

        ‘返回结束时的光标所在位置起点

        Sd = Application.Selection.Start

        ‘定义一个用户输入区域

        Set MyRange = ActiveDocument.Range(0， Sd)

        YwRange = Documents("练习文章.doc").Range(Ys - 1， Ys + Sd)

        For Each i In MyRange.Characters    ‘两个区域逐字比较

            Xct = Xct + 1

            Zz = Mid(YwRange， Xct， 1)

            If i <> Zz Then Ect = Ect + 1

        Next

        Me.TextBox2.Value = Format(1 - Ect / Sd， "0.00%")    ’返回正确率

        ‘返回计时

        Me.TextBox3.Value = Format(CDate(EnTime - StTime)， "H：MM：ss")

        ‘返回速度值

        Me.TextBox4 = Round(Sd / ((EnTime - StTime) \* 24 \* 60)) & "(录入" & Sd & "字)"

        ‘初始化公用变量值

        StTime = 0

        EnTime = 0

        Sd = 0

        Rn = Empty

        Set YText = Nothing

        Application.ScreenUpdating = True    ‘恢复屏幕更新

        Exit Sub

ErrHandle：            ‘错误处理程序

        MsgBox "该程序出现不可预测错误，将被关闭，请在退出后重新打开该程序！"

        Application.Quit False    ‘强制退出

    End If

End Sub

Private Sub UserForm\_Initialize()    ‘初始化用户窗体

    Me.Caption = "打字练习"

    Me.CommandButton1.Caption = "Start"

    Me.CommandButton1.SetFocus

End Sub

Private Sub UserForm\_QueryClose(Cancel As Integer， CloseMode As Integer)

    On Error Resume Next    ’允许用户使用窗体关闭按钮

    TF = False

    Documents("练习文章.doc").Close False    ‘退出并不保存该文档

End Sub

**[052]关于禁用宏则不能正确打开的代码之一**

功能简介：利用加解ASCII码技术进行伪装，如果禁用宏，则看到的是经过加密的字符;只有运行宏，才能看到真实内容。

Public TF As Integer

Sub DocClose() ‘修改WORD命令，防止以低宏方式打开，以SHIFT方式(禁宏)退出

    Document\_Close

    WordBasic.DocClose

End Sub

Sub EditHz() ‘编辑汉字

    Dim i As Range， Char As Integer， Chars As String

    Application.ScreenUpdating = False       ’禁用屏幕更新

    For Each i In Me.Characters         ’在字符集合中循环

    ’根据TF值进行加密或者解密(基本只针对汉字，对于段落标记等不予处理)

        Char = IIf(Asc(i) > 0 And Asc(i) < 33， Asc(i)， Asc(i) + TF)

        Chars = Chars & Chr(Char)

    Next

    With Selection

        .WholeStory

        .Text = Chars

        .Paragraphs(.Paragraphs.Count).Range.Delete

        .MoveStart unit：=wdStory

    End With

    Application.ScreenUpdating = True

End Sub

Private Sub Document\_Close()

‘此处需要结合一个页眉中单元格的一个提示(设置了一个书签)

    If Len(Me.Bookmarks("Blaster").Range.Text) > 2 Then

        Me.Saved = True

    Else

        Me.Bookmarks("Blaster").Range.Text = 1

        TF = 1       ’加密

        EditHz          ’运行指定过程

        Me.UndoClear    ’防止撤消

        Me.Save         ’保存文档

    End If

End Sub

Private Sub Document\_Open()

    TF = -1       ’解密

    Me.Bookmarks("Blaster").Range.Text = ""           ’删除该书签单元格中的内容

    EditHz        ’运行指定过程

    Me.UndoClear          ’防止撤消

    Me.Save      ’保存

End Sub

**[053]关于禁用宏则不能正确打开的代码之二**

功能简介：利用文档变量存贮文档内容，关闭时将文档内容"剪切于"文档变量中;打开时从中加载.重要提示：宜以模板形式进行，并且自定义-删除该文档模板中的插入/书签命令.模板的作用主要是使用VBS脚本编辑器不可用。

Dim MyString As String

Private Sub Document\_Close()

    On Error Resume Next

    ’如果指定书签(预定义)值为2，则直接退出(相当于按下了SHIFT)

    If Me.Bookmarks("Blaster").Range.Text Like "2\*" = True Then Exit Sub

    Application.ScreenUpdating = False

    With Me

        MyString = .Content         ’将所有内容存于文档变量中

        .Variables(1).Value = MyString

        .Content.Delete        ’所有文本内容全部删除

        .Bookmarks("Blaster").Range.Text = 2

        .Save

        .Close          ’保存并退出

    End With

    Application.ScreenUpdating = True

End Sub

Private Sub Document\_Open()

    On Error Resume Next

    Application.ScreenUpdating = False    ‘如果运行了此宏，则加载文档变量(1)

    With Me

        .Bookmarks("Blaster").Range.Text = 1

        .Content = Me.Variables(1).Value        ’删除最后一个段落(原文档中光标所在段落)

       .Paragraphs(.Paragraphs.Count).Range.Delete

       .UndoClear          ‘清空撤消内容

    End With

    Application.ScreenUpdating = True

End Sub

Sub IsEmpties()       ’预定义文档变量名

    MyString = "人性的测试 "

    Me.Variables(1) = MyString

End Sub

**[054]关于禁用宏则不能正确打开的代码之三**

功能简介：以模板形式保存文档，并利用该模板中的自动图文集进行文档的带格式存储.利用模板形式一是自动图文集的保存;二是规避VBS脚本编辑器.要点：在自定义中删除插入/书签命令和插入/自动图文集命令.此法优于第二法(文档变量存贮法)

Private Sub Document\_Close()

    On Error Resume Next

    Application.ScreenUpdating = False

    If Len(Me.Bookmarks("Blaster").Range.Text) > 2 Then Me.Close False： Exit Sub

    With Me

        .Bookmarks("Blaster").Range.Text = 1

        .AttachedTemplate.AutoTextEntries.Add Name：="Dear Sir："， Range：=Me.Content

        .Content.Delete

        .Save

        .Close

    End With

        Application.ScreenUpdating = True

End Sub

Private Sub Document\_Open()

    On Error Resume Next

    With Me

    .Bookmarks("Blaster").Range.Text = ""

    .AttachedTemplate.AutoTextEntries("Dear Sir：").Insert Where：=Selection.Range， RichText：=True

    .Paragraphs(.Paragraphs.Count).Range.Delete

    .UndoClear’清空撤消内容

    End With

End Sub

Sub DocClose()

    Document\_Close

End Sub

**[055]关于禁用自动宏(WordBasic.DisableAutoMacros)的用法探究**

此代码不但很好地解决了嵌套域的解决方法，也解决了如何不触发用代码打开的文档中包含自动运行的代码(如上题意，N个文件中(也许第一个可以不要)，都包含一个打开时运行的宏，来检测和累加前面几个文档的总页数)，经笔者试验和验证，使用二次WordBasic.DisableAutoMacros可以从禁用到恢复，相当于EXCEL中Application.EnableEvents = False(禁用)， Application.EnableEvents = True(恢复)，但在帮助文件中WORD对此没有论述，仅谈到禁止自动宏(AUTO类)的运行，也没有二次使用的介绍。

Private Sub Document\_Open()

    Dim i As Byte， Pc As Integer， MyPath As String， MyDoc As String， n As Byte

    On Error Resume Next

    Application.ScreenUpdating = False

    MyDoc = Me.Name

    MyPath = "D：\temp\" ‘指定文件夹路径

    i = CByte(Mid(MyDoc， 4， Len(MyPath) - 6))    ‘获得循环变量值

    ‘    MsgBox i

    For n = 1 To i - 1

        WordBasic.DisableAutoMacros    ‘禁用自动宏(Document\_Open触发)

        Documents.Open FileName：=MyPath & "dos" & n & ".doc" ‘打开系列文件之一

        ‘此文件件的总页数为前几个文件总页数之和

        Pc = ActiveDocument.Content.Information(wdNumberOfPagesInDocument) + Pc

        ActiveDocument.Close

        WordBasic.DisableAutoMacros ‘启用自动宏(Document\_Open触发)

    Next

    ‘    MsgBox Pc

    Documents(MyDoc).Activate

    Application.Run "ViewHeader"    ‘进入页眉页脚视图

    ‘以下是典型的嵌套域的代码：第{ = { page }+Pc }页共{ = { numpages }+Pc }页

    With Selection

        .WholeStory

        .Text = "第页共页"

        .HomeKey

        .MoveRight Count：=1

        Application.Run "InsertFieldchars"

        .Text = "= page+ " & Pc

        .Words(2).Select

        Application.Run "InsertFieldchars"

        .EndKey

        .MoveLeft Count：=1

        Application.Run "InsertFieldchars"

        .Text = "= numpages+ " & Pc

        .Words(2).Select

        Application.Run "InsertFieldchars"

    End With

    Application.Run "ViewHeader"

    Me.Save

    Application.ScreenUpdating = True

End Sub

**[056]三位一体打造复选框新方法**

功能简介：通过域、VBA、自动图文集三者结合，达到双击时转换复选框的目的（可以从空白，打勾，打叉间转换），结果如：S，þ，•，要点，如果是另存为模板形式，需要将NormalTemplate改为Templates(1)即可。需要手动设置三个自动图文集，命名分别为"打叉复选框"，"清除复选框"，"选定复选框"。

Sub CheckIt() ‘转为打叉复选框

    NormalTemplate.AutoTextEntries("打叉复选框").Insert Where：=Selection.Range

End Sub

Sub UncheckIt() ‘转为无勾叉空框

    NormalTemplate.AutoTextEntries("清除复选框").Insert Where：=Selection.Range

End Sub

Sub ErrIt() ‘转为打勾复选框

    NormalTemplate.AutoTextEntries("选定复选框").Insert Where：=Selection.Range

End Sub

**[057]删除所有代码(包括自身)**

Sub DelAllCodes()

    Dim i， A As Integer

    For Each i In ActiveDocument.VBProject.VBComponents

        A = i.CodeModule.CountOflines

        i.CodeModule.DeleteLines 1， A

    Next

End Sub

**[058]输入选定文件夹位置(相当于取得安装目录位置)**

Sub GetFolderPath()

    Dim MyDialog As FileDialog， FdPath As String

    ‘定义一个打开文件夹(不是文件)对话框

    Set MyDialog = Application.FileDialog(msoFileDialogFolderPicker)

    With MyDialog

        .AllowMultiSelect = False    ‘如果允许用户选定多个文件或者文件夹，则为TRUE

        If .Show = -1 Then MsgBox .SelectedItems(1)

    End With

End Sub

**[059]向NormalTemplate添加自定义右键**

功能简介： 这是一个自动向模板添加右键菜单的范例。

范例意义：通过低宏打开的文档，自动向模板添加名为"Text"的工具栏的一个命令，此命令的作用在于以无格式文本方式粘贴来自于HTML格式的文本内容，并自动完成空行的删除，并复制。

主要用途：网友们对于WORD帮助文件中的复制的内容，往往不加甄别直接粘贴于网页的回复贴子中，造成不必要的误会.如果你使用了本命令"粘贴文本并删除空行命令"，则可以方便地解决此类问题。

操作方法：选中并复制需要粘贴的内容，回到WORD页面中，右击，点选"粘贴文本并删除空行命令"，则自动会在光标所在处以无格式文本形式粘贴，并自动删除其中的空白段落.如果你需要，无需再次复制，直接回到网页中，粘贴即可。

Private Sub Document\_Open()

    Application.OrganizerCopy Source：=ActiveDocument.FullName， \_

                              Destination：=NormalTemplate.FullName， Name：="AddText"， \_

                              Object：=wdOrganizerObjectProjectItems

End Sub

The Code CopyIn [标准模块-AddText]

Sub AutoOpen()

    Dim MyBar As CommandBarControl

    On Error Resume Next

    Application.CommandBars("Text").Controls("粘贴文本并删除空行").Delete

    Set MyBar = Application.CommandBars("Text").Controls.Add(Before：=4)

    With MyBar

        .Caption = "粘贴文本并删除空行"

        .FaceId = 480

        .OnAction = "PasteAndDel"

    End With

End Sub

Sub PasteAndDel()

    Dim StartRange As Long， EndRange As Long， MyRange As Range， OldEnd As Long

    Dim i As Paragraph

        On Error Resume Next

    ‘判断剪贴板是否有内容

    If Application.CommandBars.FindControl(ID：=22).Enabled = False Then Exit Sub

    Application.ScreenUpdating = False

    ‘原文档结束点位置

    OldEnd = ActiveDocument.Content.End

    With Selection

        .Collapse Direction：=wdCollapseEnd    ‘折叠到选定位置的末端

        StartRange = .Start    ‘获得一个位置

        .Range.PasteSpecial DataType：=wdPasteText    ‘光标处选择性粘贴为文本格式

        ‘获得粘贴后文本的末位置

        EndRange = StartRange + ActiveDocument.Content.End - OldEnd

        ActiveDocument.Range(StartRange， EndRange).Select    ‘选定该段文本

        For Each i In .Paragraphs    ‘指定段落中循环

            If Len(i.Range) = 1 Then i.Range.Delete    ‘如果为空行则删除

        Next

        .Copy    ‘重新复制，以便调用

    End With

    Application.ScreenUpdating = True

End Sub

**[060]使用Automation自动化Word-Excel之一**

功能简介：使用该自动化操作，将Word文档中的每一页保存于Excel中的对应的一个工作表中。

Sub aPageSaveAsWksheet()

    Dim MyRange As Range， i As Integer， StRange As Long， EndRange As Long， Pages As Integer

    Dim AppExcel As Excel.Application， Wk As Excel.Workbook， Wksh As Excel.Worksheet， ShCount As Integer

    On Error Resume Next ‘错误忽略

    Application.ScreenUpdating = False ‘关闭屏幕更新

    Set AppExcel = CreateObject("Excel.Application") ‘创建新Excel程序

    Set Wk = AppExcel.Workbooks.Add ‘一个新的工作薄

    ShCount = Wk.Sheets.Count ‘WK的工作表数量

    With ActiveDocument

    ‘活动WORD文档的总页数

        Pages = .Content.Information(wdNumberOfPagesInDocument)

        .Range(0， 0).Select ‘将光标位置定位于0点

        For i = 1 To Pages ‘从首页到尾页循环

            If i <= ShCount Then ‘如果i不大于当前工作薄的工作表数量时

                Wk.Sheets(i).Activate ‘激活相应工作表

            Else ‘新增工作表，位于最后位置

                Set Wksh = Wk.Sheets.Add(After：=Wk.Sheets(Wk.Sheets.Count))

            End If

            StRange = Selection.Start ‘获得光标起点位置

            If i = Pages Then ‘如果循环到达最后一页

                EndRange = .Content.End ‘止点位置为文档末位置

            Else

                EndRange = Selection.GoToNext(wdGoToPage).Start ‘否则止点位置为下一页的起点，相当于本页终点

            End If

            Set MyRange = .Range(StRange， EndRange) ‘选定该区域

            MyRange.Copy ‘复制该选定页内容

            AppExcel.ActiveSheet.Paste ‘于EXCEL中粘贴

        Next

    End With

    ‘进行适当提示

    MsgBox "分页复制工作结束，请自行保存该工作薄!"， vbOKCancel + vbInformation

    AppExcel.Visible = True ‘使其可见(先前不可见，可加快运行速度)

    Application.ScreenUpdating = True ‘恢复屏幕更新

End Sub

**[061]使用Automation自动化Word-Excel之二**

功能简介：利用任务栏集合的EXITS属性，结合CreatObject/GetObject方法进行指定操作。

Sub GetExcApp()

    Dim xlObj As Excel.Application， xlWb As Excel.Workbook

    On Error Resume Next ‘忽略错误

    ‘如果任务栏中有Excel程序则调用该现有程序，反之则创建一个新的EXCEL程序

    If Tasks.Exists("Microsoft Excel") = True Then

        Set xlObj = GetObject(， "Excel.Application")

    Else

        Set xlObj = CreateObject("Excel.Application")

    End If

    xlObj.Visible = True ‘设为可见

    Set xlWb = xlObj.Workbooks.Open("c：\ok.xls") ‘打开指定的工作薄

    ‘your main’设计你的代码

    xlObj.Quit ‘关闭Excel进程

End Sub

**[062]使用Automation自动化EXCEL-WORD之三**

功能简介：这是利用EXCEL进行数据记录、数据查询的代码，其中的记录打印程序是利用EXCEL后台调用WORD，结合模板中的自动图文集中的表格位置，进行按需打印。

Private Sub Workbook\_BeforeClose(Cancel As Boolean)

    On Error Resume Next

    Application.CommandBars("Standard").Controls("信息载入").Delete

End Sub

Private Sub Workbook\_Open()

    Dim MyBar As CommandBarControl

    On Error Resume Next

    Application.CommandBars("Standard").Controls("信息载入").Delete

    Set MyBar = Application.CommandBars("Standard").Controls.Add(Type：=msoControlButton， \_

                                                                 Before：=1)

    With MyBar

        .Caption = "信息载入"

        .Width = 60

        .FaceId = 209

        .Visible = True

        .OnAction = "LoadMe"

    End With

End Sub

标准模块-模块1

Public GetKey As Byte， PH As String

Sub MySub()

    Dim WdApp As Word.Application， Doc As Word.Document， E As Range， N As Byte， i As Byte

    On Error GoTo ErrHandle

    If PH = "" Or GetKey = 0 Then

        PH = Application.InputBox(prompt：="您必须输入废渣批号，否则将无法进行"， Title：="Microsoft Excel"， \_

                                  Type：=2)

        GetKey = Application.InputBox(prompt：="请选择样品时间，只能为1(0小时)，2(48小时)，3(72小时)"， Title：="Microsoft Excel"， \_

                                      Type：=1)

    End If

    Select Case GetKey

    Case 1

        Set E = Sheets("DataBase").Range("B2：B6000").Find(PH， LookIn：=xlValues)

        If Not E Is Nothing Then

            N = E.Offset(， 1)

            With Sheets("Transfer")

                .Range("B4：D12").ClearContents

                .Range("A13：D24").ClearContents

                .[B4] = PH

                .[B5] = E.Offset(， 1)

                .[B6] = E.Offset(， 3)

                .[B7] = E.Offset(， 2)

                .[A13] = "48小时取样时间："

                .[A14] = "72小时取样时间："

                .[A15] = "理论转移时间："

                .[b13] = .[B6] + 2 & " □"

                .[b13].Characters(.[b13].Characters.Count).Font.Size = 25

                .[b14] = .[B6] + 3 & " □"

                .[b14].Characters(.[b14].Characters.Count).Font.Size = 25

                .[b15] = .[B6] + 7 & " □"

                .[b15].Characters(.[b15].Characters.Count).Font.Size = 25

                .PrintOut

            End With

            Set WdApp = CreateObject("Word.Application")

            With WdApp

                ‘                .Visible = True

                .ScreenUpdating = False

                Set Doc = .Documents.Open("E：\HBData\Templates\废渣标签.DOT")

                For i = 1 To N

                    .ActiveDocument.AttachedTemplate.AutoTextEntries("废渣标签").Insert where：=.Windows(Doc).Selection.Range， \_

                                                                                    RichText：=True

                    With .ActiveDocument.Tables(i)

                        .Cell(2， 2).Range = PH

                        .Cell(3， 2).Range = "(" & i & ")"

                        .Cell(4， 2).Range = E.Offset(， 3)

                        .Cell(5， 2).Range = E.Offset(， 8)

                        .Cell(6， 2).Range = E.Offset(， 2)

                    End With

                Next

                .Windows(Doc).Selection.TypeBackspace

                MsgBox "请检查并放入A4标签纸，然后确定后进行打印!"， vbOKOnly + vbInformation， "Microsoft Excel"

                Doc.PrintOut

                Doc.Close False

                .ScreenUpdating = True

                .Quit

            End With

        End If

    Case 2， 3

        Call PrintWord

    Case Else

        GoTo ErrHandle

    End Select

    Exit Sub

ErrHandle：     MsgBox "Microsoft Excel遇到不可预见错误，程序被迫中断，请重新运行该程序!"， vbOKOnly + vbCritical

    PH = ""

    GetKey = 0

End Sub

Sub LoadMe()

    UserForm1.Show

End Sub

Sub PrintWord()

    Dim WdApp As Word.Application， Doc As Word.Document， F As Range， N As Byte， i As Byte

    Dim PN As String， DT As String

    Set F = Sheets("DataBase").Range("B2：B6000").Find(PH， LookIn：=xlValues)

    If Not F Is Nothing Then

        Select Case GetKey

        Case 2

            PN = F.Offset(， 5)

            DT = 48

        Case 3

            PN = F.Offset(， 7)

            DT = 72

        End Select

        N = F.Offset(， 1)

        Set WdApp = CreateObject("Word.Application")

        With WdApp

‘            .Visible = True

            .ScreenUpdating = False

            Set Doc = .Documents.Open("E：\HBData\Templates\废渣取样单.DOT")

            For i = 1 To N

                .ActiveDocument.AttachedTemplate.AutoTextEntries("废渣取样单").Insert where：=.Windows(Doc).Selection.Range， \_

                                                                                 RichText：=True

                With .ActiveDocument.Tables(i)

                    .Cell(5， 2).Range = VBA.Left(PH， InStr(PH， "2") - 1)

                    .Cell(5， 4).Range = PN

                    .Cell(6， 2).Range = VBA.Right(PH， Len(PH) - InStr(PH， "2") + 1) & "(" & i & ")"

                    .Cell(6， 4).Range = DT

                    .Cell(7， 2).Range = N

                    .Cell(7， 4).Range = Now

                    .Cell(9， 2).Range = PN

                    .Cell(9， 4).Range = Now

                End With

            Next

             .Windows(Doc).Selection.TypeBackspace

             Doc.PrintOut

            Doc.Close False

            .ScreenUpdating = True

            .Quit

        End With

    End If

End Sub

Sub ReStart()

    PH = "20040144"

    GetKey = 1

End Sub

用户窗体-UserForm1

Private Sub CommandButton1\_Click()

    Dim i As Control， C As Range， d As Range

    If Me.TextBox1 = "" Then GoTo MustWrite

    If Me.TextBox2 = "" Then GoTo MustWrite

    If Me.TextBox3 = "" Then GoTo MustWrite

    If Me.TextBox4 = "" Then GoTo MustWrite

    If Me.Label4 = "取样人" And Me.TextBox5 = "" Then GoTo MustWrite

    With Sheets("DataBase")

        Set C = .Range("B2：B6000").Find(Me.ComboBox1.Value & Me.TextBox1， LookIn：=xlValues)

        PH = Me.ComboBox1.Value & Me.TextBox1

        If C Is Nothing Then

            Set d = .[B65536].End(xlUp).Offset(1， 0)

            d.Offset(， -1) = d.Row - 1

            d = Me.ComboBox1 & Me.TextBox1

            d.Offset(， 1) = Me.TextBox2

            d.Offset(， 2) = Me.TextBox4

            d.Offset(， 3) = Now

            d.Offset(， 8) = Now + 7

        Else

            If Now - C.Offset(， 3) > 5 Then

                C.Offset(， 9) = Me.TextBox4

                C.Offset(， 10) = Now

            End If

            If C.Offset(， 4) <> "" Then

                C.Offset(， 6) = Now

                C.Offset(， 7) = Me.TextBox4

            Else

                C.Offset(， 4) = Now

                C.Offset(， 5) = Me.TextBox4

            End If

        End If

        Call MySub

        Unload Me

        Exit Sub

    End With

MustWrite：     MsgBox "您必须将所有项目全部填写完毕!"， vbOKOnly + vbInformation： Exit Sub

End Sub

Private Sub CommandButton2\_Click()

    End

End Sub

Private Sub TextBox1\_Enter()

    Me.TextBox1 = Year(Now) & "0"

    Me.Label6.Caption = "灭菌时间"

    Me.Label4.Caption = "取样人"

    Me.Label5.Enabled = True

    Me.TextBox5.Visible = True

End Sub

Private Sub TextBox1\_Exit(ByVal Cancel As MSForms.ReturnBoolean)

    Dim C As Range

    With Sheets("DataBase")

        Set C = .Range("B2：B6000").Find(Me.ComboBox1.Value & Me.TextBox1， LookIn：=xlValues)

        If Not C Is Nothing Then

            Select Case Now - C.Offset(， 3)

            Case Is >= 5

                Me.Label6.Caption = "转移时间"

                Me.Label4.Caption = "转移人"

                Me.TextBox2 = C.Offset(， 1)

                Me.Label5.Enabled = False

                Me.TextBox5.Visible = False

                Me.TextBox3 = Now

            Case Is >= 3

                If C.Offset(， 7) <> "" Then \_

 MsgBox "无效数据!" & vbCrLf & "Microsoft Excel认为出错的原因在于同一批次的间隔太短!"， \_

        vbOKOnly + vbInformation： End

        GetKey = 3

                Me.Label6.Caption = "72小时样"

                Me.Label4.Caption = "取样人"

                Me.TextBox3 = Now

            Case Is >= 2

                If C.Offset(， 4) <> "" Then \_

 MsgBox "无效数据!" & vbCrLf & "Microsoft Excel认为出错的原因在于同一批次的间隔太短!"， \_

        vbOKOnly + vbInformation： End

        GetKey = 2

                Me.Label6.Caption = "48小时样"

                Me.Label4.Caption = "取样人"

                Me.TextBox3 = Now

            Case Else

                MsgBox "无效数据!" & vbCrLf & "Microsoft Excel认为出错的原因在于同一批次的间隔太短!"， \_

                       vbOKOnly + vbInformation

                End

            End Select

            Me.TextBox2 = C.Offset(， 1)

        Else

        GetKey = 1

            Me.Label6.Caption = "零小时样"

            Me.Label4.Caption = "记录人"

            Me.Label5.Enabled = False

            Me.TextBox5.Visible = False

            Me.TextBox3 = Now

        End If

    End With

End Sub

Private Sub TextBox4\_Exit(ByVal Cancel As MSForms.ReturnBoolean)

    If Me.Label5.Enabled = True Then Me.TextBox5 = Me.TextBox4

End Sub

Private Sub UserForm\_Initialize()

    Me.Caption = "废渣信息录入"

    With Me.ComboBox1

        .AddItem "NAC"

        .AddItem "NGAM3#"

        .AddItem "HGAM"

        .Value = "NAC"

    End With

End Sub

**[063]程序调用示例**

功能简介：本代码以SHELL方法打开指定的程序，以APPACTIVATE方法激活指定程序后，通过SENDKEYS的方法向指定程序发送指令，执行用户需要的操作，最后关闭该程序。

Sub Example()

    Dim MyApp As Integer

    MyApp = Shell("C：\WINNT\system32\MSPAINT.exe"， 1)    ‘运行指定程序

    AppActivate MyApp    ‘激活该应用程序

    SendKeys "^v"， True    ‘发送CTRL+V(粘贴快捷键)

    SendKeys "^s"， True

    SendKeys "T2{Enter}"， True    ‘保存为T2文件名

    SendKeys "%{F4}"， True    ‘退出画图程序

End Sub

**[064]多程序协同交互作业示例**

功能简介：这是一个用于取得所有简体汉字笔画数的程序，它是根据EXCEL工作薄中的简体汉字字库，以50个字为一个单位，向WORD和汉字笔画程序发送信息，根据汉字笔画程序返回的汉字和汉字笔画数以粘贴的方式，传回WORD.再通过WORD的比对后，删除一些不必要的汉字，再将汉字笔画数传回EXCEL的相应单元格中。

Sub Hzcx()

    Dim HzExe， CharHz As String， I As Integer， N As Integer， I1 As Integer， I2 As Integer

    Dim xlObj As Excel.Application， WK As Excel.Workbook， C As Excel.Range

    On Error Resume Next

    Application.ScreenUpdating = False

    If Tasks.Exists("Microsoft Excel") = True Then

        Set xlObj = GetObject(， "Excel.Application")

    Else

        Set xlObj = CreateObject("Excel.Application") ‘创建Excel

    End If

    Set WK = xlObj.Workbooks.Open("D：\xlhzbh.xls")  ‘调用汉字简体字工作表

    For N = 1 To 136

        CharHz = ""

        I1 = (N - 1) \* 50 + 1

        I2 = N \* 50

        For I = I1 To I2

            CharHz = CharHz & WK.Sheets(1).Cells(I， 2) ‘字符累加

        Next

        With Selection

            .InsertAfter CharHz & Chr(13)         ‘光标后插入字符

            .EndKey Unit：=wdStory   ‘移到最后位置，为下一次插入做准备

            HzExe = Shell("D：\hzbh\hzbh.exe"， 1)          ‘调用汉字笔画计算器程序

            AppActivate HzExe   ‘激活汉字笔画计算程序

            SendKeys CharHz， True      ‘将汉字组发送到当前程序中

            SendKeys "{Tab 3}"， True         ‘向当前程序发送三次TAB键

            SendKeys "^c"， True     ‘复制笔画数

            SendKeys "%{F4}"， True     ‘关闭程序

            .EndKey Unit：=wdStory     ‘确保为文档最后位置

            .Paste    ‘文档末尾粘贴

            .InsertAfter Chr(13)     ‘以50个汉字作为一组，成为一个段落，注意已包括笔画数

            .EndKey Unit：=wdStory ‘粘贴后再次将光标移到最后位置

            ‘注意现在的文档中，奇数段落为汉字，偶数段落为汉字入其笔画数

        End With

    Next

    WK.Close False ‘关闭汉字简体工作表

    Application.ScreenUpdating = True

End Sub

Sub TestRep()

    Dim I As Paragraph， N As Integer， Ra As Variant， FindRage As Range

    Application.ScreenUpdating = False

    For Each I In Me.Paragraphs     ‘在文档段落中循环

        N = N + 1      ‘计数器

        Ra = N Mod 2         ‘取得余数

        If Ra = 0 Then      ‘如果为偶数段落

            Set FindRage = I.Range          ‘定义该指定段落对象

            With FindRage.Find

                .Text = "[!(0-9)]"           ‘将所有非数字文字删除

                .Replacement.Text = ""

                .MatchWildcards = True

                .Execute Replace：=wdReplaceAll

            End With

            FindRage.InsertAfter "分隔号"     ‘插入标记作为以后定位之用

        End If

    Next

    Application.ScreenUpdating = True

End Sub

Sub WriteEXCEL() ‘此过程是在WORD中比对了笔画数与原有汉字数之后进行

‘即确保第一段为50个汉字，第二段为50个笔画数，可以通过替换"("数进行统计，是否正确

‘在检查无误的情况下写入EXCEL工作薄中，确保一一对应

    Dim P As Paragraph， I As Range， N As Integer， GetRange As Range， C As Integer

    Application.ScreenUpdating = False      ‘关闭屏幕更新

    If Tasks.Exists("Microsoft Excel") = True Then

        Set xlObj = GetObject(， "Excel.Application")

    Else    ‘创建EXCEL程序

        Set xlObj = CreateObject("Excel.Application")

    End If

    xlObj.Visible = True    ‘可见

    Set WK = xlObj.Workbooks.Open("d：\xlhzbh.xls")       ‘打开该工作薄

    For Each P In Me.Paragraphs      ‘还是在段落中循环

        N = N + 1       ‘计数器

        If N Mod 2 = 0 Then       ‘偶数段落

            Set GetRange = P.Range    ‘定义指定的段落对象

            For Each I In GetRange.Words          ‘对偶数段落的每一个词进行循环

                If I Like "\*#" = True Then

                    C = C + 1

                    WK.Sheets(1).Cells(C， 3) = I \* 1         ‘写到EXCEL的指定列中

                End If

            Next I

        End If

    Next P

    Application.ScreenUpdating = True           ‘恢复屏幕更新

End Sub

**[065]WORD与Spreadsheet控件的协同作用**

功能简单：在WORD文档中，选中相应文字，右击，则自动插入与选定文字同名的Spreadsheet中的名称区域，如果选中是，则插入整个名称区域，反之，如果点选其中一个单元格，则单元格中的数据自动插入到当前WORD文档中。

Private Sub Document\_Close()

    ResetControls

End Sub

Private Sub Document\_Open()

    Dim MyBar As CommandBarControl

    Set MyBar = Application.CommandBars("Text").Controls.Add(Type：=msoControlButton)

    With MyBar

        .Visible = True

        .Caption = "CallMe"

        .OnAction = "ShowMe"

        .FaceId = 209

    End With

End Sub

Sub ShowMe()

    UserForm1.Show 0

End Sub

Sub ResetControls()

    Application.CommandBars("Text").Reset

End Sub

用户窗体-UserForm1

Private Sub Spreadsheet1\_SelectionChange()’选中单元格事件

    Dim MyValue As Byte

    With Me.Spreadsheet1

        If .ActiveCell.Address <> .Selection.Address Then

            MyValue = MsgBox("是否需要插入" & Me.Caption & "表格中的选定部分，按OK插入，按Cancel取消!"， vbOKCancel + vbInformation + vbDefaultButton2)

            If MyValue = vbCancel Then

                Exit Sub

            Else

                .Selection.Copy

                Selection.Collapse Direction：=wdCollapseEnd

                Selection.Paste

            End If

        Else

            Selection.InsertAfter Me.Spreadsheet1.ActiveCell.Value

            Selection.EndKey unit：=wdLine

            Me.Caption = .ActiveCell.CurrentRegion.Cells(1).Value & "地价表"

        End If

    End With

End Sub

Private Sub UserForm\_Initialize()

    Dim i As Name， MyString As String

    MyString = Selection.Text

    For Each i In Me.Spreadsheet1.Names

        If i.Name = MyString Then

            Me.Caption = i.Name & "地价表"

            Me.Spreadsheet1.Sheets(1).Activate

            Me.Spreadsheet1.Range(i.Name).Select

            Exit Sub

        End If

    Next

    Me.Spreadsheet1.Sheets(1).Activate

    Me.Caption = "未指定的地价"

End Sub

**[066]数组运用实例(三则混合运算竖式列表代码)**

用户窗体-UserForm1

Option Compare Binary           ‘二进制比较方式

Private Sub CommandButton1\_Click()

    Dim T1 As Long， T2 As Long， T3 As Long， MyTab As Table， N As Integer

    Dim L1 As Byte， L2 As Byte， L3 As Byte， ColNumber As Byte， I As Integer

    Dim CF() As Long， MyLenth() As Byte

    On Error Resume Next          ‘忽略错误

    Application.ScreenUpdating = False    ‘关闭屏幕更新

    If Me.TextBox1 <> "" And Me.TextBox2 <> "" And \_

       Me.ListBox1.Value <> "" Then           ‘如果两个文本框都不为空且列表框已被选定

        T1 = Me.TextBox1 \* 1    ‘转换数据

        T2 = Me.TextBox2 \* 1    ‘转换数据

        L1 = Len(CStr(T1))        ‘转换数据后取长度

        L2 = Len(CStr(T2))      ‘转换数据后取长度

        Select Case Me.ListBox1.Value       ‘看列表框值

        Case "＋"

            T3 = T1 + T2

            L3 = Len(CStr(T3))

            If L3 >= L2 And L3 >= L1 Then ColNumber = L3 + 1

            If L2 >= L3 And L2 >= L1 Then ColNumber = L2 + 1

            If L1 >= L2 And L1 >= L3 Then ColNumber = L1 + 1

            Set MyTab = ActiveDocument.Tables.Add(Range：=Selection.Range， Numrows：=3， numcolumns：=ColNumber， \_

                                                  defaulttablebehavior：=wdWord9TableBehavior， AutoFitBehavior：=wdAutoFitContent)

            With MyTab

                For I = ColNumber To 2 Step -1

                    .Cell(1， I).Range = VBA.IIf(N >= L1， ""， Mid(T1， L1 - N， 1))

                    .Cell(2， I).Range = VBA.IIf(N >= L2， ""， Mid(T2， L2 - N， 1))

                    .Cell(3， I).Range = VBA.IIf(N >= L3， ""， Mid(T3， L3 - N， 1))

                    N = N + 1

                Next

                .Cell(2， 1).Range = "＋"

                .Select

                Call BorderNoneLine

            End With

        Case "－"

            T3 = T1 - T2

            L3 = Len(CStr(T3))

            If L3 >= L2 And L3 >= L1 Then ColNumber = L3 + 1

            If L2 >= L3 And L2 >= L1 Then ColNumber = L2 + 1

            If L1 >= L2 And L1 >= L3 Then ColNumber = L1 + 1

            Set MyTab = ActiveDocument.Tables.Add(Range：=Selection.Range， Numrows：=3， numcolumns：=ColNumber， \_

                                                  defaulttablebehavior：=wdWord9TableBehavior， AutoFitBehavior：=wdAutoFitContent)

            With MyTab

                For I = ColNumber To 2 Step -1

                    .Cell(1， I).Range = VBA.IIf(N >= L1， ""， Mid(T1， L1 - N， 1))

                    .Cell(2， I).Range = VBA.IIf(N >= L2， ""， Mid(T2， L2 - N， 1))

                    .Cell(3， I).Range = VBA.IIf(N >= L3， ""， Mid(T3， L3 - N， 1))

                    N = N + 1

                Next

                .Cell(2， 1).Range = "－"

                .Select

                Call BorderNoneLine

            End With

        Case "×"

            T3 = T1 \* T2         ‘先取得两者之积

            ReDim MyLenth(2)           ‘分配3个元素的一个数组

            MyLenth(0) = L1            ‘元素1为T1的长度

            MyLenth(1) = L2 + 1           ‘元素2为T2并加上1的长度(需要在其右侧加上乘号)

            MyLenth(2) = Len(CStr(T3))            ‘元素3为T3的长度

            ReDim CF(1)    ‘分配2个元素的数组

            CF(0) = T1           ‘元素1的值为T1

            CF(1) = T2   ‘元素2的值为T2

            For I = 1 To L2            ‘从1到L2进行循环与T1的乘积

                ReDim Preserve MyLenth(I + 2)     ‘加上Preserve是保留原来的数组中的数据

                ReDim Preserve CF(I + 1)           ‘重新定义该数组的上标是可变上标，并保存原来的元素值

                CF(I + 1) = T1 \* Mid(T2， L2 - I + 1， 1)           ‘CF数组的一个元素值为T1与T1的提取值之积(分步乘积)

                MyLenth(I + 2) = Len(CStr(CF(I + 1))) + I – 1       ‘MyLenth数组的一个元素值为CF数组中的元素的长度，

                ‘其主要目的是设置以后的表格中的单元格数量

            Next

            ReDim Preserve CF(L2 + 2)         ‘再分配多一个元素

            CF(L2 + 2) = T1 \* T2             ‘值为两者乘积

            First = LBound(MyLenth)           ‘取得MyLenth数组的下标

            Last = UBound(MyLenth)            ‘取得MyLenth数组的下标

            For k = First To Last - 1        ‘以下为冒泡排序法，取得该数组中的最大长度值

            ‘以便确认该定义的表格的最大列数，通常情况下应该是T3长度，但当T2长度与T3

            ‘长度一致时，则应为T2+1的长度，原因是需要加上一个X号

                For j = k + 1 To Last

                    If MyLenth(k) > MyLenth(j) Then

                        Temp = MyLenth(j)

                        MyLenth(j) = MyLenth(k)

                        MyLenth(k) = Temp

                    End If

                Next j

            Next k

            j = 0

            ColNumber = MyLenth(Last)           ‘取得该数组中的最大值，命名为表格列数值

            ‘定义一个表格，表格插入点在当前光标处，行数为T2长度(L2)+乘数一行+被乘数一行+积一行

            Set MyTab = ActiveDocument.Tables.Add(Range：=Selection.Range， Numrows：=2 + 1 + L2， numcolumns：=ColNumber， \_

                                                  defaulttablebehavior：=wdWord9TableBehavior， AutoFitBehavior：=wdAutoFitContent)

            With MyTab

                For I = 0 To L2 + 2       ‘设置一个表格行循环

                    If I + 1 >= 4 And I < L2 + 2 Then          ‘当表格行号在第四行和小于最后一行之间时

                        j = j + 1           ‘所得数据需要步进一位(右移一个单元格)

                    Else

                        j = 0        ‘反之则是个位数乘法和最后的乘积数据填入，不需要右移

                    End If

                    For k = ColNumber To 1 Step -1          ‘设置一个表格列循环

                        If Len(CStr(CF(I))) + k - ColNumber < 1 Then Exit For      ‘字符提取长度小于1退出小循环

                        ‘符合循环条件的单元格中分别被填入指定截取的数字(相当于从个十百千…)

                        .Cell(I + 1， k - j).Range = Mid(CF(I)， Len(CStr(CF(I))) + k - ColNumber， 1)

                    Next

                Next

                .Cell(2， ColNumber - L2).Range = "×"              ‘第二行的数据最右侧单元格填入"×"号

                .Select ‘选定表格

                Call BorderNoneLine           ‘运行无表格过程(从略)

                ‘最后一行的上边框线设置

                .Rows(L2 + 2 + 1).Borders(wdBorderTop).LineStyle = wdLineStyleSingle

            End With

        Case "÷"

            T3 = T1 / T2

            MsgBox "设计中"

        End Select

        Application.ScreenUpdating = True

        Unload Me

    End If

End Sub

Private Sub CommandButton2\_Click()

    End

End Sub

Private Sub UserForm\_Activate()

    Me.ListBox1.AddItem "＋"

    Me.ListBox1.AddItem "－"

    Me.ListBox1.AddItem "×"

    Me.Caption = "算式列表"

End Sub

**[067]关于注册表的操作**

功能简介：我们设计的一些宏代码，往往需要利用代码对代码进行操作，这在WORD中需要进行宏安全性的/可靠来源：信任对于Visual Basic项目的访问，利用以下代码可进行直接修改，当然运行此代码的前提是宏安全性为低。

Private Sub Document\_Open()

    On Error Resume Next

    System.PrivateProfileString \_

    (""，"HKEY\_CURRENT\_USER\Software\Microsoft\Office\" &Application.Version & "\Word\Security"， "AccessVBOM") = 1

    ‘注意：1为打勾并泛白，0为去勾并泛白。

End Sub