# Top 5 Macros for Fiction Writers

*For a full explanation on how each of these macros work, please see my blog post* [*Macros for Fiction Writers*](https://wintersediting.com/macros-fiction-writers-1/)*.*

*In the Table of Contents below, click CTRL + MacroTitle (CMD + MacroTitle on Mac) to be taken directly to a specific macro.*

*If you would like to copy all macros together so you can install them all at once, I have compiled all the macros without headings separating them in the final appendix at the end of this document. To highlight them quickly, click CTRL + F. Then in the box that appears on the left, select “Headings.” Then right click on the title “Appendix” and click “Select Heading and Content.” You’ll then be able to copy and paste all five macros at once into your macro installer.*

#### Table of Contents

[The HeadingsWordCount Macro 2](#_Toc148346496)

[The CatchPhrase Macro 2](#_Toc148346497)

[The ProperNounAlyse Macro 9](#_Toc148346498)

[The ChronologyChecker Macro 31](#_Toc148346499)

[The WordsPhrasesInContext Macro 35](#_Toc148346500)

[Appendix: All the Macros Together 40](#_Toc148346501)

## The HeadingsWordCount Macro

***Source: The Microsoft Community Forum,*** [***https://answers.microsoft.com/en-us/msoffice/forum/all/macro-to-compute-headings-word-counts/8bba18a7-e931-41d7-9481-1bcff364dd3a***](https://answers.microsoft.com/en-us/msoffice/forum/all/macro-to-compute-headings-word-counts/8bba18a7-e931-41d7-9481-1bcff364dd3a)

Sub HeadingsWordcount()

Application.ScreenUpdating = False

Dim RngHd As Range, h As Long, strOut As String

h = CLng(InputBox("Input the Heading level (e.g. 1) for the heading spans to count", "Heading Span Word Counter", 1))

If (h < 1) Or (h > 9) Then Exit Sub

With ActiveDocument.Range

With .Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = ""

.Style = "Heading " & h

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindStop

.Format = True

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

.Execute

End With

Do While .Find.Found

Set RngHd = .Paragraphs(1).Range

Set RngHd = RngHd.GoTo(What:=wdGoToBookmark, Name:="\HeadingLevel")

With RngHd

strOut = strOut & .ComputeStatistics(wdStatisticWords) - .Paragraphs.First.Range.ComputeStatistics(wdStatisticWords) & vbTab & .Paragraphs.First.Range.Text

End With

.Start = RngHd.End

.Find.Execute

Loop

End With

Set RngHd = Nothing

ActiveDocument.Range.InsertAfter vbCr & "The following word counts are associated with each level " & h & " heading:" & vbCr & strOut

Application.ScreenUpdating = True

End Sub

## The CatchPhrase Macro

***Source: Paul Beverley,*** [***http://www.archivepub.co.uk/book.html***](http://www.archivepub.co.uk/book.html)

**Sub CatchPhrase()**

' Paul Beverley - Version 18.05.18

' Searches for repeated phrases/sentences

group\_a = "25, 6(4), 5(8)"

group\_a = "4(4), 3(9)"

group\_b = "6(3), 5(8), 4(10)"

group\_c = "7(3), 6(5), 5(10), 4(15)"

' Number of spaces times n

mySpaces = " ": n = 20

highlightFinds = False

highlightFinds = True

myColour = wdYellow

goExtraFast = True

goExtraFast = False

giveSpeedWarning = True

stopAndShowTime = False

myDots = ".... "

If Application.Visible = False Then Application.Visible \_

= True: Exit Sub

myLap = 1

Set rng = ActiveDocument.Content

rng.End = 200

If LCase(rng.Text) <> rng.Text Then

If giveSpeedWarning = True Then

myResponse = MsgBox("Preparing words file. This may take some time." & vbCr \_

& vbCr & "Please ignore any ""Not Responding"" warnings." \_

& vbCr & vbCr & "Click Yes to start.", vbQuestion \_

+ vbYesNo, "WordsPhrasesInContext")

If myResponse <> vbYes Then Exit Sub

Else

StatusBar = "Preparing words file. This may take some time."

End If

Set rng = ActiveDocument.Content

Documents.Add

Selection.Text = LCase(rng.Text)

For i = 1 To 6

sps = sps & " "

Next i

' Remove all except pure text, hyphens and apostrophes

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "'"

.Wrap = wdFindContinue

.Replacement.Text = "jqjq"

.MatchCase = False

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

DoEvents

StatusBar = sps & "Preparing words file. This may take some time. Six..."

DoEvents

.Text = "[!a-zA-Z,\- ]"

.MatchWildcards = True

.Replacement.Text = " "

.Execute Replace:=wdReplaceAll

DoEvents

StatusBar = sps & "Preparing words file. This may take some time. Five..."

DoEvents

.Text = " [ ,-]{1,}"

.Replacement.Text = " "

.MatchWildcards = True

.Execute Replace:=wdReplaceAll

DoEvents

StatusBar = sps & "Preparing words file. This may take some time. Four..."

DoEvents

.Text = ","

.Replacement.Text = "cmcm"

.Execute Replace:=wdReplaceAll

DoEvents

StatusBar = sps & "Preparing words file. This may take some time. Three.."

DoEvents

.MatchWildcards = True

.Text = "-{1,}"

.Replacement.Text = "cqcq"

.Execute Replace:=wdReplaceAll

DoEvents

StatusBar = sps & "Preparing words file. This may take some time. Two..."

DoEvents

.Text = " [a-hj-z] "

.Replacement.Text = " "

.Execute Replace:=wdReplaceAll

DoEvents

StatusBar = sps & "Preparing words file. This may take some time. One!"

DoEvents

.Text = " {2,}"

.Replacement.Text = " "

.Execute Replace:=wdReplaceAll

End With

End If

CR = vbCr: CR2 = CR & CR

For j = 1 To n

sps = sps & mySpaces

Next j

myPrompt = "a = " & group\_a & CR2

myPrompt = myPrompt & "b = " & group\_b & CR2

myPrompt = myPrompt & "c = " & group\_c & CR2 & CR

myPrompt = myPrompt & "t = Test to estimate the ETA" & CR2

Do

myChoice = InputBox(myPrompt, "CatchPhrase", "a")

If myChoice = "" Then Beep: Exit Sub

Loop Until InStr("abct", myChoice) > 0 \_

Or InStr("123456789", Left(myChoice, 1)) > 0

Set resultsDoc = ActiveDocument

Set rng0 = ActiveDocument.Content

Documents.Add

Set wordsDoc = ActiveDocument

Set rng = wordsDoc.Content

rng.Text = LCase(rng0.Text)

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "@@@@@"

.Wrap = wdFindContinue

.Replacement.Text = ""

.Forward = True

.MatchCase = False

.MatchWildcards = False

.Execute

End With

If rng.Find.Found Then

rng.End = wordsDoc.Content.End

rng.Delete

End If

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = " {2,}"

.MatchWildcards = True

.Replacement.Text = " "

.Execute Replace:=wdReplaceAll

End With

Selection.HomeKey Unit:=wdStory

Dim numWds(20) As Integer

Dim numShows(20) As Integer

Dim myList(4) As String

Dim myCount As Integer

myList(1) = group\_a

myList(2) = group\_b

myList(3) = group\_c

myList(4) = "5,"

If LCase(myChoice) <> UCase(myChoice) Then

If myChoice = "t" Then

myWdsList = "5,"

jumpFwd = 200

ActiveDocument.Words(jumpFwd).Select

Selection.Collapse wdCollapseStart

Selection.TypeText Text:="this is pauljqjqs speed " \_

& "test rhubarb this is pauljqjqs speed test "

Else

myWdsList = myList(Asc(myChoice) - 96)

myWdsList = Replace(myWdsList, " ", "") & ","

myWdsList = Replace(myWdsList, ",,", ",")

End If

Else

myWdsList = myChoice & ","

End If

myRun = Split(myWdsList, ",")

numRuns = UBound(myRun) - 1

myOutput = ""

st0 = Timer

If goExtraFast = True Then Application.Visible = False

For j = 0 To numRuns

Selection.HomeKey Unit:=wdStory

myTask = myRun(j)

' Search for phrases

myPhrases = ""

shownOne = False

totWds = wordsDoc.Words.Count

phrLen = Val(myTask)

myMinWds = 2

bktPos = InStr(myTask, "(")

myTask = Mid(myTask, bktPos + 1)

If bktPos > 0 Then

myMinWds = Val(myTask)

ignoreSubPhrases = False

Else

ignoreSubPhrases = True

End If

tstPhrase = ""

For n = 1 To phrLen

tstPhrase = tstPhrase & "dummy "

Next n

i = 1

st = Timer

myPrompt = "No duplicate phrases found yet" \_

& " Wds: " & phrLen & "(" & myMinWds & ")"

displayPhrase = myPrompt

For Each wd In wordsDoc.Words

spPos = InStr(tstPhrase, " ")

tstPhrase = Mid(tstPhrase, spPos + 1) & wd.Text

i = i + 1

nw = Timer

pc = Str(Int(1000 \* i / totWds) / 10)

If InStr(pc, ".") = 0 Then pc = pc & ".0"

StatusBar = sps & pc & "% " & phrLen & "(" & \_

myMinWds & ")" & " ETA " & predictedTime

If InStr(myPhrases, tstPhrase & myDots) = 0 Then

DoEvents

Set rng = wordsDoc.Content

' Find the first one

tstLen = Len(tstPhrase)

If tstLen > 255 Then tstPhrase = Left(tstPhrase, 254)

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = tstPhrase

If ignoreSubPhrases = True Then .Font.Underline = False

.Replacement.Text = ""

.MatchWildcards = False

End With

phrFreq = -1

Do

phrFreq = phrFreq + 1

rng.Find.Execute

rng.Collapse wdCollapseEnd

Loop Until rng.Find.Found = False

If phrFreq > 1 Then

newPhrase = tstPhrase & myDots & Trim(Str(phrFreq))

displayPhrase = Replace(newPhrase, "cmcm", ",")

displayPhrase = Replace(displayPhrase, "cqcq", "-")

displayPhrase = Replace(displayPhrase, "jqjq", "'")

DoEvents

ActiveDocument.ActiveWindow.Caption = \_

sps & "LATEST FIND: " & displayPhrase & sps & sps

myPrompt = " Wds: " & phrLen & "(" & myMinWds & ")"

timeToGo = (nw - st) \* (totWds - i) / i

myTime = Time

myETA = DateAdd("s", timeToGo, myTime)

predictedTime = Left(myETA, 5)

If myChoice = "t" Then

MsgBox "ETA: " & predictedTime & " = " & \_

Int(timeToGo / 6) / 10 & " min"

wordsDoc.Close SaveChanges:=False

Exit Sub

End If

myPrompt = myPrompt & " ETA " & predictedTime

If i > totWds Then

ahfkjhasdkjgf = 0

End If

pc = Str(Int(1000 \* i / totWds) / 10)

If InStr(pc, ".") = 0 Then pc = pc & ".0"

spd = Str(Int(10 \* i / (nw - st)) / 10)

If InStr(spd, ".") = 0 Then spd = spd & ".0"

Debug.Print spd & " " & pc & "% " & myPrompt & \_

" " & displayPhrase

myPhrases = myPhrases & newPhrase & vbCr

If phrFreq > myMinWds - 1 Then

myOutput = myOutput & newPhrase & vbCr

If highlightFinds Then

oldColour = Options.DefaultHighlightColorIndex

Options.DefaultHighlightColorIndex = myColour

With rng0.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = tstPhrase

.Wrap = wdFindContinue

.Replacement.Text = ""

.Replacement.Highlight = True

.Execute Replace:=wdReplaceAll

End With

Options.DefaultHighlightColorIndex = oldColour

End If

End If

DoEvents

End If

End If

Next wd

myOutput = Replace(myOutput, "cqcq", "-")

myOutput = Replace(myOutput, "cmcm", ",")

myOutput = Replace(myOutput, "jqjq", ChrW(8217))

rng0.InsertAfter Text:=vbCr & "@@@@@@@@@@@@@@@@@@@@@ " & \_

ChrW(8211) & " " & phrLen & vbCr & myOutput & vbCr

myPhrases = ""

myOutput = ""

t = Timer - st

If t > 600 Then

ttot = Int(t / 6) / 10

tText = Str(ttot) & " min"

Else

ttot = Int(t \* 10) / 10

tText = Str(ttot) & " sec"

End If

myResult = "Ave wds/sec: " & Int(10 \* i / t) / 10 & vbCr & vbCr

myResult = myResult & "Time: " & tText

rng0.InsertAfter Text:=vbCr & "================== " & vbCr \_

& myResult & vbCr

If stopAndShowTime = True Then

Application.Visible = True

MsgBox myResult

End If

Next j

StatusBar = " "

t = Timer - st0

If t > 600 Then

ttot = Int(t / 6) / 10

tText = Str(ttot) & " min"

Else

ttot = Int(t \* 10) / 10

tText = Str(ttot) & " sec"

End If

myResult = "Ave wds/sec: " & Int(10 \* i \* (numRuns + 1) / t) / 10 \_

& vbCr & vbCr

myResult = myResult & "Total time: " & tText

rng0.InsertAfter Text:=vbCr & "================== " & vbCr \_

& "================== " & vbCr & myResult & vbCr

Application.Visible = True

Beep

ActiveDocument.ActiveWindow.Caption = ""

MsgBox myResult

wordsDoc.Close SaveChanges:=False

resultsDoc.Activate

Selection.HomeKey Unit:=wdStory

With Selection.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "@@@@@"

.Replacement.Text = ""

.MatchWildcards = False

.Execute

End With

Set rng = Selection.range

rng.End = ActiveDocument.Content.End

rng.HighlightColorIndex = wdNoHighlight

Selection.Collapse wdCollapseStart

Application.StatusBar = False

Beep

End Sub

## The ProperNounAlyse Macro

***Source: Paul Beverley,*** [***http://www.archivepub.co.uk/book.html***](http://www.archivepub.co.uk/book.html)

Sub ProperNounAlyse()

' Paul Beverley - Version 22.08.23

' Analyses similar proper nouns

minLengthCheck = 4

includeAcronyms = True

ignoreWords = "The This There Those Their They Then These That"

similarChars = "bb,b; b,p; sch,sh; ch,sh; c,k; ph,f; ss,z; s,z;" & \_

" mp,m; ll,l; nn,n; nd,n; nt,n;"

' With non-English languages, you might need to make this False

ignorePlurals = True

myScreenOff = True

Set rng = Selection.Range.Duplicate

rng.End = rng.Start + 1

myLanguage = Languages(rng.LanguageID).NameLocal

Set FUT = ActiveDocument

doingSeveralMacros = (InStr(FUT.Name, "zzTestFile") > 0)

If doingSeveralMacros = False Then

myResponse = MsgBox(" ProperNounAlyse" & vbCr & vbCr & \_

"Analyse this document?", vbQuestion \_

+ vbYesNoCancel, "ProperNounAlyse")

If myResponse <> vbYes Then Exit Sub

End If

If myScreenOff = True Then

Application.ScreenUpdating = False

On Error GoTo ReportIt

End If

myDummy = ChrW(222)

For i = 1 To 100

spcs = " " & spcs

Next i

dummyText = ChrW(197) & "zzzx "

For i = 65 To 90

dummyText = dummyText & ChrW(i) & "zzzz "

Next i

checkFinalLetters = True

' checkFinalLetters = False

' Grey on word only

thisHighlight = wdGray25

doMissingLetter = True

' doMissingLetter = False

' Bold And blue

switchTest = True

' switchTest = False

' double strikethrough

doSimilarLetters = True

' doSimilarLetters = False

' various highlight colours + underline

doVowelTest = True

' doVowelTest = False

' various highlight colours + italic

' These last two tests cycle through these colours:

maxCol = 6

ReDim myCol(maxCol) As Integer

myCol(1) = wdYellow

myCol(2) = wdBrightGreen

myCol(3) = wdTurquoise

myCol(4) = wdRed

myCol(5) = wdPink

myCol(6) = wdGray25

colcode = 0

oldColour = Options.DefaultHighlightColorIndex

Options.DefaultHighlightColorIndex = wdGray25

leadDots = " . . . "

title1 = "Proper noun list"

title2 = "Proper noun queries"

CR = vbCr: CR2 = CR & CR

convCharsUC = "AAAAAAA.EEEEIIII..OOOOO.OUUUU" & \_

"...aaaaaaa.eeeeiiiio.ooooo.ouuuu......"

convCharsLC = LCase(convCharsUC)

timeStart = Timer

' collect notes text, if any

endText = ""

footText = ""

If ActiveDocument.Endnotes.count > 0 Then

endText = ActiveDocument.StoryRanges(wdEndnotesStory).Text

End If

If ActiveDocument.Footnotes.count > 0 Then

footText = ActiveDocument.StoryRanges(wdFootnotesStory).Text

End If

' collect text in all the textboxes (if any)

sh = ActiveDocument.Shapes.count

If sh > 0 Then

ReDim shText(sh)

i = 0

For Each shp In ActiveDocument.Shapes

If shp.Type <> 24 And shp.Type <> 3 Then

If shp.TextFrame.HasText Then

i = i + 1

shText(i) = shp.TextFrame.TextRange.Text

End If

End If

Next

shCount = i

End If

' Create various documents

Set rng = ActiveDocument.Content

Documents.Add

Set firstDoc = ActiveDocument

Set fnl = ActiveDocument.Content

Documents.Add

Set tempDoc = ActiveDocument

Set tmp = ActiveDocument.Content

Documents.Add

Set allText = ActiveDocument

Selection.TypeText dummyText & vbCr

Selection.FormattedText = rng.FormattedText

Selection.Collapse wdCollapseEnd

' Add notes + shape text

Selection.TypeText endText & CR & footText & CR

If shCount > 0 Then

For i = 1 To shCount

Selection.TypeText shText(i) & CR

Next i

End If

Selection.HomeKey Unit:=wdStory

Set rng = allText.Content

rng.Revisions.AcceptAll

DoEvents

StatusBar = spcs & "Preparing copied file - 1"

DoEvents

' Delete struck-through text

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = ""

.MatchWildcards = False

.Font.StrikeThrough = True

.Replacement.Text = " "

.Execute Replace:=wdReplaceAll

End With

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "["

.MatchWildcards = False

.Replacement.Text = " "

.Execute Replace:=wdReplaceAll

End With

' Remove strange unicode characters

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "[" & ChrW(&HA000) & "-" & ChrW(&HD6FF) & "]{1,}"

.MatchWildcards = True

.Replacement.Text = " "

.Execute Replace:=wdReplaceAll

End With

DoEvents

StatusBar = spcs & "Preparing copied file - 2"

DoEvents

' Cut all and replace as pure text

Set rng = allText.Content

tmp.FormattedText = rng.FormattedText

rng.Text = tmp.Text

tmp.Delete

DoEvents

StatusBar = spcs & "Preparing copied file - 3"

' Use qqq for apostrophe

Set rng = allText.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "n" & ChrW(8217) & "t"

.MatchWildcards = False

.Replacement.Text = "nqqqt"

.Execute Replace:=wdReplaceAll

End With

' Use qq for apostrophe

With rng.Find

.Text = "O'"

.MatchCase = True

.Replacement.Text = "Oqqq"

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

' Find initial cap words

DoEvents

StatusBar = spcs & "Preparing copied file - 4"

DoEvents

myChopNum = minLengthCheck - 2

If myChop < 1 Then myChop = 1

myChop = Trim(Str(myChopNum))

myFind = "<[A-Z][a-z][a-zA-Z]{" & myChop & ",}"

If includeAcronyms = True Then myFind = \_

"<[A-Z][a-zA-Z][a-zA-Z]{" & myChop & ",}"

Set rng = allText.Range

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = myFind

.MatchWildcards = True

.MatchCase = True

.Replacement.Text = "^&"

.Replacement.Highlight = True

.Replacement.Font.StrikeThrough = True

.Execute Replace:=wdReplaceAll

End With

' Delete all non-strikethrough words

DoEvents

StatusBar = spcs & "Preparing copied file - 5"

DoEvents

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = ""

.Font.StrikeThrough = False

.MatchWildcards = False

.MatchCase = True

.Replacement.Text = "^p"

.Execute Replace:=wdReplaceAll

End With

' Delete the unwanted "proper nouns"

DoEvents

StatusBar = spcs & "Preparing copied file - 6"

igWords = Split(Trim(ignoreWords), " ")

For Each wd In igWords

Set rng = allText.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = wd & "^p"

.Wrap = wdFindContinue

.Replacement.Text = ""

.MatchCase = True

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

DoEvents

Next wd

StatusBar = spcs & "Sorting whole file"

DoEvents

i = 0

For ch = 65 To 90

For Each myPara In allText.Paragraphs

If Asc(myPara.Range) = ch Then

DoEvents

myPara.Range.Font.StrikeThrough = False

tmp.InsertAfter myPara.Range.Text

End If

Next myPara

tmp.InsertAfter Text:="Zzzzz" & CR

Set rng = tempDoc.Content

rng.Sort SortOrder:=wdSortOrderAscending, CaseSensitive:=True

' delete initial blank line

If Len(tempDoc.Paragraphs(1)) < 3 Then \_

tempDoc.Paragraphs(1).Range.Delete

' Create a frequency for each highlighted word

thisWord = ""

myCount = 0

For Each myPara In tempDoc.Paragraphs

Set rng = myPara.Range.Words(1)

DoEvents

nextWord = rng

If nextWord <> thisWord Then

' This is a new word

If Len(thisWord) > 1 Then

fnl.InsertAfter Text:=thisWord \_

& leadDots & Trim(Str(myCount)) & CR

End If

thisWord = nextWord

myCount = 1

Else

myCount = myCount + 1

End If

If nextWord = "Zzzzz" Then Exit For

i = i + 1:

If i Mod 400 = 4 Then

DoEvents

prmt = Left(thisWord, 1) & " "

prmt = prmt & prmt & prmt & prmt

StatusBar = spcs & \_

"Preparing words for frequency list - " & prmt

DoEvents

End If

Next myPara

' Remove all words except frequency counts

Set rng = tempDoc.Content

rng.Delete

Next ch

' Find any unaccounted-for words, e.g. Ångstrom

For Each myPara In allText.Paragraphs

If myPara.Range.Words(1).Font.StrikeThrough = True Then

tmp.InsertAfter myPara.Range.Text

End If

Next myPara

tempDoc.Close SaveChanges:=False

allText.Close SaveChanges:=False

firstDoc.Activate

' Remove blank lines

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "[^13]{2,}"

.Wrap = wdFindContinue

.Replacement.Text = "^p"

.Forward = True

.MatchWildcards = True

.Execute Replace:=wdReplaceAll

End With

' Resort case insensitively

Set rng = ActiveDocument.Content

rng.Sort SortOrder:=wdSortOrderAscending, \_

CaseSensitive:=False

' Delete rubbish from top and bottom of list

Do

Set rng = ActiveDocument.Paragraphs(1).Range

myLen = Len(rng.Text)

If myLen < 10 Then

rng.Select

Selection.Delete

End If

Loop Until myLen > 9

Do

lastLine = ActiveDocument.Paragraphs.count

Set rng = ActiveDocument.Paragraphs(lastLine).Range

myLen = Len(rng.Text)

If myLen < 10 Then

rng.Select

Selection.Delete

End If

Loop Until Len(rng.Text) >= 2

' Word list now has freq. count.

Do

lastLine = ActiveDocument.Paragraphs.count

Set rng = ActiveDocument.Paragraphs(lastLine).Range

myLen = Len(rng.Text)

If myLen < 10 Then

rng.Select

Selection.Delete

End If

Loop Until Len(rng.Text) >= 2

' Create another copy for doing extra tests

Set rng = ActiveDocument.Content

Documents.Add

Set finalList = ActiveDocument

finalList.Range.Text = rng.Text

Selection.HomeKey Unit:=wdStory

' Prepare data for other tests

numWords = ActiveDocument.Paragraphs.count

For i = 1 To numWords

aWord = ActiveDocument.Paragraphs(i).Range.Words(1)

n = AscW(aWord)

thisChar = ChrW(n)

If n > 129 Then

If n >= 217 Then aWord = Replace(aWord, thisChar, "U")

If n >= 210 Then aWord = Replace(aWord, thisChar, "O")

If n >= 204 Then aWord = Replace(aWord, thisChar, "I")

If n >= 200 Then aWord = Replace(aWord, thisChar, "E")

If n >= 192 Then aWord = Replace(aWord, thisChar, "A")

End If

allWords = allWords & aWord

jmp = 100

If i Mod jmp = 1 Then

pq = pq + 1

DoEvents

StatusBar = spcs & \_

"Preparing data for other tests - 1 - " & pq

DoEvents

End If

Next i

' ...for the vowel test below

DoEvents

StatusBar = spcs & "Preparing data for other tests - 2"

DoEvents

noVowelWords = " " & allWords

noVowelWords = Replace(noVowelWords, " A", "\_1")

noVowelWords = Replace(noVowelWords, " E", "\_2")

noVowelWords = Replace(noVowelWords, " I", "\_3")

noVowelWords = Replace(noVowelWords, " O", "\_4")

noVowelWords = Replace(noVowelWords, " U", "\_5")

noVowelWords = Replace(noVowelWords, " Y", "\_6")

For k = 2 To Len(noVowelWords) - 1

thisChar = Mid(noVowelWords, k, 1)

n = AscW(thisChar)

If n > 191 And n < 221 Then

myNewChar = Mid(convCharsLC, n - 191, 1)

If myNewChar <> "." Then noVowelWords = \_

Replace(noVowelWords, thisChar, myNewChar)

End If

Next k

noVowelWords = Replace(noVowelWords, "a", "")

noVowelWords = Replace(noVowelWords, "e", "")

noVowelWords = Replace(noVowelWords, "i", "")

noVowelWords = Replace(noVowelWords, "o", "")

noVowelWords = Replace(noVowelWords, "u", "")

noVowelWords = Replace(noVowelWords, "y", "")

noVowelWords = Replace(noVowelWords, "A", "")

noVowelWords = Replace(noVowelWords, "E", "")

noVowelWords = Replace(noVowelWords, "I", "")

noVowelWords = Replace(noVowelWords, "O", "")

noVowelWords = Replace(noVowelWords, "U", "")

noVowelWords = Replace(noVowelWords, "Y", "")

noVowelWords = Replace(noVowelWords, "\_1", " A")

noVowelWords = Replace(noVowelWords, "\_2", " E")

noVowelWords = Replace(noVowelWords, "\_3", " I")

noVowelWords = Replace(noVowelWords, "\_4", " O")

noVowelWords = Replace(noVowelWords, "\_5", " U")

noVowelWords = Replace(noVowelWords, "\_6", " Y")

' ...for the similar words test

DoEvents

StatusBar = spcs & "Preparing data for other tests - 3"

DoEvents

similarAllWords = " " & LCase(allWords)

similarChars = Replace(similarChars, " ", "")

sChars = Replace(similarChars, " ", "")

Do

commaPos = InStr(sChars, ",")

charWas = Left(sChars, commaPos - 1)

sChars = Mid(sChars, commaPos + 1)

semicolonPos = InStr(sChars, ";")

charNew = Left(sChars, semicolonPos - 1)

sChars = Mid(sChars, semicolonPos + 1)

similarAllWords = Replace(similarAllWords, charWas, charNew)

Loop Until Len(sChars) < 2

' Changes all the accented characters to non-accented

DoEvents

StatusBar = spcs & "Preparing data for other tests - 4"

DoEvents

sWd = similarAllWords

For k = 1 To Len(sWd) - 1

thisChar = Mid(sWd, k, 1)

n = AscW(thisChar)

myNewChar = "."

If n > 191 And n < 256 Then

myNewChar = Mid(convCharsLC, n - 191, 1)

If myNewChar <> "." Then sWd = Replace(sWd, \_

thisChar, myNewChar)

End If

Next k

similarAllWords = sWd

' Catch words with only the final two letters the same

i = 0

If checkFinalLetters = True Then

For Each myPara In ActiveDocument.Paragraphs

gotOne = False

myWord = Trim(myPara.Range.Words(1))

myLen = Len(myWord)

If myLen > 6 Then

myTarget = "^p" & Left(myWord, myLen - 2) & "^$^$ "

myCut = 2

Else

myTarget = "^p" & Left(myWord, myLen - 1) & "^$ "

myCut = 1

End If

Set rng = ActiveDocument.Content

rng.Start = myPara.Range.End - 3

rng.Collapse wdCollapseStart

With rng.Find

.Replacement.ClearFormatting

.ClearFormatting

.Text = myTarget

.Replacement.Text = ""

.Forward = True

.MatchCase = True

.MatchWildcards = False

.Wrap = wdFindStop

End With

rng.Find.Execute

Do While rng.Find.Found

gotOne = True

rng.MoveStart 1

rng.End = rng.Start + myLen - myCut

rng.HighlightColorIndex = thisHighlight

rng.Font.Bold = True

rng.Find.Execute

Loop

If gotOne = True Then

Set rng = myPara.Range.Words(1)

rng.End = rng.Start + myLen - myCut

rng.HighlightColorIndex = thisHighlight

rng.Font.Bold = True

End If

i = i + 1

If i Mod 100 = 1 Then

DoEvents

StatusBar = spcs & "Doing test (5) on " & myWord

DoEvents

End If

Next myPara

End If

If doMissingLetter = True Then

' Start of test

doneWords = ""

doneSimilarWords = ""

McList = ""

For i = 1 To ActiveDocument.Paragraphs.count - 1

myWord = ActiveDocument.Paragraphs(i).Range.Words(1)

n = AscW(myWord)

thisChar = ChrW(n)

myNewChar = "."

' Changes the capital letter, if a vowel

If n > 191 And n < 221 Then

myNewChar = Mid(convCharsUC, n - 191, 1)

If myNewChar <> "." Then myWord = Replace(myWord, \_

thisChar, myNewChar)

End If

If i Mod 50 = 1 Then

DoEvents

StatusBar = spcs & "Other tests (4) on " & myWord

DoEvents

End If

testWords = Replace(allWords, myWord, "")

captestLetters = Left(myWord, 1)

' Check if word reappears with one letter missing (1)

For k = 2 To Len(myWord) - 1

testWord = " " & Left(myWord, k - 1) & Mid(myWord, k + 1)

wordPos = InStr(allWords, testWord)

If wordPos > 0 Then

lastLetter = Mid(myWord, Len(myWord) - 1, 1)

' but not "s" at the end, unless it's a spelling error

If lastLetter = "s" Then

ignoreIt = (Application.CheckSpelling(myWord, \_

MainDictionary:=myLanguage) = True)

Else

ignoreIt = False

End If

If ignoreIt = False And ignorePlurals = True Then

colcode = (colcode + 1) Mod maxCol

thisCol = myCol(colcode + 1)

' mark the pair

leftBit = Left(allWords, InStr(allWords, testWord) \_

+ Len(testWord) - 1)

j = Len(leftBit) - Len(Replace(leftBit, " ", ""))

Set rng = ActiveDocument.Paragraphs(i).Range

rng.HighlightColorIndex = thisCol

rng.Font.Bold = True

rng.Font.Color = wdColorBlue

Set rng = ActiveDocument.Paragraphs(j).Range

rng.HighlightColorIndex = thisCol

rng.Font.Bold = True

rng.Font.Color = wdColorBlue

End If

End If

Next k

If Left(myWord, 2) = "Mc" Or Left(myWord, 3) = "Mac" Or \_

Left(myWord, 3) = "Mag" Then

McList = McList & ActiveDocument.Paragraphs(i).Range

End If

Next i

End If

If doSimilarLetters = True Then

doneWords = ""

doneSimilarWords = ""

For i = 1 To ActiveDocument.Paragraphs.count - 1

myWord = ActiveDocument.Paragraphs(i).Range.Words(1)

n = AscW(myWord)

thisChar = ChrW(n)

myNewChar = "."

' Changes the capital letter, if a vowel

If n > 191 And n < 221 Then

myNewChar = Mid(convCharsUC, n - 191, 1)

If myNewChar <> "." Then myWord = Replace(myWord, \_

thisChar, myNewChar)

End If

If i Mod 50 = 1 Then

DoEvents

StatusBar = spcs & "Other tests (3) on " & myWord

DoEvents

End If

testWords = Replace(allWords, myWord, "")

captestLetters = Left(myWord, 1)

' check similar spellings: Perutz/Peruts or Chebyshev/Chevychev

similarWord = " " & LCase(myWord)

sChars = similarChars

Do

commaPos = InStr(sChars, ",")

charWas = Left(sChars, commaPos - 1)

sChars = Mid(sChars, commaPos + 1)

semicolonPos = InStr(sChars, ";")

charNew = Left(sChars, semicolonPos - 1)

sChars = Mid(sChars, semicolonPos + 1)

similarWord = Replace(similarWord, charWas, charNew)

Loop Until Len(sChars) < 2

' Changes all the accented characters to non-accented

For k = 1 To Len(myWord) - 1

thisChar = Mid(myWord, k, 1)

n = AscW(thisChar)

If n > 191 And n < 256 Then

myNewChar = Mid(convCharsUC, n - 191, 1)

If myNewChar <> "." Then myWord = Replace(myWord, \_

thisChar, myNewChar)

End If

Next k

similarAllWords = Mid(similarAllWords, Len(similarWord))

theseWords = similarAllWords

If InStr(doneSimilarWords, similarWord) = 0 And \_

InStr(theseWords, similarWord) > 0 Then

colcode = (colcode + 1) Mod maxCol

thisCol = myCol(colcode + 1)

Set rng = ActiveDocument.Paragraphs(i).Range

rng.HighlightColorIndex = thisCol

rng.Font.Underline = True

doneSimilarWords = doneSimilarWords & similarWord

' search through all the following words

theseWords = similarAllWords

For j = 1 To numWords - i

spPos = InStr(Trim(theseWords) & " ", " ")

If Left(theseWords, spPos + 1) = similarWord Then

Set rng = ActiveDocument.Paragraphs(i + j).Range

rng.HighlightColorIndex = thisCol

rng.Font.Underline = True

End If

theseWords = Mid(theseWords, spPos + 1)

capThisLetter = Mid(theseWords, 2, 1)

If capThisLetter <> LCase(captestLetters) Then Exit For

Next j

End If

Next i

End If

If switchTest = True Then

doneWords = ""

doneSimilarWords = ""

McList = ""

For i = 1 To ActiveDocument.Paragraphs.count - 1

myWord = ActiveDocument.Paragraphs(i).Range.Words(1)

n = AscW(myWord)

thisChar = ChrW(n)

myNewChar = "."

' Changes the capital letter, if a vowel

If n > 191 And n < 221 Then

myNewChar = Mid(convCharsUC, n - 191, 1)

If myNewChar <> "." Then myWord = Replace(myWord, \_

thisChar, myNewChar)

End If

If i Mod 50 = 1 Then

DoEvents

StatusBar = spcs & "Other tests (2) on " & myWord

DoEvents

End If

testWords = Replace(allWords, myWord, "")

captestLetters = Left(myWord, 1)

' check for switched chars

wordLen = Len(myWord) - 1

For k = 1 To Len(myWord) - 3

otherWord = Left(myWord, k) & Mid(myWord, k + 2, 1) & \_

Mid(myWord, k + 1, 1) & Mid(myWord, k + 3)

wordPos = InStr(testWords, otherWord)

If wordPos > 0 Then

' Find the position of the matching word

matchWord = Mid(testWords, wordPos, Len(myWord))

leftBit = Left(allWords, InStr(allWords, matchWord) + 1)

j = Len(leftBit) - Len(Replace(leftBit, " ", "")) + 1

ActiveDocument.Paragraphs(i).Range.Font.DoubleStrikeThrough \_

= True

ActiveDocument.Paragraphs(i).Range.HighlightColorIndex \_

= thisCol

ActiveDocument.Paragraphs(j).Range.Font.DoubleStrikeThrough \_

= True

ActiveDocument.Paragraphs(j).Range.HighlightColorIndex \_

= thisCol

End If

Next k

Next i

End If

If doVowelTest = True Then

doneWords = ""

doneSimilarWords = ""

McList = ""

For i = 1 To ActiveDocument.Paragraphs.count - 1

myWord = ActiveDocument.Paragraphs(i).Range.Words(1)

n = AscW(myWord)

thisChar = ChrW(n)

myNewChar = "."

' Changes the capital letter, if a vowel

If n > 191 And n < 221 Then

myNewChar = Mid(convCharsUC, n - 191, 1)

If myNewChar <> "." Then myWord = Replace(myWord, \_

thisChar, myNewChar)

End If

If i Mod 50 = 1 Then

DoEvents

StatusBar = spcs & "Other tests (1) on " & myWord

DoEvents

End If

testWords = Replace(allWords, myWord, "")

captestLetters = Left(myWord, 1)

' check if there's a word with different vowels

otherWord = " " & Replace(myWord, "a", "")

otherWord = Replace(otherWord, "e", "")

otherWord = Replace(otherWord, "i", "")

otherWord = Replace(otherWord, "o", "")

otherWord = Replace(otherWord, "u", "")

otherWord = Replace(otherWord, "y", "")

' Delete all the accented characters

For k = 3 To Len(otherWord) - 1

thisChar = Mid(otherWord, k, 1)

n = AscW(thisChar)

If InStr("AEIOUY", thisChar) > 0 Then

otherWord = Left(otherWord, k - 1) & "=" & Mid(otherWord, k + 1)

Else

If n > 191 And n < 221 Then

myNewChar = Mid(convCharsUC, n - 191, 1)

If myNewChar <> "." Then

otherWord = Replace(otherWord, thisChar, "=")

End If

End If

End If

Next k

otherWord = Replace(otherWord, "=", "")

' otherWord is now the word under test (vowel-less)

otherWord = Replace(otherWord, ".", "")

noVowelWords = Mid(noVowelWords, Len(otherWord))

If Left(noVowelWords, 1) <> " " Then noVowelWords = \_

" " & noVowelWords

theseWords = noVowelWords

wordPos = InStr(noVowelWords, otherWord)

If InStr(doneWords, otherWord) = 0 And wordPos > 0 Then

colcode = (colcode + 1) Mod maxCol

thisCol = myCol(colcode + 1)

Set rng = ActiveDocument.Paragraphs(i).Range

rng.HighlightColorIndex = thisCol

rng.Font.Italic = True

doneWords = doneWords & otherWord

For j = 1 To numWords - i

spPos = InStr(Trim(theseWords) & " ", " ")

firstWord = Left(theseWords, spPos + 1)

theseWords = Mid(theseWords, spPos + 1)

If firstWord = otherWord Then

Set rng = ActiveDocument.Paragraphs(i + j).Range

rng.HighlightColorIndex = thisCol

rng.Font.Italic = True

End If

capThisLetter = Mid(theseWords, 2, 1)

If capThisLetter > "" And capThisLetter <> \_

captestLetters Then Exit For

Next j

End If

Next i

End If

finishOff:

Selection.EndKey Unit:=wdStory

Selection.TypeText CR2 & McList

Selection.HomeKey Unit:=wdStory

Selection.TypeText title1 & CR

Do

Selection.Expand wdParagraph

If Len(Selection) < 3 Or LCase(Selection) = \_

UCase(Selection) Then Selection.Delete

Loop Until LCase(Selection) <> UCase(Selection)

Selection.HomeKey Unit:=wdStory, Extend:=wdExtend

Selection.Style = ActiveDocument.Styles(wdStyleHeading1)

' Restore apostrophes

Set rng = finalList.Range

With rng.Find

.Text = "qqq"

.MatchCase = False

.Replacement.Text = "'"

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

' Find first highlight

Set rng = finalList.Content

With rng.Find

.Text = "Zzzzz"

.Wrap = wdFindStop

.Replacement.Text = ""

.Forward = True

.MatchWildcards = False

.Execute Replace:=wdReplaceOne

End With

Set rng = finalList.Content

With rng.Find

.Text = ""

.Highlight = True

.Wrap = wdFindStop

.Replacement.Text = ""

.Forward = True

.MatchWildcards = False

.Execute

End With

rng.Select

Selection.Collapse wdCollapseStart

Set finalList = ActiveDocument

firstDoc.Activate

' Find sets of sounds-like words

StatusBar = spcs & "Sounds-like tests"

k = 0

For Each myPara In ActiveDocument.Paragraphs

myWord = Trim(myPara.Range.Words(1))

k = k + 1

If k Mod 40 = 1 Then

DoEvents

StatusBar = spcs & "Sounds-like test: " & myWord

DoEvents

End If

hasAccent = False

For i = 1 To Len(myWord)

ascChar = AscW(Mid(myWord, i))

If ascChar > 128 Or ascChar = Asc("?") Then hasAccent = True

Next i

' Go and find the first sounds-like word

initLetter = Left(myWord, 1)

If Len(myWord) > 2 And myPara.Range.HighlightColorIndex > 0 And \_

hasAccent = False And InStr(allSets, myWord & leadDots) \_

= 0 Then

Set rng = ActiveDocument.Content

Do

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = myWord

.Wrap = wdFindStop

.Replacement.Text = ""

.MatchWildcards = False

.MatchSoundsLike = True

.Execute

End With

Set myPara = rng.Paragraphs(1).Range

rng.Collapse wdCollapseEnd

Loop Until Left(myPara, 1) = initLetter

setOfWords = myPara

gottaSet = False

rng.Collapse wdCollapseEnd

rng.Find.Execute

Do While rng.Find.Found = True

Set myPara = rng.Paragraphs(1).Range

If Left(myPara, 1) = initLetter Then

gottaSet = True

setOfWords = setOfWords & myPara

End If

rng.Collapse wdCollapseEnd

rng.Find.Execute

Loop

If gottaSet = True Then allSets = allSets & setOfWords & CR

End If

Next myPara

Selection.WholeStory

If Len(allSets) < 2 Then

Selection.TypeText "None found with this test"

Else

Selection.TypeText allSets

End If

Selection.HomeKey Unit:=wdStory

Selection.TypeText "Proper nouns by sound" & CR

Selection.HomeKey Unit:=wdStory, Extend:=wdExtend

Selection.Style = ActiveDocument.Styles(wdStyleHeading1)

Selection.HomeKey Unit:=wdStory

Set rng = ActiveDocument.Content

rng.HighlightColorIndex = 0

rng.Copy

ActiveDocument.Close SaveChanges:=False

finalList.Activate

' Remove highlighting from second half of words

' that are only case changes of one another

totParas = ActiveDocument.Paragraphs.count

For i = 1 To totParas - 1

A = Trim(ActiveDocument.Paragraphs(i).Range.Words(1))

b = Trim(ActiveDocument.Paragraphs(i + 1).Range.Words(1))

A = Mid(A, 2)

b = Mid(b, 2)

If LCase(A) = LCase(b) And Len(A) > 2 Then

If (UCase(A) = A And LCase(b) = b) Or (UCase(b) = b And \_

LCase(A) = A) Then

ActiveDocument.Paragraphs(i).Range.Words(1).HighlightColorIndex = 0

ActiveDocument.Paragraphs(i + 1).Range.Words(1).HighlightColorIndex \_

= 0

End If

End If

If i Mod 50 = 0 Then

DoEvents

StatusBar = spcs & "Final checks: " & totParas - i

DoEvents

End If

Next i

myOnames = ""

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "^13O[!a-z]"

.Wrap = wdFindStop

.Replacement.Text = ""

.Forward = True

.MatchSoundsLike = False

.MatchWildcards = True

.Execute

End With

Do While rng.Find.Found = True

rng.Collapse wdCollapseEnd

rng.Expand wdWord

wd = Mid(rng.Text, 3)

rng.Expand wdParagraph

pa = rng.Text

Set rng2 = ActiveDocument.Content

With rng2.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "^13" & wd

.Wrap = wdFindStop

.Replacement.Text = ""

.Forward = True

.MatchWildcards = True

.Execute

End With

If rng2.Find.Found Then

rng2.Collapse wdCollapseEnd

rng2.Expand wdParagraph

pa2 = rng2.Text

myOnames = myOnames & pa2 & pa & vbCr

End If

rng.Collapse wdCollapseEnd

rng.End = rng.End - 2

rng.Find.Execute

Loop

If myOnames > "" Then

Selection.EndKey Unit:=wdStory

Selection.TypeText "Possible O'<something> errors" & vbCr

Selection.MoveUp , 1

Selection.Style = ActiveDocument.Styles(wdStyleHeading1)

Selection.EndKey Unit:=wdStory

Selection.TypeText myOnames

Selection.HomeKey Unit:=wdStory

End If

Set rng = ActiveDocument.Content

finalList.Activate

Selection.EndKey Unit:=wdStory

Selection.TypeText vbCr & vbCr & vbCr

Selection.Paste

Selection.HomeKey Unit:=wdStory

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = myDummy

.Wrap = wdFindContinue

.Replacement.Text = " "

.Forward = True

.MatchCase = False

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "^$zzz^$" & leadDots & "1" & vbCr

.Wrap = wdFindContinue

.Replacement.Text = ""

.Forward = True

.MatchCase = False

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

' Clear clipboard

Set rng = ActiveDocument.Content

rng.End = 2

rng.Copy

Set finalList = ActiveDocument

StatusBar = "Creating queries list"

Set rng = ActiveDocument.Content

Documents.Add

Selection.FormattedText = rng.FormattedText

Set queriesDoc = ActiveDocument

ActiveDocument.Paragraphs(1).Range.Delete

Set rng = ActiveDocument.Content

rng.Font.StrikeThrough = True

For Each myPara In ActiveDocument.Paragraphs

Set ch = myPara.Range.Characters(1)

chCol = ch.HighlightColorIndex

If chCol > 0 Then

myPara.Range.Font.StrikeThrough = False

End If

myLen = Len(myPara.Range.Text)

If myLen > 4 Then

If chCol > 0 Then

myPara.Range.Font.StrikeThrough = False

End If

Set che = myPara.Range.Characters(myLen - 2)

If che.HighlightColorIndex > 0 Then

myPara.Range.Font.StrikeThrough = False

End If

End If

Next myPara

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = ""

.Font.StrikeThrough = True

.Wrap = wdFindContinue

.Replacement.Text = "^p"

.Forward = True

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

DoEvents

End With

Set rng = ActiveDocument.Content

rng.Font.StrikeThrough = False

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "[^13]{3,}"

.Wrap = wdFindContinue

.Replacement.Text = "^p^p"

.Forward = True

.MatchWildcards = True

.Execute Replace:=wdReplaceAll

DoEvents

End With

For Each myPara In ActiveDocument.Paragraphs

myText = myPara.Range.Text

If Len(myText) > 4 Then

Set ch = myPara.Range.Characters(1)

numChars = myPara.Range.Characters.count

Set myEnd = myPara.Range.Characters(numChars)

colNum = ch.HighlightColorIndex Mod 8

If ch.Font.Bold = True Then

myTxt = "qcqc " & Str(colNum + 1) & " = zczc"

Else

myTxt = "qcqc zczc"

End If

If ch.Font.Underline > 0 And colNum > 0 Then

myBit = "\* "

myTxt = Replace(myTxt, " = ", "")

Else

myBit = ""

End If

myPara.Range.InsertBefore myBit & myTxt

If ch.Font.Italic = True Then

myEnd.InsertBefore "qpqp= " & Chr(65 + colNum)

End If

End If

i = i + 1

If i Mod 20 = 0 And Len(myText) > 4 Then

myText = Replace(myText, vbCr, "")

StatusBar = spcs & "Creating queries list: " & myText

End If

DoEvents

Next myPara

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "\\* qcqc(\*)zczc"

.Wrap = wdFindContinue

.Replacement.Text = "\* \1^t"

.Replacement.Highlight = False

.MatchWildcards = True

.Execute Replace:=wdReplaceAll

DoEvents

End With

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "qcqc(\*)zczc"

.Wrap = wdFindContinue

.Replacement.Text = "\1^t"

.Replacement.Highlight = False

.MatchWildcards = True

.Execute Replace:=wdReplaceAll

DoEvents

End With

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "qpqp(\*)^13"

.Replacement.Text = "^t\1^p"

.Replacement.Highlight = False

.MatchWildcards = True

.Execute Replace:=wdReplaceAll

DoEvents

End With

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "= ^$"

.Replacement.Text = ""

.Replacement.Font.Bold = False

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

DoEvents

End With

Set rng = ActiveDocument.Content

rng.Font.Bold = False

rng.Font.Italic = False

rng.Font.DoubleStrikeThrough = False

rng.Font.Underline = False

rng.Font.Color = wdColorBlack

Selection.HomeKey Unit:=wdStory

Selection.TypeText title2 & CR

Set rng = ActiveDocument.Content.Paragraphs(2).Range

If rng.Text = vbCr Then rng.Delete

Set rng = ActiveDocument.Content.Paragraphs(1).Range

rng.Style = ActiveDocument.Styles(wdStyleHeading1)

StatusBar = " "

Options.DefaultHighlightColorIndex = oldColour

lighterColour = wdGray25

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "= ^$"

.Replacement.Text = ""

.Replacement.Font.ColorIndex = lighterColour

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

DoEvents

.Text = "^# ="

.Replacement.Text = ""

.Execute Replace:=wdReplaceAll

DoEvents

End With

Application.ScreenUpdating = True

If doingSeveralMacros = False Then

Debug.Print Timer - timeStart

myTime = (Int(10 \* (Timer - timeStart) / 60) / 10)

Beep

If myTime > 0 Then MsgBox myTime & " minutes"

Else

FUT.Activate

End If

Exit Sub

ReportIt:

Application.ScreenUpdating = True

On Error GoTo 0

Resume

End Sub

## The ChronologyChecker Macro

***Source: Paul Beverley,*** [***http://www.archivepub.co.uk/book.html***](http://www.archivepub.co.uk/book.html)

Sub ChronologyChecker()

' Paul Beverley - Version 15.04.22

' Copies paragraphs containing date references into a new file

' Case sensitive

myColour\_1 = wdYellow

myWords\_1 = "Monday, Tuesday, Wednesday, Thursday, Friday,"

myWords\_1 = myWords\_1 & "Saturday, Sunday,"

myColour\_2 = wdBrightGreen

myWords\_2 = "January, February, April, June, July, August,"

myWords\_2 = myWords\_2 & "September, October, November, December"

' Case insensitive

myColour\_3 = wdYellow

myWords\_3 = "years old, tomorrow, next day, morning, evening, week, month"

' Case insensitive + whole word

myColour\_4 = wdYellow

myWords\_4 = "age, aged"

' Case sensitive AND whole word

myColour\_5 = wdBrightGreen

myWords\_5 = "May, March, Mon, Tue, Tues, Wed, Weds, Thu, Thurs, Fri, Sat, Sun"

' For years

myColour\_6 = wdTurquoise

multiSpace = 4

myWords\_1 = Replace(myWords\_1, " ", "")

myWords\_1 = Replace("," & myWords\_1 & ",", ",,", ",")

myWords\_2 = Replace(myWords\_2, " ", "")

myWords\_2 = Replace("," & myWords\_2 & ",", ",,", ",")

myWords\_3 = Replace(myWords\_3, " ", "")

myWords\_3 = Replace("," & myWords\_3 & ",", ",,", ",")

myWords\_4 = Replace(myWords\_4, " ", "")

myWords\_4 = Replace("," & myWords\_4 & ",", ",,", ",")

myWords\_5 = Replace(myWords\_5, " ", "")

myWords\_5 = Replace("," & myWords\_5 & ",", ",,", ",")

allWords = Replace(myWords\_1 & myWords\_2 & myWords\_3 & myWords\_4 \_

& myWords\_5, ",,", ",")

For i = 1 To multiSpace

SP = SP & vbCr

Next i

Set rng = ActiveDocument.Content

Documents.Add

For Each myPar In rng.Paragraphs

copyIt = False

For Each wd In myPar.Range.Words

DoEvents

mywd = Trim(wd.Text)

myTest = "," & LCase(mywd) & ","

If InStr(LCase(allWords), myTest) > 0 Then

copyIt = True

Exit For

End If

If Len(mywd) = 4 And LCase(mywd) = UCase(mywd) Then

' Is the first character 1 or 2?

isYear = (InStr("12", Left(mywd, 1)) > 0)

' Are the other three characters digits 0-9?

For i = 2 To 4

j = Asc(Mid(mywd, i)) - 48

If j < 0 Or j > 9 Then isYear = False

Next i

If isYear = True Then

copyIt = True

Exit For

End If

End If

DoEvents

Next wd

If copyIt Then

myPar.Range.Copy

Selection.Paste

Selection.Collapse wdCollapseEnd

Selection.TypeText SP

DoEvents

End If

Next myPar

Selection.HomeKey Unit:=wdStory

Selection.TypeText "Dates context" & vbCr & vbCr

ActiveDocument.Paragraphs(1).Range.Style = wdStyleHeading2

Selection.MoveLeft , 2

oldColour = Options.DefaultHighlightColorIndex

Options.DefaultHighlightColorIndex = myColour\_1

ActiveDocument.Content.HighlightColorIndex = wdNoHighlight

mywd = Split(myWords\_1, ",")

For i = 1 To UBound(mywd) - 1

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = mywd(i)

.Replacement.Text = ""

.Replacement.Highlight = True

.MatchCase = True

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

DoEvents

Next i

mywd = Split(myWords\_2, ",")

Options.DefaultHighlightColorIndex = myColour\_2

For i = 1 To UBound(mywd) - 1

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = mywd(i)

.Replacement.Text = ""

.Replacement.Highlight = True

.MatchCase = True

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

DoEvents

Next i

mywd = Split(myWords\_3, ",")

Options.DefaultHighlightColorIndex = myColour\_3

For i = 1 To UBound(mywd) - 1

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = mywd(i)

.Replacement.Text = ""

.Replacement.Highlight = True

.MatchCase = False

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

DoEvents

Next i

mywd = Split(myWords\_4, ",")

Options.DefaultHighlightColorIndex = myColour\_4

For i = 1 To UBound(mywd) - 1

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = mywd(i)

.Replacement.Text = ""

.Replacement.Highlight = True

.MatchCase = False

.MatchWholeWord = True

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

DoEvents

Next i

mywd = Split(myWords\_5, ",")

Options.DefaultHighlightColorIndex = myColour\_5

For i = 1 To UBound(mywd) - 1

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = mywd(i)

.Replacement.Text = ""

.Replacement.Highlight = True

.MatchCase = True

.MatchWholeWord = True

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

DoEvents

Next i

Options.DefaultHighlightColorIndex = myColour\_6

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "<[12][0-9]{3}>"

.Replacement.Text = ""

.Replacement.Highlight = True

.MatchCase = True

.MatchWholeWord = False

.MatchWildcards = True

.Execute Replace:=wdReplaceAll

End With

Options.DefaultHighlightColorIndex = oldColour

For i = ActiveDocument.Paragraphs.Count To 2 Step -1

Set myPar = ActiveDocument.Paragraphs(i).Range

If Len(myPar.Text) > 1 And myPar.HighlightColorIndex = wdNoHighlight Then

myPar.Select

Selection.MoveEnd , multiSpace

Selection.Delete

End If

DoEvents

Next i

Beep

End Sub

## The WordsPhrasesInContext Macro

***Source: Paul Beverley,*** [***http://www.archivepub.co.uk/book.html***](http://www.archivepub.co.uk/book.html)

Sub WordsPhrasesInContext()

' Paul Beverley - Version 19.02.18

' Copies paragraphs containing specific names into a new file

myListName = "zzSwitchList"

' myListName = "zzFReditList"

' findWords = "Brown | Jones | Green"

findWords = ""

myBasicColour = wdBrightGreen

returnToText = False

maxWds = 10

CaseSensitive = True

multiSpace = 4

CR = vbCr

For i = 1 To multiSpace

sp = sp & vbCr

Next i

myWords = ""

Set rng = ActiveDocument.Range(Selection.Start, ActiveDocument.Content.End)

parasToEnd = rng.Paragraphs.Count

totParas = ActiveDocument.Paragraphs.Count

If (totParas / parasToEnd) > 10 Then

Selection.Expand wdParagraph

myResponse = MsgBox("Start with this line?", vbQuestion \_

+ vbYesNoCancel, "WordsPhrasesInContext")

If myResponse = vbCancel Then Exit Sub

If myResponse = vbYes Then

Set rng = Selection.Range.Duplicate

rng.Collapse wdCollapseStart

rng.End = ActiveDocument.Content.End

myWords = rng.Text

myWords = Replace(CR & myWords & CR, CR & CR, CR)

myWords = Replace(myWords, CR & CR, CR)

myWords = Replace(myWords, CR & CR, CR)

rng.Collapse wdCollapseStart

rng.Expand wdParagraph

myWd = Split(myWords, CR)

totWords = UBound(myWd) - 1

ReDim myCol(totWords) As Integer

If rng.HighlightColorIndex = wdNoHighlight Then

For i = 1 To totWords

myCol(i) = myBasicColour

Next i

Else

For i = 1 To totWords

myCol(i) = rng.HighlightColorIndex

rng.Collapse wdCollapseEnd

rng.Expand wdParagraph

Next i

End If

Else

Selection.HomeKey Unit:=wdStory

End If

End If

Set mainDoc = ActiveDocument

If myWords = "" Then

gotExternalList = False

' Does a FRedit/Switch list have a "Context words:" line?

For Each myWnd In Application.Windows

thisName = myWnd.Document.Name

If InStr(thisName, myListName) > 0 Then

myWnd.Document.Activate

gotExternalList = True

Exit For

End If

Next myWnd

If gotExternalList = False Then mainDoc.Activate

' If so, load the word and colour arrays from it

DoEvents

Set rng = ActiveDocument.Content

DoEvents

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "Context words:"

.Replacement.Text = ""

.MatchCase = True

.MatchWildcards = False

.Execute

DoEvents

End With

If rng.Find.Found = False Then

mainDoc.Activate

DoEvents

Set rng = ActiveDocument.Content

DoEvents

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "Context words:"

.Replacement.Text = ""

.MatchCase = True

.MatchWildcards = False

.Execute

End With

DoEvents

End If

If rng.Find.Found Then

rng.Expand wdParagraph

rng.Collapse wdCollapseEnd

rng.End = ActiveDocument.Content.End

myWords = rng.Text

myWords = Replace(CR & myWords & CR, CR & CR, CR)

myWords = Replace(myWords, CR & CR, CR)

myWords = Replace(myWords, CR & CR, CR)

rng.Collapse wdCollapseStart

rng.Expand wdParagraph

myWd = Split(myWords, CR)

totWords = UBound(myWd) - 1

ReDim myCol(totWords) As Integer

For i = 1 To totWords

myCol(i) = rng.HighlightColorIndex

rng.Collapse wdCollapseEnd

rng.Expand wdParagraph

Next i

End If

End If

If myWords = "" Then

If Selection.Start = Selection.End Then Selection.Expand wdWord

If findWords = "" Then

myWords = InputBox("Names to find?", "WordsInContext", \_

Trim(Selection))

If myWords = "" Then Exit Sub

Else

myWords = findWords

End If

myWords = Replace(myWords, "| ", "|")

myWords = Replace(myWords, " |", "|")

myWords = Replace(myWords, "|", CR)

myWords = Replace(CR & myWords & CR, CR & CR, CR)

myWords = Replace(myWords, CR & CR, CR)

myWords = Replace(myWords, CR & CR, CR)

myWd = Split(myWords, CR)

totWords = UBound(myWd) - 1

ReDim myCol(totWords) As Integer

For i = 1 To totWords

myCol(i) = myBasicColour

Next i

End If

' In case we're in the external list...

mainDoc.Activate

Set rng = ActiveDocument.Content

Documents.Add

myTestWords = Replace(myWords, ChrW(172), "")

For Each myPara In rng.Paragraphs

parText = myPara.Range.Text

StatusBar = parText

copyIt = False

If Left(parText, 13) = "Context words" Then Exit For

For Each wd In myPara.Range.Words

DoEvents

Set myrange = wd.Duplicate

For i = 1 To maxWds

theseWds = Trim(myrange.Text)

myTest = CR & theseWds & CR

If InStr(LCase(myTestWords), LCase(myTest)) > 0 Then

copyIt = True

Exit For: Exit For

End If

myrange.MoveEnd wdWord, 1

Next i

Next wd

If copyIt Then

myPara.Range.Copy

Selection.Paste

Selection.Collapse wdCollapseEnd

Selection.TypeText sp

DoEvents

End If

Next myPara

Selection.HomeKey Unit:=wdStory

Selection.TypeText "Words/phrases in context" & vbCr & vbCr

ActiveDocument.Paragraphs(1).Range.Style = wdStyleHeading2

Selection.MoveLeft , 2

ActiveDocument.Content.HighlightColorIndex = wdNoHighlight

oldColour = Options.DefaultHighlightColorIndex

myWd = Split(myWords, CR)

For i = 1 To totWords

If Asc(myWd(i)) = 172 Then

myWd(i) = Mid(myWd(i), 2)

CaseSensitive = False

Else

CaseSensitive = True

End If

Options.DefaultHighlightColorIndex = myCol(i)

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = myWd(i)

.Replacement.Text = ""

.Replacement.Highlight = True

.MatchCase = CaseSensitive

.MatchWildcards = False

.MatchWholeWord = False

.Execute Replace:=wdReplaceAll

End With

Next i

Options.DefaultHighlightColorIndex = oldColour

For i = ActiveDocument.Paragraphs.Count To 2 Step -1

Set myPara = ActiveDocument.Paragraphs(i).Range

If Len(myPara.Text) > 1 And myPara.HighlightColorIndex = wdNoHighlight Then

myPara.Select

Selection.MoveEnd , multiSpace

Selection.Delete

End If

Next i

If returnToText = True Then mainDoc.Activate

Beep

End Sub

## Appendix: All 5 Macros for Fiction Editors

Sub HeadingsWordcount()

Application.ScreenUpdating = False

Dim RngHd As Range, h As Long, strOut As String

h = CLng(InputBox("Input the Heading level (e.g. 1) for the heading spans to count", "Heading Span Word Counter", 1))

If (h < 1) Or (h > 9) Then Exit Sub

With ActiveDocument.Range

With .Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = ""

.Style = "Heading " & h

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindStop

.Format = True

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

.Execute

End With

Do While .Find.Found

Set RngHd = .Paragraphs(1).Range

Set RngHd = RngHd.GoTo(What:=wdGoToBookmark, Name:="\HeadingLevel")

With RngHd

strOut = strOut & .ComputeStatistics(wdStatisticWords) - .Paragraphs.First.Range.ComputeStatistics(wdStatisticWords) & vbTab & .Paragraphs.First.Range.Text

End With

.Start = RngHd.End

.Find.Execute

Loop

End With

Set RngHd = Nothing

ActiveDocument.Range.InsertAfter vbCr & "The following word counts are associated with each level " & h & " heading:" & vbCr & strOut

Application.ScreenUpdating = True

End Sub

**Sub CatchPhrase()**

' Paul Beverley - Version 18.05.18

' Searches for repeated phrases/sentences

group\_a = "25, 6(4), 5(8)"

group\_a = "4(4), 3(9)"

group\_b = "6(3), 5(8), 4(10)"

group\_c = "7(3), 6(5), 5(10), 4(15)"

' Number of spaces times n

mySpaces = " ": n = 20

highlightFinds = False

highlightFinds = True

myColour = wdYellow

goExtraFast = True

goExtraFast = False

giveSpeedWarning = True

stopAndShowTime = False

myDots = ".... "

If Application.Visible = False Then Application.Visible \_

= True: Exit Sub

myLap = 1

Set rng = ActiveDocument.Content

rng.End = 200

If LCase(rng.Text) <> rng.Text Then

If giveSpeedWarning = True Then

myResponse = MsgBox("Preparing words file. This may take some time." & vbCr \_

& vbCr & "Please ignore any ""Not Responding"" warnings." \_

& vbCr & vbCr & "Click Yes to start.", vbQuestion \_

+ vbYesNo, "WordsPhrasesInContext")

If myResponse <> vbYes Then Exit Sub

Else

StatusBar = "Preparing words file. This may take some time."

End If

Set rng = ActiveDocument.Content

Documents.Add

Selection.Text = LCase(rng.Text)

For i = 1 To 6

sps = sps & " "

Next i

' Remove all except pure text, hyphens and apostrophes

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "'"

.Wrap = wdFindContinue

.Replacement.Text = "jqjq"

.MatchCase = False

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

DoEvents

StatusBar = sps & "Preparing words file. This may take some time. Six..."

DoEvents

.Text = "[!a-zA-Z,\- ]"

.MatchWildcards = True

.Replacement.Text = " "

.Execute Replace:=wdReplaceAll

DoEvents

StatusBar = sps & "Preparing words file. This may take some time. Five..."

DoEvents

.Text = " [ ,-]{1,}"

.Replacement.Text = " "

.MatchWildcards = True

.Execute Replace:=wdReplaceAll

DoEvents

StatusBar = sps & "Preparing words file. This may take some time. Four..."

DoEvents

.Text = ","

.Replacement.Text = "cmcm"

.Execute Replace:=wdReplaceAll

DoEvents

StatusBar = sps & "Preparing words file. This may take some time. Three.."

DoEvents

.MatchWildcards = True

.Text = "-{1,}"

.Replacement.Text = "cqcq"

.Execute Replace:=wdReplaceAll

DoEvents

StatusBar = sps & "Preparing words file. This may take some time. Two..."

DoEvents

.Text = " [a-hj-z] "

.Replacement.Text = " "

.Execute Replace:=wdReplaceAll

DoEvents

StatusBar = sps & "Preparing words file. This may take some time. One!"

DoEvents

.Text = " {2,}"

.Replacement.Text = " "

.Execute Replace:=wdReplaceAll

End With

End If

CR = vbCr: CR2 = CR & CR

For j = 1 To n

sps = sps & mySpaces

Next j

myPrompt = "a = " & group\_a & CR2

myPrompt = myPrompt & "b = " & group\_b & CR2

myPrompt = myPrompt & "c = " & group\_c & CR2 & CR

myPrompt = myPrompt & "t = Test to estimate the ETA" & CR2

Do

myChoice = InputBox(myPrompt, "CatchPhrase", "a")

If myChoice = "" Then Beep: Exit Sub

Loop Until InStr("abct", myChoice) > 0 \_

Or InStr("123456789", Left(myChoice, 1)) > 0

Set resultsDoc = ActiveDocument

Set rng0 = ActiveDocument.Content

Documents.Add

Set wordsDoc = ActiveDocument

Set rng = wordsDoc.Content

rng.Text = LCase(rng0.Text)

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "@@@@@"

.Wrap = wdFindContinue

.Replacement.Text = ""

.Forward = True

.MatchCase = False

.MatchWildcards = False

.Execute

End With

If rng.Find.Found Then

rng.End = wordsDoc.Content.End

rng.Delete

End If

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = " {2,}"

.MatchWildcards = True

.Replacement.Text = " "

.Execute Replace:=wdReplaceAll

End With

Selection.HomeKey Unit:=wdStory

Dim numWds(20) As Integer

Dim numShows(20) As Integer

Dim myList(4) As String

Dim myCount As Integer

myList(1) = group\_a

myList(2) = group\_b

myList(3) = group\_c

myList(4) = "5,"

If LCase(myChoice) <> UCase(myChoice) Then

If myChoice = "t" Then

myWdsList = "5,"

jumpFwd = 200

ActiveDocument.Words(jumpFwd).Select

Selection.Collapse wdCollapseStart

Selection.TypeText Text:="this is pauljqjqs speed " \_

& "test rhubarb this is pauljqjqs speed test "

Else

myWdsList = myList(Asc(myChoice) - 96)

myWdsList = Replace(myWdsList, " ", "") & ","

myWdsList = Replace(myWdsList, ",,", ",")

End If

Else

myWdsList = myChoice & ","

End If

myRun = Split(myWdsList, ",")

numRuns = UBound(myRun) - 1

myOutput = ""

st0 = Timer

If goExtraFast = True Then Application.Visible = False

For j = 0 To numRuns

Selection.HomeKey Unit:=wdStory

myTask = myRun(j)

' Search for phrases

myPhrases = ""

shownOne = False

totWds = wordsDoc.Words.Count

phrLen = Val(myTask)

myMinWds = 2

bktPos = InStr(myTask, "(")

myTask = Mid(myTask, bktPos + 1)

If bktPos > 0 Then

myMinWds = Val(myTask)

ignoreSubPhrases = False

Else

ignoreSubPhrases = True

End If

tstPhrase = ""

For n = 1 To phrLen

tstPhrase = tstPhrase & "dummy "

Next n

i = 1

st = Timer

myPrompt = "No duplicate phrases found yet" \_

& " Wds: " & phrLen & "(" & myMinWds & ")"

displayPhrase = myPrompt

For Each wd In wordsDoc.Words

spPos = InStr(tstPhrase, " ")

tstPhrase = Mid(tstPhrase, spPos + 1) & wd.Text

i = i + 1

nw = Timer

pc = Str(Int(1000 \* i / totWds) / 10)

If InStr(pc, ".") = 0 Then pc = pc & ".0"

StatusBar = sps & pc & "% " & phrLen & "(" & \_

myMinWds & ")" & " ETA " & predictedTime

If InStr(myPhrases, tstPhrase & myDots) = 0 Then

DoEvents

Set rng = wordsDoc.Content

' Find the first one

tstLen = Len(tstPhrase)

If tstLen > 255 Then tstPhrase = Left(tstPhrase, 254)

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = tstPhrase

If ignoreSubPhrases = True Then .Font.Underline = False

.Replacement.Text = ""

.MatchWildcards = False

End With

phrFreq = -1

Do

phrFreq = phrFreq + 1

rng.Find.Execute

rng.Collapse wdCollapseEnd

Loop Until rng.Find.Found = False

If phrFreq > 1 Then

newPhrase = tstPhrase & myDots & Trim(Str(phrFreq))

displayPhrase = Replace(newPhrase, "cmcm", ",")

displayPhrase = Replace(displayPhrase, "cqcq", "-")

displayPhrase = Replace(displayPhrase, "jqjq", "'")

DoEvents

ActiveDocument.ActiveWindow.Caption = \_

sps & "LATEST FIND: " & displayPhrase & sps & sps

myPrompt = " Wds: " & phrLen & "(" & myMinWds & ")"

timeToGo = (nw - st) \* (totWds - i) / i

myTime = Time

myETA = DateAdd("s", timeToGo, myTime)

predictedTime = Left(myETA, 5)

If myChoice = "t" Then

MsgBox "ETA: " & predictedTime & " = " & \_

Int(timeToGo / 6) / 10 & " min"

wordsDoc.Close SaveChanges:=False

Exit Sub

End If

myPrompt = myPrompt & " ETA " & predictedTime

If i > totWds Then

ahfkjhasdkjgf = 0

End If

pc = Str(Int(1000 \* i / totWds) / 10)

If InStr(pc, ".") = 0 Then pc = pc & ".0"

spd = Str(Int(10 \* i / (nw - st)) / 10)

If InStr(spd, ".") = 0 Then spd = spd & ".0"

Debug.Print spd & " " & pc & "% " & myPrompt & \_

" " & displayPhrase

myPhrases = myPhrases & newPhrase & vbCr

If phrFreq > myMinWds - 1 Then

myOutput = myOutput & newPhrase & vbCr

If highlightFinds Then

oldColour = Options.DefaultHighlightColorIndex

Options.DefaultHighlightColorIndex = myColour

With rng0.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = tstPhrase

.Wrap = wdFindContinue

.Replacement.Text = ""

.Replacement.Highlight = True

.Execute Replace:=wdReplaceAll

End With

Options.DefaultHighlightColorIndex = oldColour

End If

End If

DoEvents

End If

End If

Next wd

myOutput = Replace(myOutput, "cqcq", "-")

myOutput = Replace(myOutput, "cmcm", ",")

myOutput = Replace(myOutput, "jqjq", ChrW(8217))

rng0.InsertAfter Text:=vbCr & "@@@@@@@@@@@@@@@@@@@@@ " & \_

ChrW(8211) & " " & phrLen & vbCr & myOutput & vbCr

myPhrases = ""

myOutput = ""

t = Timer - st

If t > 600 Then

ttot = Int(t / 6) / 10

tText = Str(ttot) & " min"

Else

ttot = Int(t \* 10) / 10

tText = Str(ttot) & " sec"

End If

myResult = "Ave wds/sec: " & Int(10 \* i / t) / 10 & vbCr & vbCr

myResult = myResult & "Time: " & tText

rng0.InsertAfter Text:=vbCr & "================== " & vbCr \_

& myResult & vbCr

If stopAndShowTime = True Then

Application.Visible = True

MsgBox myResult

End If

Next j

StatusBar = " "

t = Timer - st0

If t > 600 Then

ttot = Int(t / 6) / 10

tText = Str(ttot) & " min"

Else

ttot = Int(t \* 10) / 10

tText = Str(ttot) & " sec"

End If

myResult = "Ave wds/sec: " & Int(10 \* i \* (numRuns + 1) / t) / 10 \_

& vbCr & vbCr

myResult = myResult & "Total time: " & tText

rng0.InsertAfter Text:=vbCr & "================== " & vbCr \_

& "================== " & vbCr & myResult & vbCr

Application.Visible = True

Beep

ActiveDocument.ActiveWindow.Caption = ""

MsgBox myResult

wordsDoc.Close SaveChanges:=False

resultsDoc.Activate

Selection.HomeKey Unit:=wdStory

With Selection.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "@@@@@"

.Replacement.Text = ""

.MatchWildcards = False

.Execute

End With

Set rng = Selection.range

rng.End = ActiveDocument.Content.End

rng.HighlightColorIndex = wdNoHighlight

Selection.Collapse wdCollapseStart

Application.StatusBar = False

Beep

End Sub

Sub ProperNounAlyse()

' Paul Beverley - Version 22.08.23

' Analyses similar proper nouns

minLengthCheck = 4

includeAcronyms = True

ignoreWords = "The This There Those Their They Then These That"

similarChars = "bb,b; b,p; sch,sh; ch,sh; c,k; ph,f; ss,z; s,z;" & \_

" mp,m; ll,l; nn,n; nd,n; nt,n;"

' With non-English languages, you might need to make this False

ignorePlurals = True

myScreenOff = True

Set rng = Selection.Range.Duplicate

rng.End = rng.Start + 1

myLanguage = Languages(rng.LanguageID).NameLocal

Set FUT = ActiveDocument

doingSeveralMacros = (InStr(FUT.Name, "zzTestFile") > 0)

If doingSeveralMacros = False Then

myResponse = MsgBox(" ProperNounAlyse" & vbCr & vbCr & \_

"Analyse this document?", vbQuestion \_

+ vbYesNoCancel, "ProperNounAlyse")

If myResponse <> vbYes Then Exit Sub

End If

If myScreenOff = True Then

Application.ScreenUpdating = False

On Error GoTo ReportIt

End If

myDummy = ChrW(222)

For i = 1 To 100

spcs = " " & spcs

Next i

dummyText = ChrW(197) & "zzzx "

For i = 65 To 90

dummyText = dummyText & ChrW(i) & "zzzz "

Next i

checkFinalLetters = True

' checkFinalLetters = False

' Grey on word only

thisHighlight = wdGray25

doMissingLetter = True

' doMissingLetter = False

' Bold And blue

switchTest = True

' switchTest = False

' double strikethrough

doSimilarLetters = True

' doSimilarLetters = False

' various highlight colours + underline

doVowelTest = True

' doVowelTest = False

' various highlight colours + italic

' These last two tests cycle through these colours:

maxCol = 6

ReDim myCol(maxCol) As Integer

myCol(1) = wdYellow

myCol(2) = wdBrightGreen

myCol(3) = wdTurquoise

myCol(4) = wdRed

myCol(5) = wdPink

myCol(6) = wdGray25

colcode = 0

oldColour = Options.DefaultHighlightColorIndex

Options.DefaultHighlightColorIndex = wdGray25

leadDots = " . . . "

title1 = "Proper noun list"

title2 = "Proper noun queries"

CR = vbCr: CR2 = CR & CR

convCharsUC = "AAAAAAA.EEEEIIII..OOOOO.OUUUU" & \_

"...aaaaaaa.eeeeiiiio.ooooo.ouuuu......"

convCharsLC = LCase(convCharsUC)

timeStart = Timer

' collect notes text, if any

endText = ""

footText = ""

If ActiveDocument.Endnotes.count > 0 Then

endText = ActiveDocument.StoryRanges(wdEndnotesStory).Text

End If

If ActiveDocument.Footnotes.count > 0 Then

footText = ActiveDocument.StoryRanges(wdFootnotesStory).Text

End If

' collect text in all the textboxes (if any)

sh = ActiveDocument.Shapes.count

If sh > 0 Then

ReDim shText(sh)

i = 0

For Each shp In ActiveDocument.Shapes

If shp.Type <> 24 And shp.Type <> 3 Then

If shp.TextFrame.HasText Then

i = i + 1

shText(i) = shp.TextFrame.TextRange.Text

End If

End If

Next

shCount = i

End If

' Create various documents

Set rng = ActiveDocument.Content

Documents.Add

Set firstDoc = ActiveDocument

Set fnl = ActiveDocument.Content

Documents.Add

Set tempDoc = ActiveDocument

Set tmp = ActiveDocument.Content

Documents.Add

Set allText = ActiveDocument

Selection.TypeText dummyText & vbCr

Selection.FormattedText = rng.FormattedText

Selection.Collapse wdCollapseEnd

' Add notes + shape text

Selection.TypeText endText & CR & footText & CR

If shCount > 0 Then

For i = 1 To shCount

Selection.TypeText shText(i) & CR

Next i

End If

Selection.HomeKey Unit:=wdStory

Set rng = allText.Content

rng.Revisions.AcceptAll

DoEvents

StatusBar = spcs & "Preparing copied file - 1"

DoEvents

' Delete struck-through text

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = ""

.MatchWildcards = False

.Font.StrikeThrough = True

.Replacement.Text = " "

.Execute Replace:=wdReplaceAll

End With

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "["

.MatchWildcards = False

.Replacement.Text = " "

.Execute Replace:=wdReplaceAll

End With

' Remove strange unicode characters

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "[" & ChrW(&HA000) & "-" & ChrW(&HD6FF) & "]{1,}"

.MatchWildcards = True

.Replacement.Text = " "

.Execute Replace:=wdReplaceAll

End With

DoEvents

StatusBar = spcs & "Preparing copied file - 2"

DoEvents

' Cut all and replace as pure text

Set rng = allText.Content

tmp.FormattedText = rng.FormattedText

rng.Text = tmp.Text

tmp.Delete

DoEvents

StatusBar = spcs & "Preparing copied file - 3"

' Use qqq for apostrophe

Set rng = allText.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "n" & ChrW(8217) & "t"

.MatchWildcards = False

.Replacement.Text = "nqqqt"

.Execute Replace:=wdReplaceAll

End With

' Use qq for apostrophe

With rng.Find

.Text = "O'"

.MatchCase = True

.Replacement.Text = "Oqqq"

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

' Find initial cap words

DoEvents

StatusBar = spcs & "Preparing copied file - 4"

DoEvents

myChopNum = minLengthCheck - 2

If myChop < 1 Then myChop = 1

myChop = Trim(Str(myChopNum))

myFind = "<[A-Z][a-z][a-zA-Z]{" & myChop & ",}"

If includeAcronyms = True Then myFind = \_

"<[A-Z][a-zA-Z][a-zA-Z]{" & myChop & ",}"

Set rng = allText.Range

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = myFind

.MatchWildcards = True

.MatchCase = True

.Replacement.Text = "^&"

.Replacement.Highlight = True

.Replacement.Font.StrikeThrough = True

.Execute Replace:=wdReplaceAll

End With

' Delete all non-strikethrough words

DoEvents

StatusBar = spcs & "Preparing copied file - 5"

DoEvents

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = ""

.Font.StrikeThrough = False

.MatchWildcards = False

.MatchCase = True

.Replacement.Text = "^p"

.Execute Replace:=wdReplaceAll

End With

' Delete the unwanted "proper nouns"

DoEvents

StatusBar = spcs & "Preparing copied file - 6"

igWords = Split(Trim(ignoreWords), " ")

For Each wd In igWords

Set rng = allText.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = wd & "^p"

.Wrap = wdFindContinue

.Replacement.Text = ""

.MatchCase = True

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

DoEvents

Next wd

StatusBar = spcs & "Sorting whole file"

DoEvents

i = 0

For ch = 65 To 90

For Each myPara In allText.Paragraphs

If Asc(myPara.Range) = ch Then

DoEvents

myPara.Range.Font.StrikeThrough = False

tmp.InsertAfter myPara.Range.Text

End If

Next myPara

tmp.InsertAfter Text:="Zzzzz" & CR

Set rng = tempDoc.Content

rng.Sort SortOrder:=wdSortOrderAscending, CaseSensitive:=True

' delete initial blank line

If Len(tempDoc.Paragraphs(1)) < 3 Then \_

tempDoc.Paragraphs(1).Range.Delete

' Create a frequency for each highlighted word

thisWord = ""

myCount = 0

For Each myPara In tempDoc.Paragraphs

Set rng = myPara.Range.Words(1)

DoEvents

nextWord = rng

If nextWord <> thisWord Then

' This is a new word

If Len(thisWord) > 1 Then

fnl.InsertAfter Text:=thisWord \_

& leadDots & Trim(Str(myCount)) & CR

End If

thisWord = nextWord

myCount = 1

Else

myCount = myCount + 1

End If

If nextWord = "Zzzzz" Then Exit For

i = i + 1:

If i Mod 400 = 4 Then

DoEvents

prmt = Left(thisWord, 1) & " "

prmt = prmt & prmt & prmt & prmt

StatusBar = spcs & \_

"Preparing words for frequency list - " & prmt

DoEvents

End If

Next myPara

' Remove all words except frequency counts

Set rng = tempDoc.Content

rng.Delete

Next ch

' Find any unaccounted-for words, e.g. Ångstrom

For Each myPara In allText.Paragraphs

If myPara.Range.Words(1).Font.StrikeThrough = True Then

tmp.InsertAfter myPara.Range.Text

End If

Next myPara

tempDoc.Close SaveChanges:=False

allText.Close SaveChanges:=False

firstDoc.Activate

' Remove blank lines

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "[^13]{2,}"

.Wrap = wdFindContinue

.Replacement.Text = "^p"

.Forward = True

.MatchWildcards = True

.Execute Replace:=wdReplaceAll

End With

' Resort case insensitively

Set rng = ActiveDocument.Content

rng.Sort SortOrder:=wdSortOrderAscending, \_

CaseSensitive:=False

' Delete rubbish from top and bottom of list

Do

Set rng = ActiveDocument.Paragraphs(1).Range

myLen = Len(rng.Text)

If myLen < 10 Then

rng.Select

Selection.Delete

End If

Loop Until myLen > 9

Do

lastLine = ActiveDocument.Paragraphs.count

Set rng = ActiveDocument.Paragraphs(lastLine).Range

myLen = Len(rng.Text)

If myLen < 10 Then

rng.Select

Selection.Delete

End If

Loop Until Len(rng.Text) >= 2

' Word list now has freq. count.

Do

lastLine = ActiveDocument.Paragraphs.count

Set rng = ActiveDocument.Paragraphs(lastLine).Range

myLen = Len(rng.Text)

If myLen < 10 Then

rng.Select

Selection.Delete

End If

Loop Until Len(rng.Text) >= 2

' Create another copy for doing extra tests

Set rng = ActiveDocument.Content

Documents.Add

Set finalList = ActiveDocument

finalList.Range.Text = rng.Text

Selection.HomeKey Unit:=wdStory

' Prepare data for other tests

numWords = ActiveDocument.Paragraphs.count

For i = 1 To numWords

aWord = ActiveDocument.Paragraphs(i).Range.Words(1)

n = AscW(aWord)

thisChar = ChrW(n)

If n > 129 Then

If n >= 217 Then aWord = Replace(aWord, thisChar, "U")

If n >= 210 Then aWord = Replace(aWord, thisChar, "O")

If n >= 204 Then aWord = Replace(aWord, thisChar, "I")

If n >= 200 Then aWord = Replace(aWord, thisChar, "E")

If n >= 192 Then aWord = Replace(aWord, thisChar, "A")

End If

allWords = allWords & aWord

jmp = 100

If i Mod jmp = 1 Then

pq = pq + 1

DoEvents

StatusBar = spcs & \_

"Preparing data for other tests - 1 - " & pq

DoEvents

End If

Next i

' ...for the vowel test below

DoEvents

StatusBar = spcs & "Preparing data for other tests - 2"

DoEvents

noVowelWords = " " & allWords

noVowelWords = Replace(noVowelWords, " A", "\_1")

noVowelWords = Replace(noVowelWords, " E", "\_2")

noVowelWords = Replace(noVowelWords, " I", "\_3")

noVowelWords = Replace(noVowelWords, " O", "\_4")

noVowelWords = Replace(noVowelWords, " U", "\_5")

noVowelWords = Replace(noVowelWords, " Y", "\_6")

For k = 2 To Len(noVowelWords) - 1

thisChar = Mid(noVowelWords, k, 1)

n = AscW(thisChar)

If n > 191 And n < 221 Then

myNewChar = Mid(convCharsLC, n - 191, 1)

If myNewChar <> "." Then noVowelWords = \_

Replace(noVowelWords, thisChar, myNewChar)

End If

Next k

noVowelWords = Replace(noVowelWords, "a", "")

noVowelWords = Replace(noVowelWords, "e", "")

noVowelWords = Replace(noVowelWords, "i", "")

noVowelWords = Replace(noVowelWords, "o", "")

noVowelWords = Replace(noVowelWords, "u", "")

noVowelWords = Replace(noVowelWords, "y", "")

noVowelWords = Replace(noVowelWords, "A", "")

noVowelWords = Replace(noVowelWords, "E", "")

noVowelWords = Replace(noVowelWords, "I", "")

noVowelWords = Replace(noVowelWords, "O", "")

noVowelWords = Replace(noVowelWords, "U", "")

noVowelWords = Replace(noVowelWords, "Y", "")

noVowelWords = Replace(noVowelWords, "\_1", " A")

noVowelWords = Replace(noVowelWords, "\_2", " E")

noVowelWords = Replace(noVowelWords, "\_3", " I")

noVowelWords = Replace(noVowelWords, "\_4", " O")

noVowelWords = Replace(noVowelWords, "\_5", " U")

noVowelWords = Replace(noVowelWords, "\_6", " Y")

' ...for the similar words test

DoEvents

StatusBar = spcs & "Preparing data for other tests - 3"

DoEvents

similarAllWords = " " & LCase(allWords)

similarChars = Replace(similarChars, " ", "")

sChars = Replace(similarChars, " ", "")

Do

commaPos = InStr(sChars, ",")

charWas = Left(sChars, commaPos - 1)

sChars = Mid(sChars, commaPos + 1)

semicolonPos = InStr(sChars, ";")

charNew = Left(sChars, semicolonPos - 1)

sChars = Mid(sChars, semicolonPos + 1)

similarAllWords = Replace(similarAllWords, charWas, charNew)

Loop Until Len(sChars) < 2

' Changes all the accented characters to non-accented

DoEvents

StatusBar = spcs & "Preparing data for other tests - 4"

DoEvents

sWd = similarAllWords

For k = 1 To Len(sWd) - 1

thisChar = Mid(sWd, k, 1)

n = AscW(thisChar)

myNewChar = "."

If n > 191 And n < 256 Then

myNewChar = Mid(convCharsLC, n - 191, 1)

If myNewChar <> "." Then sWd = Replace(sWd, \_

thisChar, myNewChar)

End If

Next k

similarAllWords = sWd

' Catch words with only the final two letters the same

i = 0

If checkFinalLetters = True Then

For Each myPara In ActiveDocument.Paragraphs

gotOne = False

myWord = Trim(myPara.Range.Words(1))

myLen = Len(myWord)

If myLen > 6 Then

myTarget = "^p" & Left(myWord, myLen - 2) & "^$^$ "

myCut = 2

Else

myTarget = "^p" & Left(myWord, myLen - 1) & "^$ "

myCut = 1

End If

Set rng = ActiveDocument.Content

rng.Start = myPara.Range.End - 3

rng.Collapse wdCollapseStart

With rng.Find

.Replacement.ClearFormatting

.ClearFormatting

.Text = myTarget

.Replacement.Text = ""

.Forward = True

.MatchCase = True

.MatchWildcards = False

.Wrap = wdFindStop

End With

rng.Find.Execute

Do While rng.Find.Found

gotOne = True

rng.MoveStart 1

rng.End = rng.Start + myLen - myCut

rng.HighlightColorIndex = thisHighlight

rng.Font.Bold = True

rng.Find.Execute

Loop

If gotOne = True Then

Set rng = myPara.Range.Words(1)

rng.End = rng.Start + myLen - myCut

rng.HighlightColorIndex = thisHighlight

rng.Font.Bold = True

End If

i = i + 1

If i Mod 100 = 1 Then

DoEvents

StatusBar = spcs & "Doing test (5) on " & myWord

DoEvents

End If

Next myPara

End If

If doMissingLetter = True Then

' Start of test

doneWords = ""

doneSimilarWords = ""

McList = ""

For i = 1 To ActiveDocument.Paragraphs.count - 1

myWord = ActiveDocument.Paragraphs(i).Range.Words(1)

n = AscW(myWord)

thisChar = ChrW(n)

myNewChar = "."

' Changes the capital letter, if a vowel

If n > 191 And n < 221 Then

myNewChar = Mid(convCharsUC, n - 191, 1)

If myNewChar <> "." Then myWord = Replace(myWord, \_

thisChar, myNewChar)

End If

If i Mod 50 = 1 Then

DoEvents

StatusBar = spcs & "Other tests (4) on " & myWord

DoEvents

End If

testWords = Replace(allWords, myWord, "")

captestLetters = Left(myWord, 1)

' Check if word reappears with one letter missing (1)

For k = 2 To Len(myWord) - 1

testWord = " " & Left(myWord, k - 1) & Mid(myWord, k + 1)

wordPos = InStr(allWords, testWord)

If wordPos > 0 Then

lastLetter = Mid(myWord, Len(myWord) - 1, 1)

' but not "s" at the end, unless it's a spelling error

If lastLetter = "s" Then

ignoreIt = (Application.CheckSpelling(myWord, \_

MainDictionary:=myLanguage) = True)

Else

ignoreIt = False

End If

If ignoreIt = False And ignorePlurals = True Then

colcode = (colcode + 1) Mod maxCol

thisCol = myCol(colcode + 1)

' mark the pair

leftBit = Left(allWords, InStr(allWords, testWord) \_

+ Len(testWord) - 1)

j = Len(leftBit) - Len(Replace(leftBit, " ", ""))

Set rng = ActiveDocument.Paragraphs(i).Range

rng.HighlightColorIndex = thisCol

rng.Font.Bold = True

rng.Font.Color = wdColorBlue

Set rng = ActiveDocument.Paragraphs(j).Range

rng.HighlightColorIndex = thisCol

rng.Font.Bold = True

rng.Font.Color = wdColorBlue

End If

End If

Next k

If Left(myWord, 2) = "Mc" Or Left(myWord, 3) = "Mac" Or \_

Left(myWord, 3) = "Mag" Then

McList = McList & ActiveDocument.Paragraphs(i).Range

End If

Next i

End If

If doSimilarLetters = True Then

doneWords = ""

doneSimilarWords = ""

For i = 1 To ActiveDocument.Paragraphs.count - 1

myWord = ActiveDocument.Paragraphs(i).Range.Words(1)

n = AscW(myWord)

thisChar = ChrW(n)

myNewChar = "."

' Changes the capital letter, if a vowel

If n > 191 And n < 221 Then

myNewChar = Mid(convCharsUC, n - 191, 1)

If myNewChar <> "." Then myWord = Replace(myWord, \_

thisChar, myNewChar)

End If

If i Mod 50 = 1 Then

DoEvents

StatusBar = spcs & "Other tests (3) on " & myWord

DoEvents

End If

testWords = Replace(allWords, myWord, "")

captestLetters = Left(myWord, 1)

' check similar spellings: Perutz/Peruts or Chebyshev/Chevychev

similarWord = " " & LCase(myWord)

sChars = similarChars

Do

commaPos = InStr(sChars, ",")

charWas = Left(sChars, commaPos - 1)

sChars = Mid(sChars, commaPos + 1)

semicolonPos = InStr(sChars, ";")

charNew = Left(sChars, semicolonPos - 1)

sChars = Mid(sChars, semicolonPos + 1)

similarWord = Replace(similarWord, charWas, charNew)

Loop Until Len(sChars) < 2

' Changes all the accented characters to non-accented

For k = 1 To Len(myWord) - 1

thisChar = Mid(myWord, k, 1)

n = AscW(thisChar)

If n > 191 And n < 256 Then

myNewChar = Mid(convCharsUC, n - 191, 1)

If myNewChar <> "." Then myWord = Replace(myWord, \_

thisChar, myNewChar)

End If

Next k

similarAllWords = Mid(similarAllWords, Len(similarWord))

theseWords = similarAllWords

If InStr(doneSimilarWords, similarWord) = 0 And \_

InStr(theseWords, similarWord) > 0 Then

colcode = (colcode + 1) Mod maxCol

thisCol = myCol(colcode + 1)

Set rng = ActiveDocument.Paragraphs(i).Range

rng.HighlightColorIndex = thisCol

rng.Font.Underline = True

doneSimilarWords = doneSimilarWords & similarWord

' search through all the following words

theseWords = similarAllWords

For j = 1 To numWords - i

spPos = InStr(Trim(theseWords) & " ", " ")

If Left(theseWords, spPos + 1) = similarWord Then

Set rng = ActiveDocument.Paragraphs(i + j).Range

rng.HighlightColorIndex = thisCol

rng.Font.Underline = True

End If

theseWords = Mid(theseWords, spPos + 1)

capThisLetter = Mid(theseWords, 2, 1)

If capThisLetter <> LCase(captestLetters) Then Exit For

Next j

End If

Next i

End If

If switchTest = True Then

doneWords = ""

doneSimilarWords = ""

McList = ""

For i = 1 To ActiveDocument.Paragraphs.count - 1

myWord = ActiveDocument.Paragraphs(i).Range.Words(1)

n = AscW(myWord)

thisChar = ChrW(n)

myNewChar = "."

' Changes the capital letter, if a vowel

If n > 191 And n < 221 Then

myNewChar = Mid(convCharsUC, n - 191, 1)

If myNewChar <> "." Then myWord = Replace(myWord, \_

thisChar, myNewChar)

End If

If i Mod 50 = 1 Then

DoEvents

StatusBar = spcs & "Other tests (2) on " & myWord

DoEvents

End If

testWords = Replace(allWords, myWord, "")

captestLetters = Left(myWord, 1)

' check for switched chars

wordLen = Len(myWord) - 1

For k = 1 To Len(myWord) - 3

otherWord = Left(myWord, k) & Mid(myWord, k + 2, 1) & \_

Mid(myWord, k + 1, 1) & Mid(myWord, k + 3)

wordPos = InStr(testWords, otherWord)

If wordPos > 0 Then

' Find the position of the matching word

matchWord = Mid(testWords, wordPos, Len(myWord))

leftBit = Left(allWords, InStr(allWords, matchWord) + 1)

j = Len(leftBit) - Len(Replace(leftBit, " ", "")) + 1

ActiveDocument.Paragraphs(i).Range.Font.DoubleStrikeThrough \_

= True

ActiveDocument.Paragraphs(i).Range.HighlightColorIndex \_

= thisCol

ActiveDocument.Paragraphs(j).Range.Font.DoubleStrikeThrough \_

= True

ActiveDocument.Paragraphs(j).Range.HighlightColorIndex \_

= thisCol

End If

Next k

Next i

End If

If doVowelTest = True Then

doneWords = ""

doneSimilarWords = ""

McList = ""

For i = 1 To ActiveDocument.Paragraphs.count - 1

myWord = ActiveDocument.Paragraphs(i).Range.Words(1)

n = AscW(myWord)

thisChar = ChrW(n)

myNewChar = "."

' Changes the capital letter, if a vowel

If n > 191 And n < 221 Then

myNewChar = Mid(convCharsUC, n - 191, 1)

If myNewChar <> "." Then myWord = Replace(myWord, \_

thisChar, myNewChar)

End If

If i Mod 50 = 1 Then

DoEvents

StatusBar = spcs & "Other tests (1) on " & myWord

DoEvents

End If

testWords = Replace(allWords, myWord, "")

captestLetters = Left(myWord, 1)

' check if there's a word with different vowels

otherWord = " " & Replace(myWord, "a", "")

otherWord = Replace(otherWord, "e", "")

otherWord = Replace(otherWord, "i", "")

otherWord = Replace(otherWord, "o", "")

otherWord = Replace(otherWord, "u", "")

otherWord = Replace(otherWord, "y", "")

' Delete all the accented characters

For k = 3 To Len(otherWord) - 1

thisChar = Mid(otherWord, k, 1)

n = AscW(thisChar)

If InStr("AEIOUY", thisChar) > 0 Then

otherWord = Left(otherWord, k - 1) & "=" & Mid(otherWord, k + 1)

Else

If n > 191 And n < 221 Then

myNewChar = Mid(convCharsUC, n - 191, 1)

If myNewChar <> "." Then

otherWord = Replace(otherWord, thisChar, "=")

End If

End If

End If

Next k

otherWord = Replace(otherWord, "=", "")

' otherWord is now the word under test (vowel-less)

otherWord = Replace(otherWord, ".", "")

noVowelWords = Mid(noVowelWords, Len(otherWord))

If Left(noVowelWords, 1) <> " " Then noVowelWords = \_

" " & noVowelWords

theseWords = noVowelWords

wordPos = InStr(noVowelWords, otherWord)

If InStr(doneWords, otherWord) = 0 And wordPos > 0 Then

colcode = (colcode + 1) Mod maxCol

thisCol = myCol(colcode + 1)

Set rng = ActiveDocument.Paragraphs(i).Range

rng.HighlightColorIndex = thisCol

rng.Font.Italic = True

doneWords = doneWords & otherWord

For j = 1 To numWords - i

spPos = InStr(Trim(theseWords) & " ", " ")

firstWord = Left(theseWords, spPos + 1)

theseWords = Mid(theseWords, spPos + 1)

If firstWord = otherWord Then

Set rng = ActiveDocument.Paragraphs(i + j).Range

rng.HighlightColorIndex = thisCol

rng.Font.Italic = True

End If

capThisLetter = Mid(theseWords, 2, 1)

If capThisLetter > "" And capThisLetter <> \_

captestLetters Then Exit For

Next j

End If

Next i

End If

finishOff:

Selection.EndKey Unit:=wdStory

Selection.TypeText CR2 & McList

Selection.HomeKey Unit:=wdStory

Selection.TypeText title1 & CR

Do

Selection.Expand wdParagraph

If Len(Selection) < 3 Or LCase(Selection) = \_

UCase(Selection) Then Selection.Delete

Loop Until LCase(Selection) <> UCase(Selection)

Selection.HomeKey Unit:=wdStory, Extend:=wdExtend

Selection.Style = ActiveDocument.Styles(wdStyleHeading1)

' Restore apostrophes

Set rng = finalList.Range

With rng.Find

.Text = "qqq"

.MatchCase = False

.Replacement.Text = "'"

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

' Find first highlight

Set rng = finalList.Content

With rng.Find

.Text = "Zzzzz"

.Wrap = wdFindStop

.Replacement.Text = ""

.Forward = True

.MatchWildcards = False

.Execute Replace:=wdReplaceOne

End With

Set rng = finalList.Content

With rng.Find

.Text = ""

.Highlight = True

.Wrap = wdFindStop

.Replacement.Text = ""

.Forward = True

.MatchWildcards = False

.Execute

End With

rng.Select

Selection.Collapse wdCollapseStart

Set finalList = ActiveDocument

firstDoc.Activate

' Find sets of sounds-like words

StatusBar = spcs & "Sounds-like tests"

k = 0

For Each myPara In ActiveDocument.Paragraphs

myWord = Trim(myPara.Range.Words(1))

k = k + 1

If k Mod 40 = 1 Then

DoEvents

StatusBar = spcs & "Sounds-like test: " & myWord

DoEvents

End If

hasAccent = False

For i = 1 To Len(myWord)

ascChar = AscW(Mid(myWord, i))

If ascChar > 128 Or ascChar = Asc("?") Then hasAccent = True

Next i

' Go and find the first sounds-like word

initLetter = Left(myWord, 1)

If Len(myWord) > 2 And myPara.Range.HighlightColorIndex > 0 And \_

hasAccent = False And InStr(allSets, myWord & leadDots) \_

= 0 Then

Set rng = ActiveDocument.Content

Do

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = myWord

.Wrap = wdFindStop

.Replacement.Text = ""

.MatchWildcards = False

.MatchSoundsLike = True

.Execute

End With

Set myPara = rng.Paragraphs(1).Range

rng.Collapse wdCollapseEnd

Loop Until Left(myPara, 1) = initLetter

setOfWords = myPara

gottaSet = False

rng.Collapse wdCollapseEnd

rng.Find.Execute

Do While rng.Find.Found = True

Set myPara = rng.Paragraphs(1).Range

If Left(myPara, 1) = initLetter Then

gottaSet = True

setOfWords = setOfWords & myPara

End If

rng.Collapse wdCollapseEnd

rng.Find.Execute

Loop

If gottaSet = True Then allSets = allSets & setOfWords & CR

End If

Next myPara

Selection.WholeStory

If Len(allSets) < 2 Then

Selection.TypeText "None found with this test"

Else

Selection.TypeText allSets

End If

Selection.HomeKey Unit:=wdStory

Selection.TypeText "Proper nouns by sound" & CR

Selection.HomeKey Unit:=wdStory, Extend:=wdExtend

Selection.Style = ActiveDocument.Styles(wdStyleHeading1)

Selection.HomeKey Unit:=wdStory

Set rng = ActiveDocument.Content

rng.HighlightColorIndex = 0

rng.Copy

ActiveDocument.Close SaveChanges:=False

finalList.Activate

' Remove highlighting from second half of words

' that are only case changes of one another

totParas = ActiveDocument.Paragraphs.count

For i = 1 To totParas - 1

A = Trim(ActiveDocument.Paragraphs(i).Range.Words(1))

b = Trim(ActiveDocument.Paragraphs(i + 1).Range.Words(1))

A = Mid(A, 2)

b = Mid(b, 2)

If LCase(A) = LCase(b) And Len(A) > 2 Then

If (UCase(A) = A And LCase(b) = b) Or (UCase(b) = b And \_

LCase(A) = A) Then

ActiveDocument.Paragraphs(i).Range.Words(1).HighlightColorIndex = 0

ActiveDocument.Paragraphs(i + 1).Range.Words(1).HighlightColorIndex \_

= 0

End If

End If

If i Mod 50 = 0 Then

DoEvents

StatusBar = spcs & "Final checks: " & totParas - i

DoEvents

End If

Next i

myOnames = ""

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "^13O[!a-z]"

.Wrap = wdFindStop

.Replacement.Text = ""

.Forward = True

.MatchSoundsLike = False

.MatchWildcards = True

.Execute

End With

Do While rng.Find.Found = True

rng.Collapse wdCollapseEnd

rng.Expand wdWord

wd = Mid(rng.Text, 3)

rng.Expand wdParagraph

pa = rng.Text

Set rng2 = ActiveDocument.Content

With rng2.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "^13" & wd

.Wrap = wdFindStop

.Replacement.Text = ""

.Forward = True

.MatchWildcards = True

.Execute

End With

If rng2.Find.Found Then

rng2.Collapse wdCollapseEnd

rng2.Expand wdParagraph

pa2 = rng2.Text

myOnames = myOnames & pa2 & pa & vbCr

End If

rng.Collapse wdCollapseEnd

rng.End = rng.End - 2

rng.Find.Execute

Loop

If myOnames > "" Then

Selection.EndKey Unit:=wdStory

Selection.TypeText "Possible O'<something> errors" & vbCr

Selection.MoveUp , 1

Selection.Style = ActiveDocument.Styles(wdStyleHeading1)

Selection.EndKey Unit:=wdStory

Selection.TypeText myOnames

Selection.HomeKey Unit:=wdStory

End If

Set rng = ActiveDocument.Content

finalList.Activate

Selection.EndKey Unit:=wdStory

Selection.TypeText vbCr & vbCr & vbCr

Selection.Paste

Selection.HomeKey Unit:=wdStory

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = myDummy

.Wrap = wdFindContinue

.Replacement.Text = " "

.Forward = True

.MatchCase = False

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "^$zzz^$" & leadDots & "1" & vbCr

.Wrap = wdFindContinue

.Replacement.Text = ""

.Forward = True

.MatchCase = False

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

' Clear clipboard

Set rng = ActiveDocument.Content

rng.End = 2

rng.Copy

Set finalList = ActiveDocument

StatusBar = "Creating queries list"

Set rng = ActiveDocument.Content

Documents.Add

Selection.FormattedText = rng.FormattedText

Set queriesDoc = ActiveDocument

ActiveDocument.Paragraphs(1).Range.Delete

Set rng = ActiveDocument.Content

rng.Font.StrikeThrough = True

For Each myPara In ActiveDocument.Paragraphs

Set ch = myPara.Range.Characters(1)

chCol = ch.HighlightColorIndex

If chCol > 0 Then

myPara.Range.Font.StrikeThrough = False

End If

myLen = Len(myPara.Range.Text)

If myLen > 4 Then

If chCol > 0 Then

myPara.Range.Font.StrikeThrough = False

End If

Set che = myPara.Range.Characters(myLen - 2)

If che.HighlightColorIndex > 0 Then

myPara.Range.Font.StrikeThrough = False

End If

End If

Next myPara

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = ""

.Font.StrikeThrough = True

.Wrap = wdFindContinue

.Replacement.Text = "^p"

.Forward = True

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

DoEvents

End With

Set rng = ActiveDocument.Content

rng.Font.StrikeThrough = False

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "[^13]{3,}"

.Wrap = wdFindContinue

.Replacement.Text = "^p^p"

.Forward = True

.MatchWildcards = True

.Execute Replace:=wdReplaceAll

DoEvents

End With

For Each myPara In ActiveDocument.Paragraphs

myText = myPara.Range.Text

If Len(myText) > 4 Then

Set ch = myPara.Range.Characters(1)

numChars = myPara.Range.Characters.count

Set myEnd = myPara.Range.Characters(numChars)

colNum = ch.HighlightColorIndex Mod 8

If ch.Font.Bold = True Then

myTxt = "qcqc " & Str(colNum + 1) & " = zczc"

Else

myTxt = "qcqc zczc"

End If

If ch.Font.Underline > 0 And colNum > 0 Then

myBit = "\* "

myTxt = Replace(myTxt, " = ", "")

Else

myBit = ""

End If

myPara.Range.InsertBefore myBit & myTxt

If ch.Font.Italic = True Then

myEnd.InsertBefore "qpqp= " & Chr(65 + colNum)

End If

End If

i = i + 1

If i Mod 20 = 0 And Len(myText) > 4 Then

myText = Replace(myText, vbCr, "")

StatusBar = spcs & "Creating queries list: " & myText

End If

DoEvents

Next myPara

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "\\* qcqc(\*)zczc"

.Wrap = wdFindContinue

.Replacement.Text = "\* \1^t"

.Replacement.Highlight = False

.MatchWildcards = True

.Execute Replace:=wdReplaceAll

DoEvents

End With

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "qcqc(\*)zczc"

.Wrap = wdFindContinue

.Replacement.Text = "\1^t"

.Replacement.Highlight = False

.MatchWildcards = True

.Execute Replace:=wdReplaceAll

DoEvents

End With

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "qpqp(\*)^13"

.Replacement.Text = "^t\1^p"

.Replacement.Highlight = False

.MatchWildcards = True

.Execute Replace:=wdReplaceAll

DoEvents

End With

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "= ^$"

.Replacement.Text = ""

.Replacement.Font.Bold = False

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

DoEvents

End With

Set rng = ActiveDocument.Content

rng.Font.Bold = False

rng.Font.Italic = False

rng.Font.DoubleStrikeThrough = False

rng.Font.Underline = False

rng.Font.Color = wdColorBlack

Selection.HomeKey Unit:=wdStory

Selection.TypeText title2 & CR

Set rng = ActiveDocument.Content.Paragraphs(2).Range

If rng.Text = vbCr Then rng.Delete

Set rng = ActiveDocument.Content.Paragraphs(1).Range

rng.Style = ActiveDocument.Styles(wdStyleHeading1)

StatusBar = " "

Options.DefaultHighlightColorIndex = oldColour

lighterColour = wdGray25

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "= ^$"

.Replacement.Text = ""

.Replacement.Font.ColorIndex = lighterColour

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

DoEvents

.Text = "^# ="

.Replacement.Text = ""

.Execute Replace:=wdReplaceAll

DoEvents

End With

Application.ScreenUpdating = True

If doingSeveralMacros = False Then

Debug.Print Timer - timeStart

myTime = (Int(10 \* (Timer - timeStart) / 60) / 10)

Beep

If myTime > 0 Then MsgBox myTime & " minutes"

Else

FUT.Activate

End If

Exit Sub

ReportIt:

Application.ScreenUpdating = True

On Error GoTo 0

Resume

End Sub

Sub ChronologyChecker()

' Paul Beverley - Version 15.04.22

' Copies paragraphs containing date references into a new file

' Case sensitive

myColour\_1 = wdYellow

myWords\_1 = "Monday, Tuesday, Wednesday, Thursday, Friday,"

myWords\_1 = myWords\_1 & "Saturday, Sunday,"

myColour\_2 = wdBrightGreen

myWords\_2 = "January, February, April, June, July, August,"

myWords\_2 = myWords\_2 & "September, October, November, December"

' Case insensitive

myColour\_3 = wdYellow

myWords\_3 = "years old, tomorrow, next day, morning, evening, week, month"

' Case insensitive + whole word

myColour\_4 = wdYellow

myWords\_4 = "age, aged"

' Case sensitive AND whole word

myColour\_5 = wdBrightGreen

myWords\_5 = "May, March, Mon, Tue, Tues, Wed, Weds, Thu, Thurs, Fri, Sat, Sun"

' For years

myColour\_6 = wdTurquoise

multiSpace = 4

myWords\_1 = Replace(myWords\_1, " ", "")

myWords\_1 = Replace("," & myWords\_1 & ",", ",,", ",")

myWords\_2 = Replace(myWords\_2, " ", "")

myWords\_2 = Replace("," & myWords\_2 & ",", ",,", ",")

myWords\_3 = Replace(myWords\_3, " ", "")

myWords\_3 = Replace("," & myWords\_3 & ",", ",,", ",")

myWords\_4 = Replace(myWords\_4, " ", "")

myWords\_4 = Replace("," & myWords\_4 & ",", ",,", ",")

myWords\_5 = Replace(myWords\_5, " ", "")

myWords\_5 = Replace("," & myWords\_5 & ",", ",,", ",")

allWords = Replace(myWords\_1 & myWords\_2 & myWords\_3 & myWords\_4 \_

& myWords\_5, ",,", ",")

For i = 1 To multiSpace

SP = SP & vbCr

Next i

Set rng = ActiveDocument.Content

Documents.Add

For Each myPar In rng.Paragraphs

copyIt = False

For Each wd In myPar.Range.Words

DoEvents

mywd = Trim(wd.Text)

myTest = "," & LCase(mywd) & ","

If InStr(LCase(allWords), myTest) > 0 Then

copyIt = True

Exit For

End If

If Len(mywd) = 4 And LCase(mywd) = UCase(mywd) Then

' Is the first character 1 or 2?

isYear = (InStr("12", Left(mywd, 1)) > 0)

' Are the other three characters digits 0-9?

For i = 2 To 4

j = Asc(Mid(mywd, i)) - 48

If j < 0 Or j > 9 Then isYear = False

Next i

If isYear = True Then

copyIt = True

Exit For

End If

End If

DoEvents

Next wd

If copyIt Then

myPar.Range.Copy

Selection.Paste

Selection.Collapse wdCollapseEnd

Selection.TypeText SP

DoEvents

End If

Next myPar

Selection.HomeKey Unit:=wdStory

Selection.TypeText "Dates context" & vbCr & vbCr

ActiveDocument.Paragraphs(1).Range.Style = wdStyleHeading2

Selection.MoveLeft , 2

oldColour = Options.DefaultHighlightColorIndex

Options.DefaultHighlightColorIndex = myColour\_1

ActiveDocument.Content.HighlightColorIndex = wdNoHighlight

mywd = Split(myWords\_1, ",")

For i = 1 To UBound(mywd) - 1

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = mywd(i)

.Replacement.Text = ""

.Replacement.Highlight = True

.MatchCase = True

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

DoEvents

Next i

mywd = Split(myWords\_2, ",")

Options.DefaultHighlightColorIndex = myColour\_2

For i = 1 To UBound(mywd) - 1

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = mywd(i)

.Replacement.Text = ""

.Replacement.Highlight = True

.MatchCase = True

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

DoEvents

Next i

mywd = Split(myWords\_3, ",")

Options.DefaultHighlightColorIndex = myColour\_3

For i = 1 To UBound(mywd) - 1

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = mywd(i)

.Replacement.Text = ""

.Replacement.Highlight = True

.MatchCase = False

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

DoEvents

Next i

mywd = Split(myWords\_4, ",")

Options.DefaultHighlightColorIndex = myColour\_4

For i = 1 To UBound(mywd) - 1

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = mywd(i)

.Replacement.Text = ""

.Replacement.Highlight = True

.MatchCase = False

.MatchWholeWord = True

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

DoEvents

Next i

mywd = Split(myWords\_5, ",")

Options.DefaultHighlightColorIndex = myColour\_5

For i = 1 To UBound(mywd) - 1

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = mywd(i)

.Replacement.Text = ""

.Replacement.Highlight = True

.MatchCase = True

.MatchWholeWord = True

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

DoEvents

Next i

Options.DefaultHighlightColorIndex = myColour\_6

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "<[12][0-9]{3}>"

.Replacement.Text = ""

.Replacement.Highlight = True

.MatchCase = True

.MatchWholeWord = False

.MatchWildcards = True

.Execute Replace:=wdReplaceAll

End With

Options.DefaultHighlightColorIndex = oldColour

For i = ActiveDocument.Paragraphs.Count To 2 Step -1

Set myPar = ActiveDocument.Paragraphs(i).Range

If Len(myPar.Text) > 1 And myPar.HighlightColorIndex = wdNoHighlight Then

myPar.Select

Selection.MoveEnd , multiSpace

Selection.Delete

End If

DoEvents

Next i

Beep

End Sub

Sub WordsPhrasesInContext()

' Paul Beverley - Version 19.02.18

' Copies paragraphs containing specific names into a new file

myListName = "zzSwitchList"

' myListName = "zzFReditList"

' findWords = "Brown | Jones | Green"

findWords = ""

myBasicColour = wdBrightGreen

returnToText = False

maxWds = 10

CaseSensitive = True

multiSpace = 4

CR = vbCr

For i = 1 To multiSpace

sp = sp & vbCr

Next i

myWords = ""

Set rng = ActiveDocument.Range(Selection.Start, ActiveDocument.Content.End)

parasToEnd = rng.Paragraphs.Count

totParas = ActiveDocument.Paragraphs.Count

If (totParas / parasToEnd) > 10 Then

Selection.Expand wdParagraph

myResponse = MsgBox("Start with this line?", vbQuestion \_

+ vbYesNoCancel, "WordsPhrasesInContext")

If myResponse = vbCancel Then Exit Sub

If myResponse = vbYes Then

Set rng = Selection.Range.Duplicate

rng.Collapse wdCollapseStart

rng.End = ActiveDocument.Content.End

myWords = rng.Text

myWords = Replace(CR & myWords & CR, CR & CR, CR)

myWords = Replace(myWords, CR & CR, CR)

myWords = Replace(myWords, CR & CR, CR)

rng.Collapse wdCollapseStart

rng.Expand wdParagraph

myWd = Split(myWords, CR)

totWords = UBound(myWd) - 1

ReDim myCol(totWords) As Integer

If rng.HighlightColorIndex = wdNoHighlight Then

For i = 1 To totWords

myCol(i) = myBasicColour

Next i

Else

For i = 1 To totWords

myCol(i) = rng.HighlightColorIndex

rng.Collapse wdCollapseEnd

rng.Expand wdParagraph

Next i

End If

Else

Selection.HomeKey Unit:=wdStory

End If

End If

Set mainDoc = ActiveDocument

If myWords = "" Then

gotExternalList = False

' Does a FRedit/Switch list have a "Context words:" line?

For Each myWnd In Application.Windows

thisName = myWnd.Document.Name

If InStr(thisName, myListName) > 0 Then

myWnd.Document.Activate

gotExternalList = True

Exit For

End If

Next myWnd

If gotExternalList = False Then mainDoc.Activate

' If so, load the word and colour arrays from it

DoEvents

Set rng = ActiveDocument.Content

DoEvents

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "Context words:"

.Replacement.Text = ""

.MatchCase = True

.MatchWildcards = False

.Execute

DoEvents

End With

If rng.Find.Found = False Then

mainDoc.Activate

DoEvents

Set rng = ActiveDocument.Content

DoEvents

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "Context words:"

.Replacement.Text = ""

.MatchCase = True

.MatchWildcards = False

.Execute

End With

DoEvents

End If

If rng.Find.Found Then

rng.Expand wdParagraph

rng.Collapse wdCollapseEnd

rng.End = ActiveDocument.Content.End

myWords = rng.Text

myWords = Replace(CR & myWords & CR, CR & CR, CR)

myWords = Replace(myWords, CR & CR, CR)

myWords = Replace(myWords, CR & CR, CR)

rng.Collapse wdCollapseStart

rng.Expand wdParagraph

myWd = Split(myWords, CR)

totWords = UBound(myWd) - 1

ReDim myCol(totWords) As Integer

For i = 1 To totWords

myCol(i) = rng.HighlightColorIndex

rng.Collapse wdCollapseEnd

rng.Expand wdParagraph

Next i

End If

End If

If myWords = "" Then

If Selection.Start = Selection.End Then Selection.Expand wdWord

If findWords = "" Then

myWords = InputBox("Names to find?", "WordsInContext", \_

Trim(Selection))

If myWords = "" Then Exit Sub

Else

myWords = findWords

End If

myWords = Replace(myWords, "| ", "|")

myWords = Replace(myWords, " |", "|")

myWords = Replace(myWords, "|", CR)

myWords = Replace(CR & myWords & CR, CR & CR, CR)

myWords = Replace(myWords, CR & CR, CR)

myWords = Replace(myWords, CR & CR, CR)

myWd = Split(myWords, CR)

totWords = UBound(myWd) - 1

ReDim myCol(totWords) As Integer

For i = 1 To totWords

myCol(i) = myBasicColour

Next i

End If

' In case we're in the external list...

mainDoc.Activate

Set rng = ActiveDocument.Content

Documents.Add

myTestWords = Replace(myWords, ChrW(172), "")

For Each myPara In rng.Paragraphs

parText = myPara.Range.Text

StatusBar = parText

copyIt = False

If Left(parText, 13) = "Context words" Then Exit For

For Each wd In myPara.Range.Words

DoEvents

Set myrange = wd.Duplicate

For i = 1 To maxWds

theseWds = Trim(myrange.Text)

myTest = CR & theseWds & CR

If InStr(LCase(myTestWords), LCase(myTest)) > 0 Then

copyIt = True

Exit For: Exit For

End If

myrange.MoveEnd wdWord, 1

Next i

Next wd

If copyIt Then

myPara.Range.Copy

Selection.Paste

Selection.Collapse wdCollapseEnd

Selection.TypeText sp

DoEvents

End If

Next myPara

Selection.HomeKey Unit:=wdStory

Selection.TypeText "Words/phrases in context" & vbCr & vbCr

ActiveDocument.Paragraphs(1).Range.Style = wdStyleHeading2

Selection.MoveLeft , 2

ActiveDocument.Content.HighlightColorIndex = wdNoHighlight

oldColour = Options.DefaultHighlightColorIndex

myWd = Split(myWords, CR)

For i = 1 To totWords

If Asc(myWd(i)) = 172 Then

myWd(i) = Mid(myWd(i), 2)

CaseSensitive = False

Else

CaseSensitive = True

End If

Options.DefaultHighlightColorIndex = myCol(i)

Set rng = ActiveDocument.Content

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = myWd(i)

.Replacement.Text = ""

.Replacement.Highlight = True

.MatchCase = CaseSensitive

.MatchWildcards = False

.MatchWholeWord = False

.Execute Replace:=wdReplaceAll

End With

Next i

Options.DefaultHighlightColorIndex = oldColour

For i = ActiveDocument.Paragraphs.Count To 2 Step -1

Set myPara = ActiveDocument.Paragraphs(i).Range

If Len(myPara.Text) > 1 And myPara.HighlightColorIndex = wdNoHighlight Then

myPara.Select

Selection.MoveEnd , multiSpace

Selection.Delete

End If

Next i

If returnToText = True Then mainDoc.Activate

Beep

End Sub