C:\Users\Tony Lewis\AppData\Roaming\Microsoft\Word\STARTUP\FindEvents.dotm
Author: Tony Lewis
Last saved: July 28, 2012 by Tony Lewis

Macros

AutoExec	LoadEventHandler	3	Load the find/replace event handler when the template is opened
AutoExit	LoadEventHandler	3	Unload the find/replace event handler when the template is closed

LoadEventHandler

```
Option Explicit

' ============================================================================
' Macros in this template are Copyright © Tony Lewis <tlewis@exelana.com> 2012
'
' This program is free software: you can redistribute it and/or modify it
' under the terms of the GNU General Public License as published by the
' Free Software Foundation, either version 3 of the License, or (at your
' option) any later version.

' ============================================================================
' Provide the event handler with access to the Windows 32 API
Private Declare Function EnumWindows Lib "user32" _
    (ByVal lpEnumFunc As Long, ByVal lParam As Long) As Boolean
Private Declare Function GetCurrentProcessId Lib "kernel32" _
    () As Long
Private Declare Function GetWindowText Lib "user32" Alias "GetWindowTextA" _
    (ByVal hWnd As Long, ByVal lpString As String, ByVal cch As Long) As Long
Private Declare Function GetWindowTextLength Lib "user32" Alias "GetWindowTextLengthA" _
    (ByVal hWnd As Long) As Long
Private Declare Function GetWindowThreadProcessId Lib "user32" _
    (ByVal hWnd As Long, lpdwProcessId As Long) As Long

Private evHandler As eventHandler
Dim theCaption As String
Dim thePID As Long
Dim theHWnd As Long

' ============================================================================
' Load the find/replace event handler when the template is opened
Sub AutoExec()
    Set evHandler = New eventHandler
End Sub

' Unload the find/replace event handler when the template is closed
Sub AutoExit()
    Set evHandler = Nothing
End Sub

' ============================================================================
' Find a window with a specific caption
Function cbFindCaption(ByVal hWnd As Long, ByVal lParam As Long) As Boolean
Dim PID As Long
Dim textLen As Long
Dim windowText As String

    GetWindowThreadProcessId hWnd, PID
    If PID = thePID Then
        textLen = GetWindowTextLength(hWnd)
        windowText = Space(textLen)
        GetWindowText hWnd, windowText, textLen + 1
        If windowText = theCaption Then
            theHWnd = hWnd
        End If
    End If
    cbFindCaption = True
End Function

Public Function GetHWndFromCaption(caption As String) As Long
    thePID = GetCurrentProcessId()
    theHWnd = -1
    theCaption = caption
```

```
    EnumWindows AddressOf cbFindCaption, ByVal 0&
    GetHWndFromCaption = theHWnd
End Function
```

eventHandler

```
Option Explicit

' ============================================================================
' Event handler - captures changes to the selection caused by the
' Find and Replace dialog and adjusts the position of the result on the
' screen.
Private Type apiRect
    left As Long
    top As Long
    right As Long
    bottom As Long
End Type

Private Declare Function GetWindowRect Lib "user32" _
    (ByVal hWnd As Long, lpRect As apiRect) As Long

Const debugEvents = False
Private WithEvents app As Word.Application
Private isMoving As Boolean
Private ptHeight As Long
Private ptLeft As Long
Private ptTop As Long
Private ptWidth As Long

Private Sub app_WindowSelectionChange(ByVal Sel As Selection)
Dim hWndFind As Long
Dim rect As apiRect
Dim rng As Range

    If isMoving Then Exit Sub
    hWndFind = GetHWndFromCaption("Find and Replace")
    ' If the Find and Replace dialog is not found, then don't reposition the text on the screen
    If hWndFind = -1 Then Exit Sub

    If debugEvents Then MsgBox "SelectionChange: scroll surrounding paragraphs into view."
    isMoving = True
    If Selection.StoryType = wdTextFrameStory Then
        ' Try to fit the entire text box on the visible part of the screen
        ScrollRange hWndFind, Selection.ShapeRange(1).TextFrame.TextRange
        ' but, if it doesn't fit then scroll the selection onto the screen
        Set rng = Selection.Range
        ActiveWindow.ScrollIntoView rng, True

    ElseIf Selection.StoryType = wdEndnotesStory _
      Or Selection.StoryType = wdFootnotesStory Then
        GetWindowRect hWndFind, rect
        Set rng = Selection.Range
        If RangeAbove(rect, rng) Then
            ' Make sure the selection is below the Find/Replace dialog
            ActiveWindow.SmallScroll up:=5
        End If

    Else
        ' Try to get the previous and next paragraphs on the screen
        Set rng = Selection.Range
        Selection.SetRange rng.Start, rng.Start
        Selection.MoveUp Unit:=wdParagraph, Count:=IIf(Selection.Paragraphs(1).Range.Start < rng.Start, 2, 1)
        ActiveWindow.ScrollIntoView Selection.Range, True
        Selection.MoveDown Unit:=wdParagraph, Count:=2
        Selection.SetRange Selection.Paragraphs(1).Range.End - 1, Selection.Paragraphs(1).Range.End - 1
        Selection.MoveRight
        ActiveWindow.ScrollIntoView Selection.Range, True
        Selection.SetRange rng.Start, rng.End
```

```
            ActiveWindow.ScrollIntoView Selection.Range, True
        End If

        ' Finally, try to fit the selection in the visible part of the screen
        ScrollRange hWndFind, rng
        ActiveWindow.ScrollIntoView Selection.Range, True
        isMoving = False
End Sub

Private Function RangeAbove(rect As apiRect, rng As Range) As Boolean
        RangeAbove = False
        ActiveWindow.GetPoint ptLeft, ptTop, ptWidth, ptHeight, rng
        If ptTop + ptHeight < rect.bottom Then
            RangeAbove = True
        End If
End Function

Private Function RangeOverlaps(rect As apiRect, rng As Range) As Boolean
        RangeOverlaps = False
        ActiveWindow.GetPoint ptLeft, ptTop, ptWidth, ptHeight, rng
        If (rect.top < ptTop And rect.bottom > ptTop _
          Or rect.top < ptTop + ptHeight And rect.bottom > ptTop + ptHeight) Then
            RangeOverlaps = True
        End If
End Function

Private Sub ScrollRange(hWnd As Long, rng As Range)
Dim rect As apiRect
Dim lastTop As Long
Dim overlaps As Boolean

        ActiveWindow.ScrollIntoView rng, True
        GetWindowRect hWnd, rect
        lastTop = -1
        Do
            overlaps = False
            If RangeOverlaps(rect, rng) Then
                ' Don't keep trying if the last attempt didn't make a difference
                If lastTop <> ptTop Then
                    overlaps = True
                    ActiveWindow.SmallScroll up:=1
                End If
            End If
            lastTop = ptTop
        Loop Until Not overlaps
End Sub

Private Sub Class_Initialize()
        If debugEvents Then MsgBox "Load event handler"
        Set app = Word.Application
        isMoving = False
End Sub

Private Sub Class_Terminate()
        If debugEvents Then MsgBox "Unload event handler"
        Set app = Nothing
End Sub
```