# The Macros

The macros have been moved! They are now available via the internet, which means you’re sure that you’re getting the most up-to-date version. Below is the macro you need to use to access them, plus instructions as to how to use this new *MacroFetch* macro, but if you want to get the macro from a Word file, as before, that’s available in the download as TheMacrosAll.

***To download a macro from the website:***

1) In *ComputerTools4Eds* you can just click the link on the name of the macro you want.

2) Use *MacroFetch*, which ‘fetches’ from the website.

***Either:***

You can copy the name of the macro you want – from an email or wherever – into the clipboard, i.e. Ctrl-C.

***or:***

You can type the name of the macro into a Word file.

*N.B. Macro names are case sensitive, so ‘****Fr****edit’ won’t load ‘****FR****edit’.*

In either case, in Word, you can now run the *MacroFetch* macro.

3) When the macro appears in your browser, select all (Ctl-A) and copy (Ctrl-C).

4) Open VBA; select a suitable place (maybe at the very end of the macro area) and paste in your new macro.

**Sub MacroFetch()**

' Paul Beverley - Version 18.02.21

' Fetches the macro named in the clipboard or at the cursor

Set MyData = New DataObject

MyData.GetFromClipboard

On Error Resume Next

theClip = Trim(MyData.GetText)

On Error GoTo 0

clipHasMacro = True

ch1 = Left(theClip, 1)

If LCase(ch1) = ch1 Then clipHasMacro = False

For i = 1 To Len(theClip) - 3

ch = Mid(theClip, i, 1)

If LCase(ch) = UCase(ch) Then

clipHasMacro = False

Exit For

End If

DoEvents

Next i

If clipHasMacro = True And Len(theClip) > 2 Then

myMacroName = theClip

myFolder = Left(myMacroName, 1)

myFullName = "http://www.wordmacrotools.com/macros/" & \_

myFolder & "/" & myMacroName

ActiveDocument.FollowHyperlink Address:=myFullName

End If

Selection.Collapse wdCollapseEnd

Selection.Expand wdWord

If Len(Selection) < 3 Then

Selection.MoveLeft , 3

Selection.Expand wdWord

End If

Do While InStr(ChrW(8217) & "' ", Right(Selection.Text, 1)) > 0

Selection.MoveEnd , -1

DoEvents

Loop

myMacroName = Selection

gotMacro = True

ch1 = Left(myMacroName, 1)

If LCase(ch1) = ch1 Then gotMacro = False

For i = 1 To Len(myMacroName) - 3

ch = Mid(myMacroName, i, 1)

If LCase(ch) = UCase(ch) Then

gotMacro = False

Exit For

End If

DoEvents

Next i

If Len(myMacroName) > 3 And gotMacro = True Then

myFolder = Left(myMacroName, 1)

myFullName = "http://www.wordmacrotools.com/macros/" & \_

myFolder & "/" & myMacroName

ActiveDocument.FollowHyperlink Address:=myFullName

End If

ShowVisualBasicEditor = True

End Sub

## Updating a macro, while keeping its keystroke

1) Click in the macro name and run *MacroFetchUpdate*.

2) Alt-F8 to get to the Macros window and delete the old version of the macro.

3) Click Edit to go into VBA, and Ctrl-V to paste the new version of the macro.

4) Close VBA and run *MacroFetchUpdate* a second time, and it will restore your original keystroke.

**Sub MacroFetchUpdate()**

' Paul Beverley - Version 26.06.22

' Updates an existing macro, preserving the keystroke

Application.ScreenUpdating = False

Documents.Add

Selection.Paste

theClip = ActiveDocument.Content.Text

ActiveDocument.Close SaveChanges:=False

Application.ScreenUpdating = True

If Len(theClip) > 30 Then theClip = "??": ' Impossible macro name

clipHasMacro = True

ch1 = Left(theClip, 1)

If LCase(ch1) = ch1 Then clipHasMacro = False

For i = 1 To Len(theClip) - 3

ch = Mid(theClip, i, 1)

If LCase(ch) = UCase(ch) Then

clipHasMacro = False

Exit For

End If

DoEvents

Next i

theClip = Replace(Trim(theClip), vbCr, "")

Debug.Print Trim(theClip) & "|"

If clipHasMacro = True And Len(Trim(theClip)) > 2 Then

myMacroName = theClip

Else

Selection.Collapse wdCollapseStart

Selection.Expand wdWord

If Len(Selection) < 3 Then

Selection.MoveLeft , 3

Selection.Expand wdWord

End If

Do While InStr(ChrW(8217) & "' ", Right(Selection.Text, 1)) > 0

Selection.MoveEnd , -1

DoEvents

Loop

myMacroName = Selection

End If

Selection.Expand wdParagraph

If InStr(Selection, "+") > 0 Then GoTo restoreKeystroke

' Attempt to read keystroke

myKeyString = ""

For Each kb In KeyBindings

If kb.KeyCategory = 2 Then ' It's a Macro

cmd = kb.Command

gottit = (InStr(cmd, "." & myMacroName) > 0)

If Left(cmd, 6) = "Normal" And gottit And \_

Right(cmd, Len(myMacroName)) = myMacroName Then

myKeyString = kb.KeyString

Exit For

End If

End If

Next kb

Debug.Print myKeyString

If myKeyString > "" Then

Documents.Add

Selection.TypeText Text:=myMacroName & vbTab & myKeyString & vbCr

Selection.HomeKey Unit:=wdStory

myResponse = MsgBox("Now, you need to..." \_

& vbCr & vbCr & "1) Copy the text of the new version" \_

& vbCr & vbCr & "2) Delete the existing copy of the macro" \_

& vbCr & vbCr & "3) Paste the new version into Visual Basic (VBA)" \_

& vbCr & vbCr & vbCr & " Finally, rerun this macro", \_

vbOKOnly, "MacroFetchUpdate")

Else

MsgBox "Macro " & myMacroName & "has no keystroke assigned to it."

End If

Call MacroFetch

Exit Sub

restoreKeystroke:

dotPos = InStr(Selection, ".")

If dotPos > 0 Then Selection.MoveStart , dotPos

dotPos = InStr(Selection, ".")

If dotPos > 0 Then Selection.MoveStart , dotPos

tabPos = InStr(Selection, vbTab)

Selection.MoveEnd , -1

myKeyString = Mid(Selection, tabPos + 1)

myResponse = MsgBox("Set keystroke for macro: " & myMacroName & \_

vbCr & vbCr & " to: " & myKeyString & "?", \_

vbQuestion + vbYesNoCancel, "MacroFetchUpdate")

If myResponse = vbYes Then

ks = myKeyString

kCode = 0

If InStr(ks, "Ctrl+") > 0 Then

kCode = kCode + 512

ks = Replace(ks, "Ctrl+", "")

End If

If InStr(ks, "Alt+") > 0 Then

kCode = kCode + 1024

ks = Replace(ks, "Alt+", "")

End If

If InStr(ks, "Shift+") > 0 Then

kCode = kCode + 256

ks = Replace(ks, "Shift+", "")

End If

' Ordinary capital letters

If ks Like "[A-Z]" Then

aCode = Asc(UCase(ks))

ks = ""

End If

' Ordinary numbers

If ks Like "[0-9]" Then

aCode = Asc(UCase(ks))

ks = ""

End If

' F keys

If Left(ks, 1) = "F" And Len(ks) > 1 Then

aCode = 111 + Val(Replace(ks, "F", ""))

ks = ""

End If

Select Case ks

Case "!": aCode = wdKey1: kCode = kCode + 256 ' Shifted number 1

Case """": aCode = wdKey2: kCode = kCode + 256 ' Shifted number 2

Case ChrW(163): aCode = wdKey3: kCode = kCode + 256 ' Shifted number 3

Case "$": aCode = wdKey4: kCode = kCode + 256 ' Shifted number 4

Case "%": aCode = wdKey5: kCode = kCode + 256 ' Shifted number 5

Case "^": aCode = wdKey6: kCode = kCode + 256 ' Shifted number 6

Case "&": aCode = wdKey7: kCode = kCode + 256 ' Shifted number 7

Case "\*": aCode = wdKey8: kCode = kCode + 256 ' Shifted number 8

Case "(": aCode = wdKey9: kCode = kCode + 256 ' Shifted number 9

Case ")": aCode = wdKey0: kCode = kCode + 256 ' Shifted number 0

Case "'": aCode = 192

Case "-": aCode = wdKeyHyphen ' The - key

Case "#": aCode = 222

Case ",": aCode = wdKeyComma ' The , key

Case ".": aCode = wdKeyPeriod ' The key

Case "/": aCode = wdKeySlash

Case ":": aCode = wdKeySemiColon: kCode = kCode + 256

Case ";": aCode = wdKeySemiColon ' The ; key

Case "?": aCode = wdKeySlash: kCode = kCode + 256

Case "@": aCode = 192: kCode = kCode + 256

Case "[": aCode = wdKeyOpenSquareBrace ' The ] key

Case "\": aCode = wdKeyBackSlash ' The \ key

Case "]": aCode = wdKeyCloseSquareBrace ' The ] key

Case "\_": aCode = wdKeyHyphen ' Does it need? kCode = kCode + 256

Case "`": aCode = 223

Case "{": aCode = wdKeyOpenSquareBrace: kCode = kCode + 256

Case "}": aCode = wdKeyCloseSquareBrace: kCode = kCode + 256

Case "~": aCode = 222: kCode = kCode + 256

Case "+": aCode = wdKeyEquals ' kCode = kCode + 256

Case "<": aCode = wdKeyComma: kCode = kCode + 256

Case "=": aCode = wdKeyEquals ' The = key

Case ">": aCode = wdKeyPeriod: kCode = kCode + 256

Case "Backspace": aCode = wdKeyBackspace ' The Backspace key

Case "Clear (Num 5)": aCode = wdKeyNumeric5: kCode = kCode + 256

Case "Delete": aCode = wdKeyDelete ' The Delete key

Case "Down": aCode = 40 ' The Down Cursor key

Case "End": aCode = 35 ' The EndD key

Case "Home": aCode = wdKeyHome ' The HOME key

Case "Insert": aCode = wdKeyInsert ' The Insert key

Case "Left": aCode = 37 ' The Left Cursor key

Case "Num -": aCode = wdKeyNumericSubtract ' The - key on the keypad

Case "Num \*": aCode = wdKeyNumericMultiply ' The \* key on the keypad

Case "Num /": aCode = wdKeyNumericDivide ' The / key on the keypad

Case "Num +": aCode = wdKeyNumericAdd ' The + key on the keypad

Case "Num 0": aCode = wdKeyNumeric0 ' The 0 key on the keypad

Case "Num 1": aCode = wdKeyNumeric1 ' The 1 key on the keypad

Case "Num 2": aCode = wdKeyNumeric2 ' The 2 key on the keypad

Case "Num 3": aCode = wdKeyNumeric3 ' The 3 key on the keypad

Case "Num 4": aCode = wdKeyNumeric4 ' The 4 key on the keypad

Case "Num 5": aCode = wdKeyNumeric5 ' The 5 key on the keypad

Case "Num 6": aCode = wdKeyNumeric6 ' The 6 key on the keypad

Case "Num 7": aCode = wdKeyNumeric7 ' The 7 key on the keypad

Case "Num 8": aCode = wdKeyNumeric8 ' The 8 key on the keypad

Case "Num 9": aCode = wdKeyNumeric9 ' The 9 key on the keypad

Case "Page Down": aCode = 34 ' The Page Down key

Case "Page Up": aCode = 33 ' The Page Up key

Case "Return": aCode = wdKeyReturn ' The Enter or Return key

Case "Right": aCode = 39 ' The Right Cursor key

Case "Space": aCode = wdKeySpacebar ' The Spacebar key

Case "Up": aCode = 38 ' The Up Cursor key

End Select

kCode = kCode + aCode

hexCode = Replace(Hex(kCode), "FFFF", "")

KeyBindings.Add KeyCode:=kCode, \_

KeyCategory:=wdKeyCategoryMacro, Command:=myMacroName

Beep

MsgBox "Keystroke restored"

Else

Beep

End If

End Sub