

02 Downloading and running macros

Downloading macros from the internet

The macros in the TheMacroAll Word file are also available individually from the internet. If you want, say *CaseNextChar*, the URL is:

<http://www.wordmacrotools.com/macros/C/CaseNextChar>

or for *FRedit*, it's:

<http://www.wordmacrotools.com/macros/F/FRedit>

But these URLs are available as single-click links, and also via macro-fetching macros.

Macros for downloading macros

There are three macros created for this purpose, and all three can specify the macro name in two ways. The point is that the macro names must be correctly 'cased', so if you want *CaseNextChar*, it's no good trying to use 'Casenextchar'; this will fail to find the macro you're looking for.

So, rather than having to type out the name, each of the downloading macros starts by looking at the text currently in the clipboard. If you had selected and copied a macro name, say from an email or a web page, you would just click in a Word file – any Word file – and run the relevant macro.

If the clipboard doesn't contain a single word that looks like a macro name, then the fetching macro looks instead at the word at the cursor in the currently open file, and assumes that's the name of the macro you're wanting to download.

The first macro, *MacroFetch*, creates the relevant URL and launches it to our website.

The second macro, *MacroFetchUpdate*, is for use if you're wanting to update a macro that has a keystroke already allocated to it. So, as with *MacroFetch*, have the macro name in the clipboard or as a word at the cursor, and run the macro. It will store the name of the macro and its keystroke in the current Word file.

Next select and copy the macro; return to your Word file and use the Macros window (Alt-F8 or Option-F8) and delete the current version of the macro that you're upgrading. Go into VBA and paste in the new version and then run *MacroFetchUpdate* again. This time, instead of fetching a macro, the macro will read the keystroke from inside the Word file and allocate it to the new version of the macro.

The third macro, *MacroNameToLink*, takes the macro name, as before, and simply creates the relevant URL and puts it into the clipboard, from where it is ready to be pasted for whatever purpose. So, you could paste it into an email to give to someone, or paste it into a web browser to examine the macro.

```
Sub MacroFetch()
```

```
Sub MacroFetchUpdate()
```

```
Sub MacroNameToLink()
```

Running the macros

There are basically three ways you can run macros:

1. from the Macros dialogue box
2. by adding an icon to the toolbar at the top of the Word screen
3. by pressing a particular key combination.

I use the dialogue box for those macros that I use very rarely, but I never use icons (2). I run 99% of my macros from keystrokes because it's so much faster than using icons. Once you get more than a small number of icons for macros, it just becomes impractical to use icons.

"But I can't remember keystrokes!" OK, let me ask you a question: do you have to remember where the gears are in your car? If you do something often enough, it becomes automatic. What's more, there's a pattern to the gears, which helps. So I suggest that you make a 'pattern' for your keystrokes. Use keys that have some significance to you, and/or use various key combinations with the F-keys, and put a strip of card with the macro names written on it. For very frequently used keystrokes, I suggest that you use a key combination that you can press just with your left hand, and/or keys on the numeric keypad with your right hand – the numeric pad means that it's less far for you to move your hand away from the mouse.

"What keystrokes should I use?"

I often get asked this, but keystrokes are only any use if you can remember them (obviously!), so they really have to be your choice. And there's one more factor to stop me *telling* you what to use: on some computers, some special keystrokes have been taken over by certain other software or hardware features, e.g. video cards.

So, how would I advise you on what keystrokes to choose?

First, for any keystrokes you use regularly, the choice doesn't have to be of a significant letter because you will remember them anyway. So my advice for those you do use regularly, is to assign them to easy left-hand-only movements, which will be fast, and you can leave your right hand on the mouse. So, for me, *CaseNextChar* is Alt-S, and *MultiSwitch* is Alt-Q.

After that, for me, other commonly used macros go on Alt/Ctrl/Shift combined with keys on the numeric keypad, e.g. *HighlightPlus (Minus)* are Ctrl-Alt-Num+ (Num-), and *ColourPlus (Minus)* are Ctrl-Shift-Num+ (Num-)

The other thing I do is **link** the keystrokes, so as *CaseNextChar* is Alt-S, *CaseNextWord* is Ctrl-Alt-S.

Then I use Ctrl-Alt-Q for *SwapWords*, Ctrl-Shift-Q for *SwapCharacters*, and Shift-Alt-Q for *SwapPreviousCharacters*.

Then Ctrl-Alt-. is *FullPoint*, Ctrl-Alt-, is *Comma*, Ctrl-Alt-; is *Semicolon* etc.

And Ctrl-Alt-Shift-' is *PunctuationToSingleQuote*, Ctrl-Alt-Shift-" is *PunctuationToDoubleQuote*, etc.

For those macros you don't use very often, my rule is, if you can't remember the allocated keystroke, run *CustomKeys* and, in the 'Press new key' slot, type the keystroke you **think** is allocated to the macro you want to run. If it is **not** allocated to that keystroke, then change it so that it **is!** In other words, that keystroke was intuitive to you, and so that's the keystroke you're most likely to remember, so use it.

Allocating a keystroke (Word 2013 and 2010)

(Video: youtu.be/XXs6z-QhzPw)

1. Right-click on a blank part of the ribbon and click on 'Customize the Ribbon'.
2. Underneath the left-hand column, below the scrollable window, it says 'Keyboard shortcuts: Customize' Click the 'Customize' button next to it.
3. In the Customize Keyboard window that appears, in the left-hand list (Categories), find 'Macros' (pressing 'm' on the keyboard, twice, will get you there quickly).
4. The right-hand list becomes (not surprisingly) 'Macros'. Select your chosen macro name.
5. Click in 'Press new shortcut key' and do just that: press the keyboard shortcut that you want to associate with this macro.
6. To the left of that box is a 'Current keys' box. This box shows whether that macro already has a keystroke assigned to it. Also, immediately under that box is a line telling you whether the keystroke you pressed is currently assigned to something else – another macro or a Word command or a special character – or whether it is currently 'Unassigned'.

7. If you're happy that you want this keystroke to be uniquely linked to your selected macro then click the 'Assign' button.

Allocating a keystroke (Word 2011/365 – Mac)

1. On the Tools menu, click 'Customize Keyboard'.
2. In the Customize Keyboard window that appears, in the left-hand list (Categories), find 'Macros' (pressing 'm' on the keyboard, twice, will get you there quickly).
3. The right-hand list becomes a list of the macros that are installed. Select your chosen macro name.
4. Click in 'Press new keyboard shortcut' and do just that: press the keyboard shortcut that you want to associate with this macro.
5. Above that box is a 'Current keys' box. This box shows whether that macro already has a keystroke assigned to it. Also, immediately under the 'new keyboard shortcut' box is a line telling you whether the keystroke you pressed is currently assigned to something else - another macro or a Word command or a special character – or whether it is currently 'Unassigned'.
6. If you're happy that you want this keystroke to be uniquely linked to your selected macro then click the 'Assign' button.

Allocating a keystroke (Word 2007)

1. Click the 'Customize Quick Access Toolbar' menu – the little down-arrow at the right-hand end of the Quick Access Toolbar (QAT).
2. Choose 'More Commands'.
3. At 'Keyboard Shortcuts' at the bottom of the box, click 'Customize'.
4. In the left-hand list (Categories), select 'Macros' (pressing 'm' on the keyboard, twice, will get you there quickly).
5. Click in 'Press new shortcut key' and do just that: press the keyboard shortcut that you want to associate with this macro.
6. To the left of that box is a 'Current keys' box. This will show whether that macro already has a keystroke assigned to it. Also, immediately under that box is a line telling you whether the keystroke you selected is currently assigned to something else – another macro or a Word command or a special character – or whether it is 'Unassigned'.
7. If you're happy that you want this keystroke to be uniquely linked to your selected macro then click the 'Assign' button.
8. Then 'Close', and your keystroke is ready to use.

Allocating a keystroke (Word 2002/3)

1. Open the 'Tools–Customize' tab.
2. Click 'Keyboard'.
3. Then, in the left-hand list (Categories), select 'Macros' (pressing 'm' on the keyboard, twice, will get you there quickly).
4. The right-hand list becomes (not surprisingly) 'Macros'.
5. Select the macro name.
6. Click in the 'Press new shortcut key' box.
7. Press the keystroke you want to use.
8. Just below that box, a line will appear saying 'Currently assigned to:' and, hopefully, '[unassigned]'.
9. If it is already assigned to another function within Word, you'll have to decide if it is a function that you would want to use via a keystroke and, if so, choose a different keystroke.
10. Click 'Assign'.
11. Then 'Close', and your keystroke is ready to use.

Tip – using the Customize keyboard dialog

This dialogue box is also useful where

- a. you can't remember which keystroke you have used for a given macro
- b. you can't remember the macro name for a keystroke that you already use.

For (b), just click in the 'Press new shortcut key' box, press the relevant keystroke and look in the 'Currently assigned to:' line.

Adding icons (Word 2007/2010)

1. Right-click on a blank space on the screen's toolbar. This brings up the Quick Access Toolbar (QAT)
2. From the QAT, click the 'Customize Quick Access Toolbar' menu.
3. Choose 'More Commands'.
4. (In Word 2010 only, you also now need to click on 'Customize Ribbon'.)
5. In the 'Choose Commands From' list, select 'Macros'.
6. Select 'Normal.NewMacros.<MacroName>' from the list below.
7. Click 'Add'.
8. The macro will appear at the bottom of the QAT list on the right-hand side.
9. Click 'Modify' (under the QAT list).
10. Choose a symbol for your macro.
11. In 'Display Name', shorten 'Normal.NewMacros.MyNewMacro' down to 'MacroName'.
12. Click OK.
13. Use the up/down arrows on the left of the QAT list to move your MacroName symbol to the desired location on the QAT.
14. Click OK.

Adding icons (Word 2002/3)

1. Open the 'Tools->Customize' tab.
2. Select the 'Commands' tab.
3. In the left-hand list (Categories), select 'Macros'.
4. In the right-hand list find 'Normal.NewMacros.<MacroName>'.
5. Drag it up to the toolbar.
6. The cursor will have an 'x' in it, but it will turn into a '+' when you are over a bit of the bar where you are permitted to drop it.

If you want to customise the appearance of the macro icon, do the following:

1. With the 'Customize' box open, right-click on your macro.
2. Click 'Default Style'.
3. Right-click again.
4. Choose either an existing icon from 'Change Button Image'.
5. Or create your own button from 'Edit Button Image'.

Updating macros

If you have a macro that you are already using, and you hear that there's a more up-to-date version, how do you make the upgrade? The important thing to remember is *not* to delete the whole of the macro, from

```
Sub Something()  
to  
End Sub
```

If you just delete the old version of the macro, the associated icon and/or keystroke will be lost and you will have to set it up again.

Instead, just delete the 'meat' of the macro, leaving the Sub and End Sub lines, for example:

```
Sub Citechecker()  
  
End Sub
```

Then copy the 'meat' of the new macro, i.e. not the Sub and End Sub lines, and paste it in the space that you have left for it. (Don't worry about there being extra blank lines – they are totally irrelevant to the working of the macro.)

What happens when things go wrong?

(Video: youtu.be/AY6B-IkLEN8)

I'll give you a general description of what to do to report an error to me, and then I'll list a few errors that sometimes occur, giving you a suggestion of the possible cause. *Oh, but first...*

A general suggestion

If you get errors with some of the macros – especially all the ...*Allyse* macros – it may be worth creating a text-only version of your file and then running the macro on that copy.

One way is to use:

Ctrl-A, Ctrl-C, create a new file, Paste as Pure Text

but to be sure that you're getting **all** the text – including what's in the foot/endnotes and text boxes – you can use the macro *CopyTextSimple*. (Better still, especially for large files, use *CopyTextVerySimple*.)

“Word has crashed!” – but has it?

There are times when Word seems to have crashed – but it might not have. Just because it claims that it's “Not responding” it doesn't mean that it has crashed. Try to be patient – go and make a cup of tea, perhaps, especially if you realise that you had *omitted to save the file* (bad idea!) before you ran that macro.

But if it still seems to have crashed (e.g. the red ‘X’ box at the top right of the Word window is only glowing dull red), you might still be able to regain control of Word (and thus to save that unsaved file!).

One trick you can try (on PCs only, sorry, unless Mac users can tell us your equivalent action) is to open the Windows Task Manager by pressing Ctrl-Alt-Delete. The Task Manager display will show you whether Word is still active. *Whether it is or not*, do NOT click on End Task. Click Cancel, and then **either** you will be able to click in the Word window and you'll have control back, **or** you'll have to click the dull red close box, and Word will do its best to save your unsaved file(s).

How to respond to – and report to WMT – an error

Sometimes, when you try to run a macro, it generates an error, and Visual Basic (VBA) asks you what you want to do, offering you:

End, Debug, Help.

Ironically, the least helpful of these is to click ‘Help’. Don't bother.

If you just want to give up altogether and ignore the idea of using that macro, you can click on End.

To find out what went wrong – perhaps so that you can report the error to WMT – the first thing to do is to make a note of how VBA describes the error. Here's an example:

*Runtime error ‘5174’:
This file could not be found.*

MS Word won't let you copy and paste the error message, but you could perhaps go over to your email software or to a text editor, such as NotePad, and start to compose an email to WMT, typing in this error message.

Next, click on Debug. Debugging is a technique that programmers use to try to work out what has gone wrong with a program. This will take you into VBA with one of the lines of the macro highlighted in yellow, maybe looking something like this:

```
If gottaList = False Then
  Documents.Open dirName & listName
Else
  listDoc.Activate
```

End If

Make a note of the line so that you can report it to us. However, this time, you *can* do it by selecting a bit of the macro, either side of the yellow line, copying it, and then pasting it into a Word file (or your email), where it will appear as ordinary text. But please explain exactly **which** line was actually highlighted in yellow – this is important if you want us to correct the problem.

Next, you have to stop the debugging process, or ‘reset’ VBA. You do this by clicking the Reset button on VBA’s top tool bar. Look for the set of three icons – as on an AV device: Play, Pause and Stop. The ‘Reset’ button is the square block, as used for ‘Stop’ on an AV player.

Send us that information, and we’ll see what we can work out.

(If you **don’t** stop the debugging process and simply go back into Word, all will **seem** to be OK. However, when you later try to run another macro, it will generate the error: ‘*Can’t execute in break mode*’. You then have to click ‘OK’, select the VBA window and click the ‘Reset’ icon, as mentioned above.)

Some possible errors and their possible causes

“**Variable not defined**” – Search in VBA amongst your macros, and see if there’s a line saying `Option Explicit`. If so, put an apostrophe in front of it, to disable it. This won’t harm the operation of any of the other macros.

“**The Find What text contains a Pattern Match expression which is not valid.**” – There are various reasons for this. In general, just report it to me, as above.

However, but if you’re using any of the *XyzAllyse* macros, such as *DocAllyse*, then it’s likely to be a problem with what’s called the ‘list separator’ used in the operating system of your computer. This is especially likely if you’ve got a computer set up for mainland Europe or South Africa. Here are my standard instructions:

The ‘list separator’ used within Word needs to be a comma, not a semicolon.

However, this is not a Word option, rather it’s an operating system option.

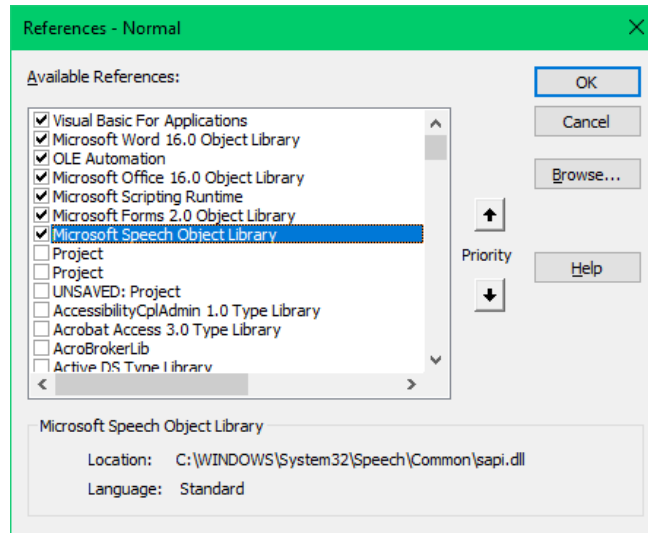
So, on Windows 7, 8.1 and 10, it is in the Control Panel under ‘Clock Language and Region’ and then ‘Region’ and then ‘Additional settings’ (which is a button near the bottom of the Region window). In ‘Additional settings’, the fourth from the bottom is ‘List separator’. Change it to a comma and click OK.

“**Compile error: ambiguous name detected: BlahBlah**” (where ‘BlahBlah’ is the name of a macro) – This means that, in pasting an extra macro into VBA, you’ve ended up with *two* copies of the macro *BlahBlah*. So the solution is to delete one of them. How? You do it in Visual Basic, and you do it very carefully, making sure to delete a whole macro from ‘Sub’ to ‘End Sub’, inclusive.

Note: You can select a whole macro automatically, by double-clicking. However, you need to know *where* to double click: you do so in the 2 mm-wide white margin, between the actual words of the macro and the light-grey vertical strip that looks like (but isn’t) a vertical scroll bar (which is the vertical bar on the right, as with all application windows).

“**User-defined type not defined**” – If you get this error message, try the following: go into Visual Basic and click on: Tools – References.

It **should** look something like this:



In particular, check that you've got both Microsoft Scripting Runtime and Microsoft Forms 2.0 Object Library. Or maybe others are missing from the list of ticked items?

If not, look down through the list of unticked items and see if you can find either or both of them. If so, tick them, click OK, close VBA and then [b]close and re-open Word[/b].

Does that do the trick? (If not, try a complete restart of the computer.)

“Compile error: procedure too long” – This normally only occurs if you're trying to use DocAlyse. And only on a Mac. So if you get this error, try using the macro, [DocAlyseForMac](#). If even that gives the same error, there is a solution: try [DocAlyseForThinMacs](#) and if that fails, there's finally a [DocAlyseForVeryThinMacs](#).