**Word Count Frequency of a Document**

The following macro prompts the user for a word, and then counts the number of times that word appears in the document. It will continue to ask for another word until the user clicks on the Cancel button.

Sub FindWords()

Dim sResponse As String

Dim iCount As Integer

' Input different words until the user clicks cancel

Do

' Identify the word to count

sResponse = InputBox( \_

Prompt:="What word do you want to count?", \_

Title:="Count Words", Default:="")

If sResponse > "" Then

' Set the counter to zero for each loop

iCount = 0

Application.ScreenUpdating = False

With Selection

.HomeKey Unit:=wdStory

With .Find

.ClearFormatting

.Text = sResponse

' Loop until Word can no longer

' find the search string and

' count each instance

Do While .Execute

iCount = iCount + 1

Selection.MoveRight

Loop

End With

' show the number of occurences

MsgBox sResponse & " appears " & iCount & " times"

End With

Application.ScreenUpdating = True

End If

Loop While sResponse <> ""

End Sub

If you want to determine all the unique words in a document, along with how many times each of them appears in the document, then a different approach is needed. The following macro will do just that.

Sub WordFrequency()

Const maxwords = 9000 'Maximum unique words allowed

Dim SingleWord As String 'Raw word pulled from doc

Dim Words(maxwords) As String 'Array to hold unique words

Dim Freq(maxwords) As Integer 'Frequency counter for unique words

Dim WordNum As Integer 'Number of unique words

Dim ByFreq As Boolean 'Flag for sorting order

Dim ttlwds As Long 'Total words in the document

Dim Excludes As String 'Words to be excluded

Dim Found As Boolean 'Temporary flag

Dim j, k, l, Temp As Integer 'Temporary variables

Dim ans As String 'How user wants to sort results

Dim tword As String '

' Set up excluded words

Excludes = "[the][a][of][is][to][for][by][be][and][are]"

' Find out how to sort

ByFreq = True

ans = InputBox("Sort by WORD or by FREQ?", "Sort order", "WORD")

If ans = "" Then End

If UCase(ans) = "WORD" Then

ByFreq = False

End If

Selection.HomeKey Unit:=wdStory

System.Cursor = wdCursorWait

WordNum = 0

ttlwds = ActiveDocument.Words.Count

' Control the repeat

For Each aword In ActiveDocument.Words

SingleWord = Trim(LCase(aword))

'Out of range?

If SingleWord < "a" Or SingleWord > "z" Then

SingleWord = ""

End If

'On exclude list?

If InStr(Excludes, "[" & SingleWord & "]") Then

SingleWord = ""

End If

If Len(SingleWord) > 0 Then

Found = False

For j = 1 To WordNum

If Words(j) = SingleWord Then

Freq(j) = Freq(j) + 1

Found = True

Exit For

End If

Next j

If Not Found Then

WordNum = WordNum + 1

Words(WordNum) = SingleWord

Freq(WordNum) = 1

End If

If WordNum > maxwords - 1 Then

j = MsgBox("Too many words.", vbOKOnly)

Exit For

End If

End If

ttlwds = ttlwds - 1

StatusBar = "Remaining: " & ttlwds & ", Unique: " & WordNum

Next aword

' Now sort it into word order

For j = 1 To WordNum - 1

k = j

For l = j + 1 To WordNum

If (Not ByFreq And Words(l) < Words(k)) \_

Or (ByFreq And Freq(l) > Freq(k)) Then k = l

Next l

If k <> j Then

tword = Words(j)

Words(j) = Words(k)

Words(k) = tword

Temp = Freq(j)

Freq(j) = Freq(k)

Freq(k) = Temp

End If

StatusBar = "Sorting: " & WordNum - j

Next j

' Now write out the results

tmpName = ActiveDocument.AttachedTemplate.FullName

Documents.Add Template:=tmpName, NewTemplate:=False

Selection.ParagraphFormat.TabStops.ClearAll

With Selection

For j = 1 To WordNum

.TypeText Text:=Trim(Str(Freq(j))) \_

& vbTab & Words(j) & vbCrLf

Next j

End With

System.Cursor = wdCursorNormal

j = MsgBox("There were " & Trim(Str(WordNum)) & \_

" different words ", vbOKOnly, "Finished")

End Sub

When you open a document and run this macro, you are asked if you want to create a list sorted by word or by frequency. If you choose word, then the resulting list is shown in alphabetical order. If you choose frequency, then the resulting list is in descending order based on how many times the word appeared in the document.

While the macro is running, the status bar indicates what is happening. Depending on the size of your document and the speed of your computer, the macro may take a while to complete. (I ran it with a 719-page document with over 349,000 words and it took about five minutes to complete.)

Note that there is a line in the macro that sets a value in the Excludes string. This string contains words that the macro will ignore when putting together the word list. If you want to add words to the exclusion list, simply add them to the string, between [square brackets]. Also, make sure the exclusion words are in lowercase.