**Sub DocAlyseForMac()**

' Version 04.02.20

' Analyses various aspects of a document

showWild = True

' prompts to count number of tests

' cc =50

For i = 1 To 15

spcs = " " & spcs

Next i

Set FUT = ActiveDocument

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

If doingSeveralMacros = False Then

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

"Analyse this document?", vbQuestion \_

+ vbYesNoCancel, "DocAlyseForMac")

If myResponse <> vbYes Then Exit Sub

End If

myTrack = ActiveDocument.TrackRevisions

ActiveDocument.TrackRevisions = False

' Use main file for italic 'et al' count...

myTot = ActiveDocument.range.End

Set rng = ActiveDocument.Content

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "<et al>"

.Font.Italic = True

.Replacement.Text = "^&!"

.Wrap = wdFindContinue

.MatchWildcards = True

.MatchWholeWord = False

.MatchSoundsLike = False

.Execute Replace:=wdReplaceAll

End With

italEtAls = ActiveDocument.range.End - myTot

If italEtAls > 0 Then WordBasic.EditUndo

' ...and superscript degree count

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "[oO0]"

.Font.Superscript = True

.Replacement.Text = "vbvb"

.Replacement.Font.Superscript = False

.MatchWildcards = True

.Execute Replace:=wdReplaceAll

End With

funnyDegrees = (ActiveDocument.range.End - myTot) / 3

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = " vbvb"

.Replacement.Text = "^&!"

.MatchWildcards = True

.Execute Replace:=wdReplaceAll

End With

funnyDegreesSp = ActiveDocument.range.End - myTot - funnyDegrees \* 3

If funnyDegreesSp > 0 Then WordBasic.EditUndo

If funnyDegrees > 0 Then WordBasic.EditUndo

' StatusBar = spcs & myPrompt

DoEvents

Selection.HomeKey Unit:=wdStory

Set rngOld = ActiveDocument.Content

Documents.Add

Set rng = ActiveDocument.Content

rng.FormattedText = rngOld.FormattedText

myEnd = rng.End

Set rng2 = ActiveDocument.Content

rng.Collapse wdCollapseEnd

rng.Text = rng2.Text

Set rng3 = ActiveDocument.Content

rng3.End = myEnd - 1

rng3.Select

Selection.Delete

myRslt = ""

Set rng = ActiveDocument.Content

myTot = ActiveDocument.range.End

CR = vbCr: CR2 = CR & CR

tr = Chr(9) & "0zczc" & CR: sp = " "

Selection.HomeKey Unit:=wdStory

' Ten or 10

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

With rng.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "<ten>"

.Replacement.Text = "^&!"

.MatchWildcards = True

.Execute Replace:=wdReplaceAll

End With

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = " <10>[!,]"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

If i + g > 0 Then myRslt = myRslt & "ten" & vbTab & \_

Trim(Str(i)) & CR & "10" & vbTab & Trim(Str(g)) & CR2

' spelt-out lower-case numbers over nine

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "<[efnst][efghinorvwx]{2,4}ty"

rng.Find.Execute Replace:=wdReplaceAll

a = ActiveDocument.range.End - myTot

If a > 0 Then WordBasic.EditUndo

rng.Find.Text = "<ten>"

rng.Find.Execute Replace:=wdReplaceAll

b = ActiveDocument.range.End - myTot

If b > 0 Then WordBasic.EditUndo

rng.Find.Text = "<eleven>"

rng.Find.Execute Replace:=wdReplaceAll

C = ActiveDocument.range.End - myTot

If C > 0 Then WordBasic.EditUndo

rng.Find.Text = "<twelve>"

rng.Find.Execute Replace:=wdReplaceAll

d = ActiveDocument.range.End - myTot

If d > 0 Then WordBasic.EditUndo

rng.Find.Text = "<[efnst][efghinuorvwx]{2,4}teen>"

rng.Find.Execute Replace:=wdReplaceAll

E = ActiveDocument.range.End - myTot

If E > 0 Then WordBasic.EditUndo

rng.Find.Text = "<hundred>"

rng.Find.Execute Replace:=wdReplaceAll

f = ActiveDocument.range.End - myTot

If f > 0 Then WordBasic.EditUndo

If a + b + C + d + E + f > 0 Then myRslt = myRslt & \_

"spelt-out numbers (11-999)" & vbTab & \_

Trim(Str(a + b + C + d + E + f)) & CR2

' Four-digit numbers

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "[!.]<[0-9]{4}>[!,]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

' take off 20xx dates

rng.Find.Text = "[!.]<20[0-9]{2}>[!,]"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

' take off 13xx to 19xx dates

rng.Find.Text = "[!.]<1[3-9][0-9]{2}>[!,]"

rng.Find.Execute Replace:=wdReplaceAll

k = ActiveDocument.range.End - myTot

If k > 0 Then WordBasic.EditUndo

i = i - g - k

If i < 1 Then i = 1

' Four figs with comma

rng.Find.Text = "[!.]<[0-9],[0-9]{3}>[!,]"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

' Four figs with hard or ordinary space

rng.Find.Text = "[!.]<[0-9][^0160^32][0-9]{3}>[!,]"

rng.Find.Execute Replace:=wdReplaceAll

k = ActiveDocument.range.End - myTot

If k > 0 Then WordBasic.EditUndo

If i + g + k > 0 Then

myRslt = myRslt & "Four-digit numbers:" & CR \_

& "nnnn" & vbTab & Trim(Str(i)) & CR \_

& "n,nnn" & vbTab & Trim(Str(g)) & CR \_

& "n nnn" & vbTab & Trim(Str(k)) & CR

If showWild = True Then myRslt = myRslt & \_

"<[0-9]{4}>" & sp \_

& "<[0-9],[0-9]{3}>" & sp \_

& "<[0-9][^0160^32][0-9]{3}>" & tr

myRslt = myRslt & CR

End If

' Dates with 'mid' in front

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "mid [0-9]{4}"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "mid-[0-9]{4}"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

rng.Find.Text = "mid[0-9]{4}"

rng.Find.Execute Replace:=wdReplaceAll

k = ActiveDocument.range.End - myTot

If k > 0 Then WordBasic.EditUndo

If i + g + k > 0 Then

myRslt = myRslt & "mid 1900(s)" & vbTab \_

& Trim(Str(i)) & CR & "mid-1900(s)" & vbTab & \_

Trim(Str(g)) & CR & "mid1900(s)" & vbTab & \_

Trim(Str(k)) & CR

If showWild = True Then myRslt = myRslt & \_

" mid[0-9]{4}" & tr

myRslt = myRslt & CR

End If

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "mid [0-9]{2}[!0-9]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "mid-[0-9]{2}[!0-9]"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

rng.Find.Text = "mid[0-9]{2}[!0-9]"

rng.Find.Execute Replace:=wdReplaceAll

k = ActiveDocument.range.End - myTot

If k > 0 Then WordBasic.EditUndo

If i + g + k > 0 Then

myRslt = myRslt & "mid 90(s)" & vbTab \_

& Trim(Str(i)) & CR & "mid-90(s)" & vbTab & \_

Trim(Str(g)) & CR & "mid90(s)" & vbTab & \_

Trim(Str(k)) & CR

If showWild = True Then myRslt = myRslt & \_

" mid[0-9]{2}[!0-9]" & tr

myRslt = myRslt & CR

End If

' Serial comma/not serial comma

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "[a-zA-Z\-]@, [a-zA-Z\-]@, and "

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

myRslt = myRslt & "serial comma" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "[a-zA-Z\-]@, [a-zA-Z\-]@ and "

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

myRslt = myRslt & "no serial comma" & vbTab & Trim(Str(i)) & CR

If showWild = True Then myRslt = myRslt & \_

"[a-zA-Z\-]@, [a-zA-Z\-]@, and " & sp & \_

"[a-zA-Z\-]@, [a-zA-Z\-]@ and " & tr

myRslt = myRslt & CR

' hard spaces

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "^s"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

' hard hyphens

rng.Find.Text = "^~"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

myRslt = myRslt & "hard spaces" & vbTab & Trim(Str(i)) \_

& CR & "hard hyphens" & vbTab & Trim(Str(g)) & CR2

' Types of ellipsis

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = ChrW(8230)

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "..."

rng.Find.Execute Replace:=wdReplaceAll

j = ActiveDocument.range.End - myTot

If j > 0 Then WordBasic.EditUndo

rng.Find.Text = ". . ."

rng.Find.Execute Replace:=wdReplaceAll

k = ActiveDocument.range.End - myTot

If k > 0 Then WordBasic.EditUndo

If i + j + k > 0 Then

myRslt = myRslt & "Types of ellipsis:" & CR & \_

"proper ellipsis" & vbTab & Trim(Str(i)) & CR \_

& "triple dots" & vbTab & Trim(Str(j)) & CR \_

& "spaced triple dots" & vbTab & Trim(Str(k)) & CR2

End If

' Ellipsis, etc spacing

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

allChars = ChrW(8230) & "/" & ChrW(8211) \_

& ChrW(8212) & "-"

myNames = "Ellipsis Solidus En dash Em dash Hyphen "

For myGo = 0 To 4

sol = Mid(allChars, myGo + 1, 1)

myName = Trim(Mid(myNames, (9 \* myGo) + 1, 9))

rng.Find.Text = sol

rng.Find.Execute Replace:=wdReplaceAll

t = ActiveDocument.range.End - myTot

If t > 0 Then

WordBasic.EditUndo

rng.Find.Text = " " & sol & " "

rng.Find.Execute Replace:=wdReplaceAll

bth = ActiveDocument.range.End - myTot

If bth > 0 Then WordBasic.EditUndo

rng.Find.Text = "[! ]" & sol & " "

rng.Find.MatchWildcards = True

rng.Find.Execute Replace:=wdReplaceAll

ftr = ActiveDocument.range.End - myTot

If ftr > 0 Then WordBasic.EditUndo

rng.Find.Text = " " & sol & "[! ]"

rng.Find.Execute Replace:=wdReplaceAll

bfr = ActiveDocument.range.End - myTot

If bfr > 0 Then WordBasic.EditUndo

rng.Find.Text = "[! ]" & sol & "[! ]"

rng.Find.Execute Replace:=wdReplaceAll

nthr = ActiveDocument.range.End - myTot

If nthr > 0 Then WordBasic.EditUndo

myRslt = myRslt & myName & " spacing:" & CR & "space before only" \_

& vbTab & Trim(Str(bfr)) & CR & "space after only" & \_

vbTab & Trim(Str(ftr)) & CR & "spaced both ends" & \_

vbTab & Trim(Str(bth)) & CR

If myGo < 4 Then

myRslt = myRslt & "not spaced" & vbTab & Trim(Str(nthr)) & CR2

Else

myRslt = myRslt & CR

End If

myRslt = myRslt & CR

End If

Next myGo

' line breaks

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "^l"

rng.Find.MatchWildcards = False

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

' page breaks

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "^m"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

myRslt = myRslt & "line breaks" & vbTab & Trim(Str(i)) \_

& CR & "page breaks" & vbTab & Trim(Str(g)) & CR2

' Single/double quotes

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = ChrW(8216)

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

singleCurl = i

myRslt = myRslt & "curly open single quote" & vbTab & \_

Trim(Str(i)) & CR

rng.Find.Text = ChrW(8220)

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

myRslt = myRslt & "curly open double quote" & vbTab & \_

Trim(Str(i)) & CR

rng.Find.Text = Chr(39)

rng.Find.MatchWildcards = True

rng.Find.MatchCase = True

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

myRslt = myRslt & "straight single quote" & vbTab & \_

Trim(Str(i)) & CR

rng.Find.Text = Chr(34)

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

myRslt = myRslt & "straight double quote" & vbTab & \_

Trim(Str(i)) & CR2

' etc(.)

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "<etc[!.]"

rng.Find.Execute Replace:=wdReplaceAll

h = ActiveDocument.range.End - myTot

If h > 0 Then WordBasic.EditUndo

rng.Find.Text = "<etc."

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "<etc. [A-Z]"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

rng.Find.Text = "<etc.^13"

rng.Find.Execute Replace:=wdReplaceAll

k = ActiveDocument.range.End - myTot

If k > 0 Then WordBasic.EditUndo

If h + i + g + k > 0 Then myRslt = myRslt & "etc" & \_

vbTab & Trim(Str(h)) & CR & "etc." & vbTab & \_

Trim(Str(i - g - k)) & CR2

' et al(.)

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "<et al[!.]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "<et al."

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

If g + i + italEtAls > 0 Then myRslt = myRslt & "et al." \_

& vbTab & Trim(Str(g)) & CR & "et al (italic, total)" & \_

vbTab & Trim(Str(italEtAls)) & CR & "et al (no dot)" & \_

vbTab & Trim(Str(i)) & CR2

' i.e./ie

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "i.e."

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

myRslt = myRslt

rng.Find.Text = "<ie>"

rng.Find.MatchWildcards = True

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

If i + g > 0 Then myRslt = myRslt & "ie" & vbTab & Trim(Str(g)) & CR \_

& "i.e." & vbTab & Trim(Str(i)) & CR2

' e.g./eg

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "e.g."

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "<eg>"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

If i + g > 0 Then myRslt = myRslt & "eg" & vbTab & Trim(Str(g)) & CR \_

& "e.g." & vbTab & Trim(Str(i)) & CR2

' Initials with surnames

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "<[A-Z]. [A-Z]. [A-Z][a-z]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "<[A-Z][a-z]{2,}, [A-Z]. [A-Z]. "

rng.Find.Execute Replace:=wdReplaceAll

i2 = ActiveDocument.range.End - myTot

If i2 > 0 Then WordBasic.EditUndo

aBit = "J. L. B. Matekoni" & vbTab & Trim(Str(i + i2)) & CR

g = i + i2

rng.Find.Text = "<[A-Z].[A-Z]. [A-Z][a-z]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "<[A-Z][a-z]{2,}, [A-Z].[A-Z]."

rng.Find.Execute Replace:=wdReplaceAll

i2 = ActiveDocument.range.End - myTot

If i2 > 0 Then WordBasic.EditUndo

aBit = aBit & "J.L.B. Matekoni" & vbTab & Trim(Str(i + i2)) & CR

g = g + i + i2

rng.Find.Text = "<[A-Z] [A-Z] [A-Z][a-z]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "<[A-Z][a-z]{2,}, [A-Z] [A-Z] "

rng.Find.Execute Replace:=wdReplaceAll

i2 = ActiveDocument.range.End - myTot

If i2 > 0 Then WordBasic.EditUndo

aBit = aBit & "J L B Matekoni" & vbTab & Trim(Str(i + i2)) & CR

g = g + i + i2

rng.Find.Text = "<[A-Z]{2}> [A-Z][a-z]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "<[A-Z][a-z]{2,}, [A-Z]{2}"

rng.Find.Execute Replace:=wdReplaceAll

i2 = ActiveDocument.range.End - myTot

If i2 > 0 Then WordBasic.EditUndo

aBit = aBit & "JLB Matekoni" & vbTab & Trim(Str(i + i2)) & \_

" (Beware! This can be inflated by, e.g. BBC Enterprises.)" & CR

If showWild = True Then aBit = aBit & "<[A-Z]. [A-Z]. [A-Z][a-z]" \_

& sp & "<[A-Z].[A-Z]. [A-Z][a-z]" & sp & "<[A-Z] [A-Z] [A-Z][a-z]" \_

& sp & "<[A-Z]{2}> [A-Z][a-z]" & tr

If showWild = True Then aBit = aBit & "<[A-Z][a-z]{2,}, [A-Z]. [A-Z]. " \_

& sp & "<[A-Z][a-z]{2,}, [A-Z].[A-Z]." & sp & "<[A-Z][a-z]{2,}, [A-Z] [A-Z] " \_

& sp & "<[A-Z][a-z]{2,}, [A-Z]{2}" & tr

aBit = aBit & CR

If g + i + i2 > 0 Then myRslt = myRslt & aBit

' Convention for page numbers

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "<p. [1-9]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "<pp. [1-9]"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

k = i + g

aBit = "p/pp. 123" & vbTab & Trim(Str(k)) & CR

rng.Find.Text = "<p.[1-9]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "<pp.[1-9]"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

aBit = aBit & "p/pp.123" & vbTab & Trim(Str(i + g)) & CR

k = k + i + g

rng.Find.Text = "<p [1-9]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "<pp [1-9]"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

aBit = aBit & "p/pp 123" & vbTab & Trim(Str(i + g)) & CR

k = k + i + g

rng.Find.Text = "<p[1-9]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "<pp[1-9]"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

aBit = aBit & "p/pp123" & vbTab & Trim(Str(i + g)) & CR

If showWild = True Then aBit = aBit & \_

" <pp[1-9]" & tr

aBit = aBit & CR

If k + i + g > 0 Then myRslt = myRslt & aBit

' Convention for ed/eds/edn

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "<ed>[!.]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "<eds>[!.]"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

rng.Find.Text = "<edn>[!.]"

rng.Find.Execute Replace:=wdReplaceAll

h = ActiveDocument.range.End - myTot

If h > 0 Then WordBasic.EditUndo

rng.Find.Text = "<ed."

rng.Find.Execute Replace:=wdReplaceAll

k = ActiveDocument.range.End - myTot

If k > 0 Then WordBasic.EditUndo

rng.Find.Text = "<eds."

rng.Find.Execute Replace:=wdReplaceAll

m = ActiveDocument.range.End - myTot

If m > 0 Then WordBasic.EditUndo

rng.Find.Text = "<edn."

rng.Find.Execute Replace:=wdReplaceAll

j = ActiveDocument.range.End - myTot

If j > 0 Then WordBasic.EditUndo

If k + m + j + i + g + h > 0 Then myRslt = myRslt \_

& "ed" & vbTab & Trim(Str(i)) & CR & "eds" \_

& vbTab & Trim(Str(g)) & CR & "edn" & vbTab & \_

Trim(Str(h)) & CR & "ed." \_

& vbTab & Trim(Str(k)) & CR & "eds." & vbTab & \_

Trim(Str(m)) & CR & "edn." \_

& vbTab & Trim(Str(j)) & CR2

' Convention for am/pm

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "[1-9][ap]m"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

aBit = "2pm" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "[1-9][ap].m."

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

aBit = aBit & "2p.m." & vbTab & Trim(Str(g)) & CR

rng.Find.Text = "[1-9] [ap]m"

rng.Find.Execute Replace:=wdReplaceAll

k = ActiveDocument.range.End - myTot

If k > 0 Then WordBasic.EditUndo

aBit = aBit & "2 pm" & vbTab & Trim(Str(k)) & CR

rng.Find.Text = "[1-9] [ap].m."

rng.Find.Execute Replace:=wdReplaceAll

h = ActiveDocument.range.End - myTot

If h > 0 Then WordBasic.EditUndo

aBit = aBit & "2 p.m." & vbTab & Trim(Str(h)) & CR2

If k + i + g + h > 0 Then myRslt = myRslt & aBit

' US/UK spelling

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "[bpiv]our[ ,.s]"

rng.Find.Execute Replace:=wdReplaceAll

a = ActiveDocument.range.End - myTot

If a > 0 Then WordBasic.EditUndo

rng.Find.Text = "elling>"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

rng.Find.Text = "elled>"

rng.Find.Execute Replace:=wdReplaceAll

f = ActiveDocument.range.End - myTot

If f > 0 Then WordBasic.EditUndo

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "[bpiv]or[ ,.s]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "rior[ ,.s]"

rng.Find.Execute Replace:=wdReplaceAll

q = ActiveDocument.range.End - myTot

If q > 0 Then WordBasic.EditUndo

rng.Find.Text = "eling>"

rng.Find.Execute Replace:=wdReplaceAll

v = ActiveDocument.range.End - myTot

If v > 0 Then WordBasic.EditUndo

rng.Find.Text = "eled>"

rng.Find.Execute Replace:=wdReplaceAll

k = ActiveDocument.range.End - myTot

If k > 0 Then WordBasic.EditUndo

If a + g + f + i + q + v + k > 0 Then myRslt = \_

myRslt & "UK spelling (appx)" & vbTab & \_

Trim(Str(a + g + f)) & CR & \_

"US spelling (appx)" & vbTab & \_

Trim(Str(i - q + v + k)) & CR & \_

"(For a more accurate count, please use UKUScount.)" & CR2

' US/UK punctuation

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "[a-zA-Z][""" & ChrW(8221) & "][,.]"

rng.Find.Execute Replace:=wdReplaceAll

dfgsdfg = ActiveDocument.range.End

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "[a-zA-Z]['" & ChrW(8217) & "][,.]"

rng.Find.Execute Replace:=wdReplaceAll

j = ActiveDocument.range.End - myTot

If j > 0 Then WordBasic.EditUndo

rng.Find.Text = "[a-zA-Z][,.][""" & ChrW(8221) & "][,.]"

rng.Find.Execute Replace:=wdReplaceAll

k = ActiveDocument.range.End - myTot

If k > 0 Then WordBasic.EditUndo

rng.Find.Text = "[a-zA-Z]['" & ChrW(8217) & "][,.]"

rng.Find.Execute Replace:=wdReplaceAll

m = ActiveDocument.range.End - myTot

If m > 0 Then WordBasic.EditUndo

If i + j + k + m > 0 Then myRslt = myRslt & \_

"UK punctuation (appx)" & vbTab & \_

Trim(Str(i + j)) & CR & "US punctuation (appx)" \_

& vbTab & Trim(Str(k + m)) & CR2

' Initial capital after colon?

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "[a-zA-Z]: [A-Z][a-z]"

rng.Find.Execute Replace:=wdReplaceAll

dfgsdfg = ActiveDocument.range.End

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "[a-zA-Z]: [a-z]"

rng.Find.Execute Replace:=wdReplaceAll

j = ActiveDocument.range.End - myTot

If j > 0 Then WordBasic.EditUndo

If i + j > 0 Then myRslt = myRslt & \_

"Initial capital after colon" & vbTab & \_

Trim(Str(i)) & CR & "Lowercase after colon" \_

& vbTab & Trim(Str(j)) & CR2

' is/iz

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "ise>"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "ise[sd]>"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

rng.Find.Text = "ising>"

rng.Find.Execute Replace:=wdReplaceAll

k = ActiveDocument.range.End - myTot

If k > 0 Then WordBasic.EditUndo

rng.Find.Text = "isation"

rng.Find.Execute Replace:=wdReplaceAll

l = ActiveDocument.range.End - myTot

If l > 0 Then WordBasic.EditUndo

rng.Find.Text = "[armvt]ising"

rng.Find.Execute Replace:=wdReplaceAll

p = ActiveDocument.range.End - myTot

If p > 0 Then WordBasic.EditUndo

rng.Find.Text = "[arvtw]ise"

rng.Find.Execute Replace:=wdReplaceAll

q = ActiveDocument.range.End - myTot

If q > 0 Then WordBasic.EditUndo

rng.Find.Text = "ex[eo]rcis[ei]"

rng.Find.Execute Replace:=wdReplaceAll

r = ActiveDocument.range.End - myTot

If r > 0 Then WordBasic.EditUndo

myRslt = myRslt & "-is- (very appx)" & vbTab & \_

Trim(Str(i + g + k + l - p - q - r)) & CR

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "ize>"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "ize[sd]>"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

rng.Find.Text = "izing>"

rng.Find.Execute Replace:=wdReplaceAll

k = ActiveDocument.range.End - myTot

If k > 0 Then WordBasic.EditUndo

rng.Find.Text = "ization"

rng.Find.Execute Replace:=wdReplaceAll

l = ActiveDocument.range.End - myTot

If l > 0 Then WordBasic.EditUndo

rng.Find.Text = "[Pp]riz[ie]"

rng.Find.Execute Replace:=wdReplaceAll

p = ActiveDocument.range.End - myTot

If p > 0 Then WordBasic.EditUndo

rng.Find.Text = "[Sse]@iz[ie]"

rng.Find.Execute Replace:=wdReplaceAll

q = ActiveDocument.range.End - myTot

If q > 0 Then WordBasic.EditUndo

myRslt = myRslt & "-iz- (very appx)" & vbTab \_

& Trim(Str(i + g + k + l - p - q)) & CR & \_

"(For a more accurate count, please use IZIScount.)" \_

& CR2

' data singular/plural

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "<data is>"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "<data has>"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

rng.Find.Text = "<data was>"

rng.Find.Execute Replace:=wdReplaceAll

h = ActiveDocument.range.End - myTot

If h > 0 Then WordBasic.EditUndo

rng.Find.Text = "<[Tt]his data>"

rng.Find.Execute Replace:=wdReplaceAll

k = ActiveDocument.range.End - myTot

If k > 0 Then WordBasic.EditUndo

myRslt = myRslt

l = i + g + h + k

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "<data are>"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "<data have>"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

rng.Find.Text = "<data were>"

rng.Find.Execute Replace:=wdReplaceAll

h = ActiveDocument.range.End - myTot

If h > 0 Then WordBasic.EditUndo

rng.Find.Text = "<[Tt]hese data>"

rng.Find.Execute Replace:=wdReplaceAll

k = ActiveDocument.range.End - myTot

If k > 0 Then WordBasic.EditUndo

If l + i + h + g + k > 0 Then myRslt = myRslt & \_

"data singular" & \_

vbTab & Trim(Str(l)) & CR & "data plural" & \_

vbTab & Trim(Str(i + g + h + k)) & CR2

' past participle -rnt -elt

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "sp[oi]@lt>"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

rng.Find.Text = "lea[np]t>"

rng.Find.Execute Replace:=wdReplaceAll

h = ActiveDocument.range.End - myTot

If h > 0 Then WordBasic.EditUndo

rng.Find.Text = "[l ][be][ua]rnt>"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "[ds][wpm]elt>"

rng.Find.Execute Replace:=wdReplaceAll

k = ActiveDocument.range.End - myTot

If k > 0 Then WordBasic.EditUndo

rng.Find.Text = "sp[oi]@[l]@ed>"

rng.Find.Execute Replace:=wdReplaceAll

p = ActiveDocument.range.End - myTot

If p > 0 Then WordBasic.EditUndo

rng.Find.Text = "lea[np]ed>"

rng.Find.Execute Replace:=wdReplaceAll

q = ActiveDocument.range.End - myTot

If q > 0 Then WordBasic.EditUndo

rng.Find.Text = "[l ][be][ua]rned>"

rng.Find.Execute Replace:=wdReplaceAll

r = ActiveDocument.range.End - myTot

If r > 0 Then WordBasic.EditUndo

rng.Find.Text = "[ds][wpm]elled>"

rng.Find.Execute Replace:=wdReplaceAll

s = ActiveDocument.range.End - myTot

If g + h + i + k + p + q + r + s > 0 Then myRslt = myRslt & \_

"-rnt -elt" & vbTab & Trim(Str(g + h + i + k)) & CR & \_

"-rned -elled" & vbTab & Trim(Str(p + q + r + s)) & CR2

' amid(st), among(st), while(st)

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "<[Aa]mid>"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

rng.Find.Text = "<[Aa]mong>"

rng.Find.Execute Replace:=wdReplaceAll

h = ActiveDocument.range.End - myTot

If h > 0 Then WordBasic.EditUndo: g = g + h

rng.Find.Text = "<[Ww]hile>"

rng.Find.Execute Replace:=wdReplaceAll

h = ActiveDocument.range.End - myTot

If h > 0 Then WordBasic.EditUndo: g = g + h

rng.Find.Text = "<[Aa]midst>"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "<[Aa]mongst>"

rng.Find.Execute Replace:=wdReplaceAll

h = ActiveDocument.range.End - myTot

If h > 0 Then WordBasic.EditUndo: i = i + h

rng.Find.Text = "<[Ww]hilst>"

rng.Find.Execute Replace:=wdReplaceAll

h = ActiveDocument.range.End - myTot

If h > 0 Then WordBasic.EditUndo: i = i + h

If i + g > 0 Then

myRslt = myRslt & "amid/among/while" & vbTab & Trim(Str(g)) & CR

myRslt = myRslt & "amidst/amongst/whilst" & vbTab & Trim(Str(i)) & CR

If showWild = True Then myRslt = myRslt & "[dgl]st>" & tr

myRslt = myRslt & CR

End If

' fig/figure

aBit = ""

rng.Find.Text = "<fig>[!.]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then

WordBasic.EditUndo

aBit = aBit & "fig" & vbTab & Trim(Str(i)) & CR

End If

rng.Find.Text = "<Fig>[!.]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then

WordBasic.EditUndo

aBit = aBit & "Fig" & vbTab & Trim(Str(i)) & CR

End If

rng.Find.Text = "<fig."

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then

WordBasic.EditUndo

aBit = aBit & "fig." & vbTab & Trim(Str(i)) & CR

End If

rng.Find.Text = "<Fig."

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then

WordBasic.EditUndo

aBit = aBit & "Fig." & vbTab & Trim(Str(i)) & CR

End If

rng.Find.Text = "<figs>[!.]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then

WordBasic.EditUndo

aBit = aBit & "figs" & vbTab & Trim(Str(i)) & CR

End If

rng.Find.Text = "<Figs>[!.]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then

WordBasic.EditUndo

aBit = aBit & "Figs" & vbTab & Trim(Str(i)) & CR

End If

rng.Find.Text = "<figs."

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then

WordBasic.EditUndo

aBit = aBit & "figs." & vbTab & Trim(Str(i)) & CR

End If

rng.Find.Text = "figure [0-9\(]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then

WordBasic.EditUndo

aBit = aBit & "figure" & vbTab & Trim(Str(i)) & CR

End If

rng.Find.Text = "[!.] Figure [0-9\(]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then

WordBasic.EditUndo

aBit = aBit & "Figure" & vbTab & Trim(Str(i)) & CR

End If

If aBit > "" Then myRslt = myRslt & aBit & CR

' Chapter/chapter

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "[!.] Chapter [0-9]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "chapter [0-9]"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

If i + g > 0 Then

myRslt = myRslt & "Chapter (number)" & vbTab & Trim(Str(i)) & CR \_

& "chapter (number)" & vbTab & Trim(Str(g)) & CR

If showWild = True Then myRslt = myRslt & "chapter [0-9]" & tr

myRslt = myRslt & CR

End If

' Section/section

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "[!.] Section [0-9]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "section [0-9]"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

If i + g > 0 Then

myRslt = myRslt & "Section (number)" & vbTab & \_

Trim(Str(i)) & CR & "section (number)" \_

& vbTab & Trim(Str(g)) & CR

If showWild = True Then myRslt = myRslt & \_

"section [0-9]" & tr

myRslt = myRslt & CR

End If

' No./no.

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = " No. [0-9]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = " No [0-9]"

rng.Find.Execute Replace:=wdReplaceAll

j = ActiveDocument.range.End - myTot

If j > 0 Then WordBasic.EditUndo

rng.Find.Text = " no. [0-9]"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

rng.Find.Text = " No.[0-9]"

rng.Find.Execute Replace:=wdReplaceAll

k = ActiveDocument.range.End - myTot

If k > 0 Then WordBasic.EditUndo

rng.Find.Text = " No[0-9]"

rng.Find.Execute Replace:=wdReplaceAll

l = ActiveDocument.range.End - myTot

If l > 0 Then WordBasic.EditUndo

rng.Find.Text = " no.[0-9]"

rng.Find.Execute Replace:=wdReplaceAll

m = ActiveDocument.range.End - myTot

If m > 0 Then WordBasic.EditUndo

If i + j + g + k + l + m > 0 Then

myRslt = myRslt & "No (number)" & vbTab & Trim(Str(i)) \_

& CR & "No. (number)" & vbTab & Trim(Str(j)) & CR \_

& "no. (number)" & vbTab & Trim(Str(g)) & CR

myRslt = myRslt & "No(number)" & vbTab & Trim(Str(k)) \_

& CR & "No.(number)" & vbTab & Trim(Str(l)) & CR \_

& "no.(number)" & vbTab & Trim(Str(m)) & CR2

End If

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = " Vol. [0-9]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = " Vol [0-9]"

rng.Find.Execute Replace:=wdReplaceAll

j = ActiveDocument.range.End - myTot

If j > 0 Then WordBasic.EditUndo

rng.Find.Text = " vol. [0-9]"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

rng.Find.Text = " Vol.[0-9]"

rng.Find.Execute Replace:=wdReplaceAll

k = ActiveDocument.range.End - myTot

If k > 0 Then WordBasic.EditUndo

rng.Find.Text = " Vol[0-9]"

rng.Find.Execute Replace:=wdReplaceAll

l = ActiveDocument.range.End - myTot

If l > 0 Then WordBasic.EditUndo

rng.Find.Text = " vol.[0-9]"

rng.Find.Execute Replace:=wdReplaceAll

m = ActiveDocument.range.End - myTot

If m > 0 Then WordBasic.EditUndo

If i + j + g + k + l + m > 0 Then

myRslt = myRslt & "Vol (number)" & vbTab & Trim(Str(i)) \_

& CR & "Vol. (number)" & vbTab & Trim(Str(j)) & CR \_

& "vol. (number)" & vbTab & Trim(Str(g)) & CR

myRslt = myRslt & "Vol(number)" & vbTab & Trim(Str(k)) \_

& CR & "Vol.(number)" & vbTab & Trim(Str(l)) & CR \_

& "vol.(number)" & vbTab & Trim(Str(m)) & CR

myRslt = myRslt & CR

End If

' equations

aBit = ""

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.MatchWildcards = True

rng.Find.Text = "<eq [0-9]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "eq" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "<eq. [0-9]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "eq." & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "<eqn [0-9]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "eqn" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "<Eqn [0-9]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "Eqn" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "eqns [0-9]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "eqns" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "eqs [0-9]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "eqs" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "<eq \("

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "eq (n.n)" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "<eq. \("

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "eq. (n.n)" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "<Eq. \("

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "Eq. (n.n)" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "<eqn \("

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "eqn (n.n)" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "<Eqn \("

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "Eqn (n.n)" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "eqns \("

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "eqns (n.n)" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "eqs \("

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "eqs" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "Eqs \("

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "Eqs" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "Eqs. \("

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "Eqs." & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "equation \("

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "equation (n.n)" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "[!.] Equation \("

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "Equation (n.n)" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "equations \("

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "equations (n.n)" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "[!.] Equations \("

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "Equations (n.n)" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "equation [0-9]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "equation" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "[!.] Equation [0-9]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "Equation" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "equations [0-9]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "equations" & vbTab & Trim(Str(i)) & CR

rng.Find.Text = "[!.] Equations [0-9]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

aBit = aBit & "Equations" & vbTab & Trim(Str(i)) & CR

If aBit > "" Then myRslt = myRslt & aBit & CR

' units

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "[0-9][^32^160][kKcmM][NgAVm]>"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "[0-9][^32^160][NgAVm]>"

rng.Find.Execute Replace:=wdReplaceAll

j = ActiveDocument.range.End - myTot

If j > 0 Then WordBasic.EditUndo

rng.Find.Text = "[0-9][kKcmM][NgAVm]>"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

rng.Find.Text = "[0-9][NgAVm]>"

rng.Find.Execute Replace:=wdReplaceAll

h = ActiveDocument.range.End - myTot

If h > 0 Then WordBasic.EditUndo

If i + j + g + h > 0 Then

myRslt = myRslt & "spaced units (3 mm)" & vbTab & \_

Trim(Str(i + j)) & CR & "unspaced units (3mm)" \_

& vbTab & Trim(Str(g + h)) & CR

If showWild = True Then myRslt = myRslt & \_

"[0-9][^32^160][kKcmM][NgAVm]>" & sp & \_

"[0-9][kKcmM][NgAVm]>" & sp & \_

"[0-9][NgAVm]>" & sp & \_

"[0-9][kKcmM][NgAVmg]>" & tr

myRslt = myRslt & CR

End If

' percentages

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "[0-9]%"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "[0-9][^32^160]%"

rng.Find.Execute Replace:=wdReplaceAll

j = ActiveDocument.range.End - myTot

If j > 0 Then WordBasic.EditUndo

rng.Find.Text = "[0-9] per cent>"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

rng.Find.Text = "[0-9] percent>"

rng.Find.Execute Replace:=wdReplaceAll

h = ActiveDocument.range.End - myTot

If h > 0 Then WordBasic.EditUndo

rng.Find.Text = "[a-z]{3,} per cent>"

rng.Find.Execute Replace:=wdReplaceAll

k = ActiveDocument.range.End - myTot

If k > 0 Then WordBasic.EditUndo

rng.Find.Text = "[a-z]{3,} percent>"

rng.Find.Execute Replace:=wdReplaceAll

m = ActiveDocument.range.End - myTot

If m > 0 Then WordBasic.EditUndo

If i + j + g + h + k + m > 0 Then

myRslt = myRslt & "unspaced, e.g. 9%" & vbTab & \_

Trim(Str(i)) & CR & "spaced, e.g. 9 %" \_

& vbTab & Trim(Str(j)) & CR & "9 per cent" & vbTab & \_

Trim(Str(g)) & CR & "9 percent" \_

& vbTab & Trim(Str(h)) & CR & "nine per cent" & vbTab & \_

Trim(Str(k)) & CR & "nine percent" \_

& vbTab & Trim(Str(m)) & CR2

End If

' Feet and inches

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

Options.AutoFormatAsYouTypeReplaceQuotes = False

rng.Find.Text = "[0-9]'"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "[0-9]"""

rng.Find.Execute Replace:=wdReplaceAll

j = ActiveDocument.range.End - myTot

If j > 0 Then WordBasic.EditUndo

rng.Find.Text = "[0-9]" & ChrW(8242)

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

rng.Find.Text = "[0-9]" & ChrW(8243)

rng.Find.Execute Replace:=wdReplaceAll

h = ActiveDocument.range.End - myTot

If h > 0 Then WordBasic.EditUndo

If i + j + g + h > 0 Then

myRslt = myRslt & "feet (straight) 9'" & vbTab & \_

Trim(Str(i)) & CR & "inches (straight) 9""" \_

& vbTab & Trim(Str(j)) & CR & "single prime 9" & \_

ChrW(8242) & vbTab & Trim(Str(g)) & CR & \_

"double prime 9" & ChrW(8243) & vbTab & \_

Trim(Str(h)) & CR2

End If

' focus(s)

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "[Ff]ocus[ei]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "[Ff]ocuss[ei]"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

If i + g > 0 Then myRslt = myRslt & "focus..." & \_

vbTab & Trim(Str(i)) & CR \_

& "focuss..." & vbTab & Trim(Str(g)) & CR2

' benefit(t)

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "[Bb]enefit[ei]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "[Bb]enefitt[ei]"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

If i + g > 0 Then myRslt = myRslt & "benefit..." & \_

vbTab & Trim(Str(i)) & CR \_

& "benefitt..." & vbTab & Trim(Str(g)) & CR2

' co(-)oper...

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "[Cc]o-op[ei]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "[Cc]oop[ei]"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

If i + g > 0 Then myRslt = myRslt & "co-oper..." & \_

vbTab & Trim(Str(i)) & CR \_

& "cooper..." & vbTab & Trim(Str(g)) & CR2

' Co-ordin

rng.Find.Text = "[Cc]o-ord[ei]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "[Cc]oord[ei]"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

If i + g > 0 Then myRslt = myRslt & "co-ord..." & \_

vbTab & Trim(Str(i)) & CR \_

& "coord..." & vbTab & Trim(Str(g)) & CR2

' Can't, cannot can not

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "[Cc]an[!a-z]t>"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "[Cc]annot"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

rng.Find.Text = "[Cc]an not"

rng.Find.Execute Replace:=wdReplaceAll

h = ActiveDocument.range.End - myTot

If h > 0 Then WordBasic.EditUndo

If i + h + g > 0 Then myRslt = myRslt & "can't" & \_

vbTab & Trim(Str(i)) & CR \_

& "cannot" & vbTab & Trim(Str(g)) & CR \_

& "can not" & vbTab & Trim(Str(h)) & CR2

' Wasn't, isn't, hasn't etc

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "[owh ][aie]sn[!a-z]t>"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "[owh ][aie]s not"

rng.Find.Execute Replace:=wdReplaceAll

h = ActiveDocument.range.End - myTot

If h > 0 Then WordBasic.EditUndo

If i + h > 0 Then myRslt = myRslt & \_

"wasn't, isn't, hasn't etc" \_

& vbTab & Trim(Str(i)) & CR \_

& "was not, is not, has not etc" & vbTab & \_

Trim(Str(h)) & CR2

' Ok, OK, ok, okay

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = "<OK>"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = "<Ok>"

rng.Find.Execute Replace:=wdReplaceAll

g = ActiveDocument.range.End - myTot

If g > 0 Then WordBasic.EditUndo

rng.Find.Text = "<ok>"

rng.Find.Execute Replace:=wdReplaceAll

h = ActiveDocument.range.End - myTot

If h > 0 Then WordBasic.EditUndo

rng.Find.Text = "<okay>"

rng.Find.Execute Replace:=wdReplaceAll

j = ActiveDocument.range.End - myTot

If j > 0 Then WordBasic.EditUndo

If i + h + g + j > 0 Then myRslt = myRslt & "OK" & \_

vbTab & Trim(Str(i)) & CR \_

& "Ok" & vbTab & Trim(Str(g)) & CR \_

& "ok" & vbTab & Trim(Str(h)) & CR \_

& "okay" & vbTab & Trim(Str(j)) & CR2

' Funny characters

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

rng.Find.Text = ChrW(178)

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

myRslt = myRslt & "funny 'squared' character" \_

& vbTab & Trim(Str(i)) & CR2

rng.Find.Text = "[áÁàÀâÂäÄÃãÅåçÇéÉèÈêÊëËíÍìÌîÎ" & \_

"ñÑóÓòÒôÔöÖõÕøØßúÚùÙûÛüÜýÝÿŸ]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: \_

myRslt = myRslt & "diacritics" & vbTab & Trim(Str(i)) & CR2

rng.Find.Text = "[" & ChrW(161) & ChrW(191) & ChrW(8249) \_

& ChrW(8250) & ChrW(171) & ChrW(187) & "]"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo: myRslt = myRslt & \_

"Continental punctuation" & vbTab & Trim(Str(i)) & CR2

' Ordinary degree symbols

rng.Find.Text = ChrW(176)

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = " " & ChrW(176)

rng.Find.Execute Replace:=wdReplaceAll

isp = ActiveDocument.range.End - myTot

If isp > 0 Then WordBasic.EditUndo

If i > 0 Then myRslt = myRslt & "degree symbols closed" \_

& vbTab & Trim(Str(i - isp)) & CR \_

& "degree symbols spaced" \_

& vbTab & Trim(Str(isp)) & CR2

' Funny degrees

rng.Find.Text = "º"

rng.Find.Execute Replace:=wdReplaceAll

i = ActiveDocument.range.End - myTot

If i > 0 Then WordBasic.EditUndo

rng.Find.Text = " º"

rng.Find.Execute Replace:=wdReplaceAll

isp = ActiveDocument.range.End - myTot

If isp > 0 Then WordBasic.EditUndo

If i + funnyDegrees > 0 Then

myRslt = myRslt & "funny degrees closed" \_

& vbTab & Trim(Str(i + funnyDegrees - isp - \_

funnyDegreesSp)) & CR \_

& "funny degrees spaced" \_

& vbTab & Trim(Str(isp + funnyDegreesSp)) & CR2

End If

appx = ""

If colouredText > 0 Then

If colourOverflow = True Then appx = " (I think)"

myRslt = myRslt & "text in coloured font" \_

& appx & vbTab & Trim(Str(colouredText - 1)) & CR2

End If

If lineBreaks > 0 Then

myRslt = myRslt & "line breaks" \_

& vbTab & Trim(Str(i + lineBreaks)) & CR2

End If

If pageBreaks > 0 Then

myRslt = myRslt & "page breaks" \_

& vbTab & Trim(Str(i + pageBreaks)) & CR2

End If

' Medical bits go in here

myRslt = myRslt & CR

Selection.HomeKey Unit:=wdStory

Selection.TypeText CR & myRslt & CR2

Selection.Start = 0

Selection.Font.Bold = True

Selection.ParagraphFormat.TabStops(CentimetersToPoints(5#)).Position = \_

CentimetersToPoints(5#)

' Grey out the zero lines

' cc =cc - 1

DoEvents

' StatusBar = spcs & "Test number " & cc

With Selection.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "^13([!^13]@)^t0"

.Wrap = wdFindContinue

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

.Replacement.Font.Bold = False

.Replacement.Font.Color = wdColorGray25

.MatchWildcards = True

.Execute Replace:=wdReplaceAll

End With

With Selection.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = "^t^=zczc"

.Wrap = wdFindContinue

.Replacement.Text = ""

.MatchWildcards = False

.Execute Replace:=wdReplaceAll

End With

Set rng = ActiveDocument.Content

With Selection.Find

.ClearFormatting

.Replacement.ClearFormatting

.Text = ""

.Replacement.Text = ""

.MatchWildcards = False

.Execute

End With

Selection.HomeKey Unit:=wdStory

Selection.TypeText "Docalyse" & vbCr

ActiveDocument.Paragraphs(1).Style = \_

ActiveDocument.Styles(wdStyleHeading1)

Selection.HomeKey Unit:=wdStory

If doingSeveralMacros = False Then

Beep

Else

FUT.Activate

End If

End Sub