**Macros from Square One for PC**

This document was created to help editors who have never used macros to now get started. Step by step it will teach you the essential skills for using macros in your daily editing. It is focused on the macros created by Paul Beverley.

**How to use this document:** Read through the introductory material in the first section. Then roll your sleeves up and get ready to start learning and practising the new skills. You may want to get together with a couple of other editors to learn together; two or three heads are often better than one.

If you want to **share this document** with a colleague (please do!), it would be best share the download link, as this will ensure they have the most up-to-date version (**Spanish language** versions are now available for [PC](http://www.archivepub.co.uk/documents/MSO_PC_ES) and [Mac](http://www.archivepub.co.uk/documents/MSO_Mac_ES)):

PC: [www.archivepub.co.uk/documents/MSO\_PC](http://www.archivepub.co.uk/documents/MSO_PC)

Mac: [www.archivepub.co.uk/documents/MSO\_Mac](http://www.archivepub.co.uk/documents/MSO_Mac)

Pace yourself: There is a lot of information here to absorb. You don’t need to work through all the steps at once. Take a break between sections. The three ‘Next steps’ sections at the end provide you with learning options for the next few months. The more time you can give yourself to practise a few macros at a time, the more successful you will be.

A note on style: Although this document follows UK style conventions, because editors around the world use macros, examples are shown using other styles as well.

# Macros – the essential information

First, why do editors use macros?

• Macros increase our **efficiency** by speeding up routine edits.

• Macros increase our **accuracy** by helping us to catch more mistakes.

• Macros **reduce** the chances of **repetitive strain injuries** by reducing how much we have to press the mouse and keyboard.

*Now a definition:* A macro is a small computer program, written in the Visual Basic for Applications (VBA) language, that you install in Word. But that’s a bit technical sounding, isn’t it?

*Here’s a simpler definition:* A macro is a tool to help you edit. As editors, we use many tools: dictionaries, style guides, Word’s spellchecking and Track Change features, online reference sources, and perhaps PerfectIt. Macros provide further tools; they can even help you use some of the other tools more efficiently.

***Important:*** *You don’t need any programming knowledge to use macros*. The main ‘technical’ skill you need is to copy the text of a macro (i.e., the computer program) into the VBA area of Word. This can feel intimidating because getting to the VBA area requires multiple steps. But don’t worry. We will walk you through it one step at a time.

**Using macros requires you to learn some new skills:**

• How to install macros in Word

• How to assign keyboard shortcuts to your macros

• How to find the macros best suited for the type of editing you do

The rest of this document will teach you these skills.

Lastly, a bit of macro anatomy.

***Again, you don’t need any programming skills!*** But when you are pasting the macro text into Word, do it slowly and carefully, or your macro may not work.

You need to know how to identify the first and last lines of a macro. Here’s the code for a macro called SwapCharacters:

Sub SwapCharacters()

Selection.MoveEnd , 1

Selection.Cut

Selection.MoveLeft , 1

Selection.Paste

End Sub

The first line of a macro always contains the word Sub followed by the macro name, SwapCharacters in this example, and ending with (). And the last line of a macro will always be End Sub. That’s all you need to know. (The highlighting here is just to indicate the various bits – there’s no highlighting when you put the code into VBA.)

# Installing macros in Word – your first macro

You are now ready to install your first macro. ‘Installing a macro’ just means copying the code and pasting it in the correct place, nothing more technical than that.

**Where do we put the macros?** They are stored in the **Normal template** (also call the ‘global template’), which is accessed via the VBA window.

**How do we access the VBA window?** You can get in through any Microsoft Word document, and once there, you simply copy and paste the macro code.

The first macro you will install is GoogleFetch, which speeds up searching Google for a word or a set of words in your project document. But more on that later.

Follow these steps to install GoogleFetch on your computer.

1. Click the link for the website address for the country of your choice:

• Australia: [www.wordmacrotools.com/macros/G/GoogleFetchAU](http://www.wordmacrotools.com/macros/G/GoogleFetchAU)

• Canada: [www.wordmacrotools.com/macros/G/GoogleFetchCA](http://www.wordmacrotools.com/macros/G/GoogleFetchCA)

• India: [www.wordmacrotools.com/macros/G/GoogleFetchIN](http://www.wordmacrotools.com/macros/G/GoogleFetchIN)

• Ireland: [www.wordmacrotools.com/macros/G/GoogleFetchIE](http://www.wordmacrotools.com/macros/G/GoogleFetchIE)

• New Zealand: [www.wordmacrotools.com/macros/G/GoogleFetchNZ](http://www.wordmacrotools.com/macros/G/GoogleFetchNZ)

• South Africa: [www.wordmacrotools.com/macros/G/GoogleFetchSA](https://www.wordmacrotools.com/macros/G/GoogleFetchSA)

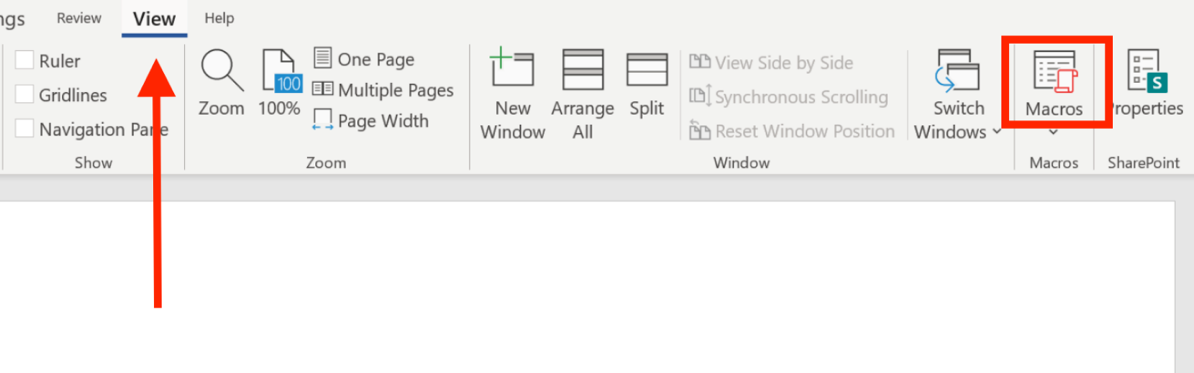
• UK: [www.wordmacrotools.com/macros/G/GoogleFetchUK](http://www.wordmacrotools.com/macros/G/GoogleFetchUK)

• US: [www.wordmacrotools.com/macros/G/GoogleFetchUS](http://www.wordmacrotools.com/macros/G/GoogleFetchUS)

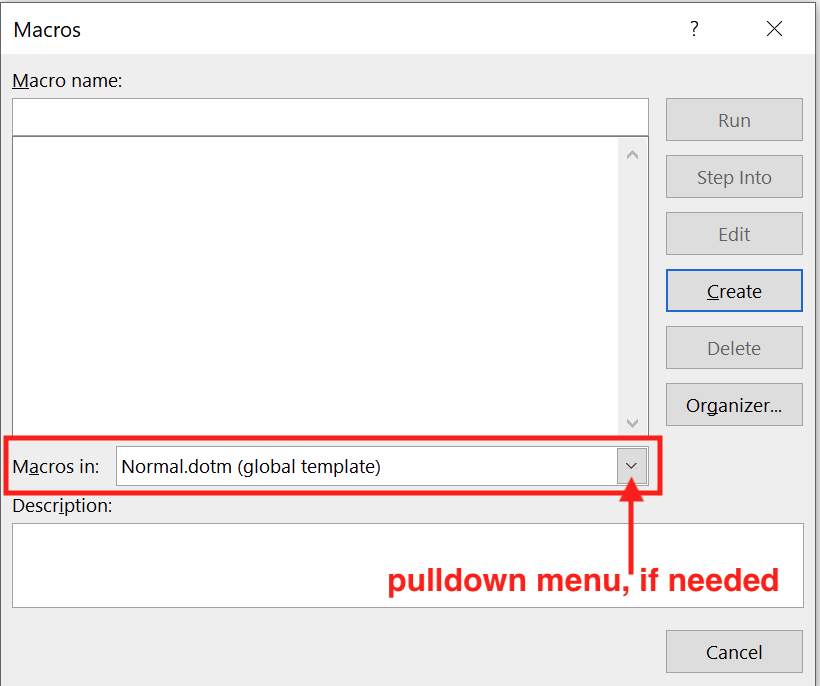
• If your country is not listed, please contact Paul Beverley at [paul@archivepub.co.uk](mailto:paul@archivepub.co.uk).

2. The website will display the computer code for the macro. Select all the text of the web page with **Ctrl**+**A** and copy it with **Ctrl**+**C**.

(If you are new to keyboard shortcuts, **Ctrl**+**A** means: hold down the **Ctrl** key and then press the **A** key. Similarly, for **Ctrl**+**C,** hold down the **Ctrl** key and press the **C** key.)

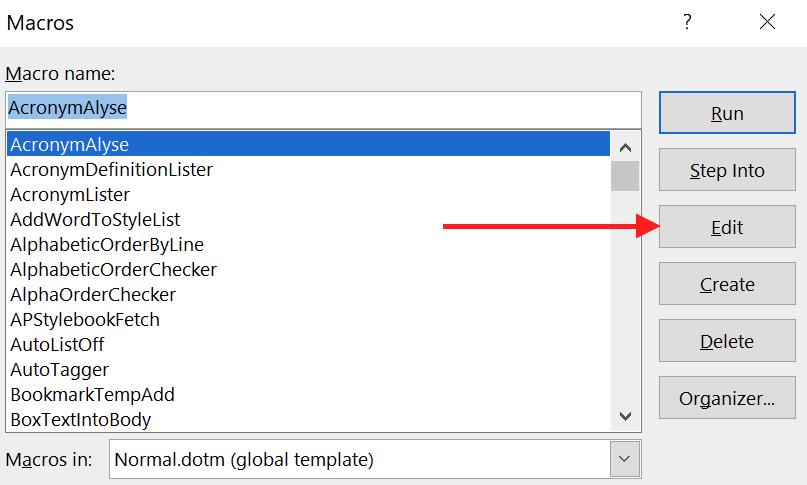
3. In a blank Word document, open the Macros dialogue box by selecting the **View** tab in the ribbon and clicking on the **Macros** icon.

4. You are now in the Macros dialogue box, which lists any macros that are already installed. Does the **Macros in:** window say **Normal.dotm (global template)**? If not, use the pulldown menu to change it. The global template is where we want to install our macros.



5. To install macros, you have to open the VBA window – this involves clicking the **Edit** button.

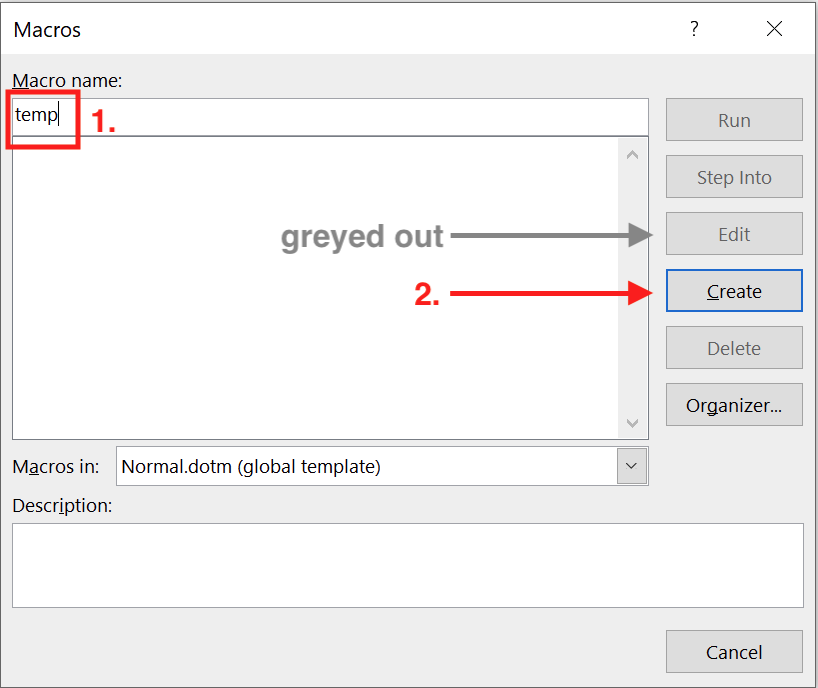
• **5a.** If you **are** able to click the **Edit** button, click it and you can then proceed to step 6.



• **5b.** **If the Edit button is greyed out and won’t click**, you need to create a temporary macro – just to get you into VBA; you will only ever have to do this one time, to install your very first macro:

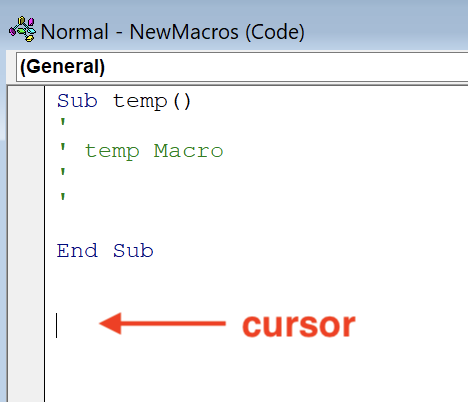
1. In the **Macro name:** box type, **temp**.

2. Click the **Create** button.



6. You are now in the VBA window ready to install the macro code.**Important:** Place the cursor at the **very** **bottom of the window, below all the other text**.

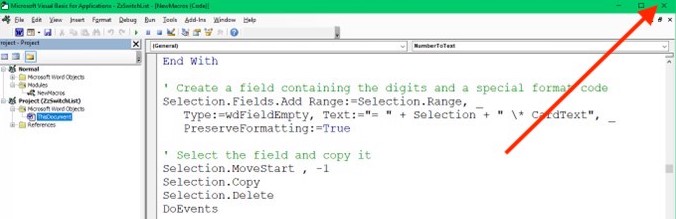
• Often when you open the VBA window, the cursor will be sitting *in the middle of* one of the macros. But if you paste the new macro inside an existing one, neither macro will work.



7. Paste the text you copied earlier by clicking **Ctrl**+**V**.

*Congratulations, you’ve just installed your first macro!*

8. Close the VBA window by clicking on the ‘x’ **box** in the **very upper right** of the **Microsoft Visual Basic** window (not the one a bit lower down). VBA automatically saves your changes.



# Practising your first macro

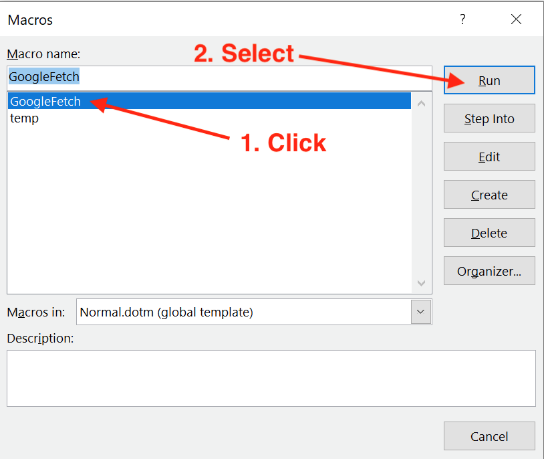
You’re now ready to see how GoogleFetch works; it automates searching the internet. Try for yourself:

1. In the sentence below, place the cursor in the word ‘Decentraland’. (GoogleFetch is smart enough to know you that want it to search for that entire word, *even though you haven’t selected it*.)

**Ex:** MANA is the cryptocurrency exclusive to Decentraland, one of several virtual worlds.

2. Select the **View** tab in the ribbon and select the **Macros** icon.

3. Click on GoogleFetch in the list of macros and select **Run**.



Notice how GoogleFetch replaces all the steps you would normally perform to look up “Decentraland” in Google: (1) copy the text you want to search, (2) open your browser, (3) load Google, (4) paste the search term, and (5) click Enter to execute the search.

Running macros by using the **Macro** dialogue box is a little slow though, don’t you think? That’s why we need to run macros using a *keyboard shortcut*.

# Keyboard shortcuts – the essentials

You may already be familiar with shortcuts and not realize it. Common ones are Ctrl+S to save your work and Ctrl+P to print. And we’ve already used Ctrl+A to select all the text, Ctrl+C to copy the text, and Ctrl+V to paste the text.

Word comes packaged with many keyboard shortcuts like these. But you can assign extra keyboard shortcuts to run your own macros, using a key combination of your choice.

On a PC, shortcuts must use:

• The **Ctrl** and/or **Alt** key, and

• A letter, number, or symbol key.

• For more options, you can add the **Shift** key.

**🡪** **Tips for how to choose a keyboard shortcut:**

• To help you remember your shortcuts, choose a key combination that has some logical connection to the assigned macro. For example, with GoogleFetch, including the **G** key will help you remember the shortcut (G = Google).

• Also consider what combinations of keys are easy for your hands to press.

• It is easy to change your keyboard shortcuts later, if you change your mind.

# Assigning keyboard shortcuts to macros

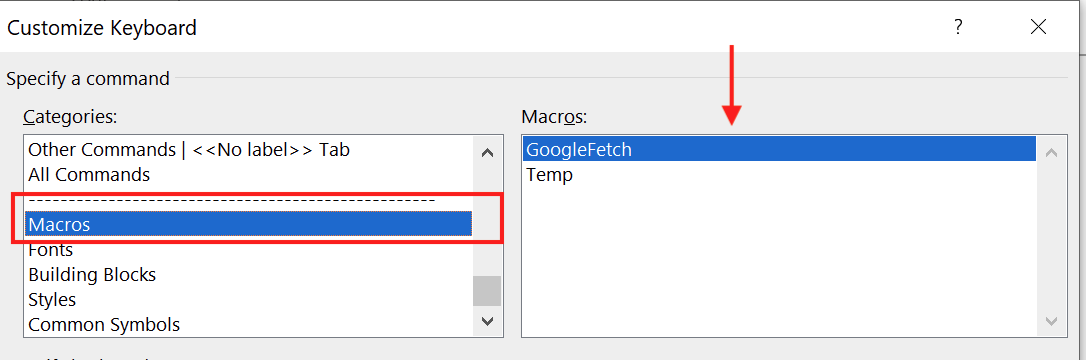
Let’s assign a shortcut to GoogleFetch.

A screenshot of a computer

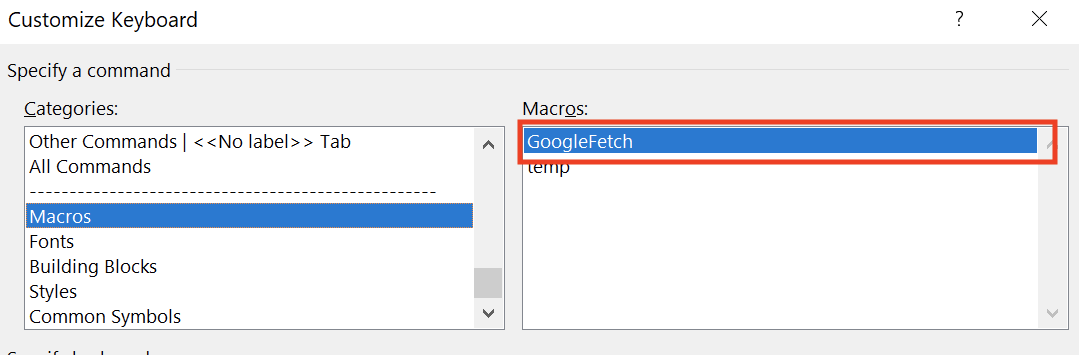
Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidence1. Open the Customize Keyboard dialogue box by (1) right-clicking the mouse anywhere in the Ribbon and selecting **Customize the Ribbon** and (2) selecting the **Keyboard shortcuts:** **Customize** button at the bottom of the window

2. Under **Categories** on the left, scroll down to **Macros** and select it. Notice that the box on the right now lists all the macros currently installed on your computer.

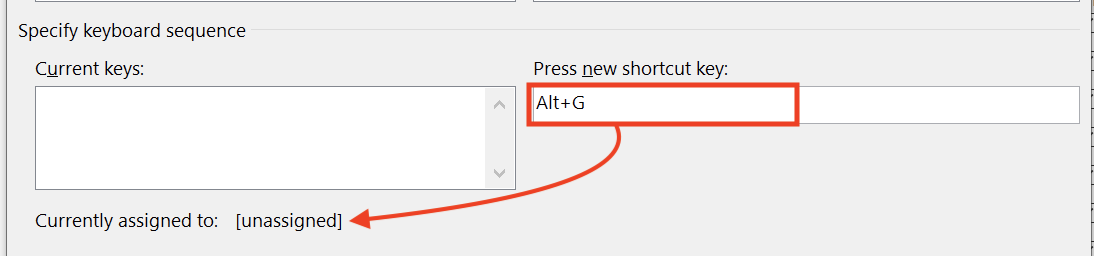


3. In the box on the right, select the macro you want to assign a keyboard shortcut to by clicking on it. For this example, select GoogleFetch (or whichever country version you are using).



4. Think of a suitable keyboard shortcut to assign to the macro – use the tips in the [**Keyboard shortcuts – the essentials**](#_Keyboard_shortcuts_–) section to help you. Then click inside the **Press new keyboard shortcut** box and type your desired shortcut key combination.

• **4a.** Does it say Currently assigned to: [unassigned]? If so, this means that the key combination is free to use. For instance, for the GoogleFetch example, press and hold down the **Alt** key and then click the **G** key.Notice how it shows that **Alt**+**G** is [unassigned].



A screenshot of a computer

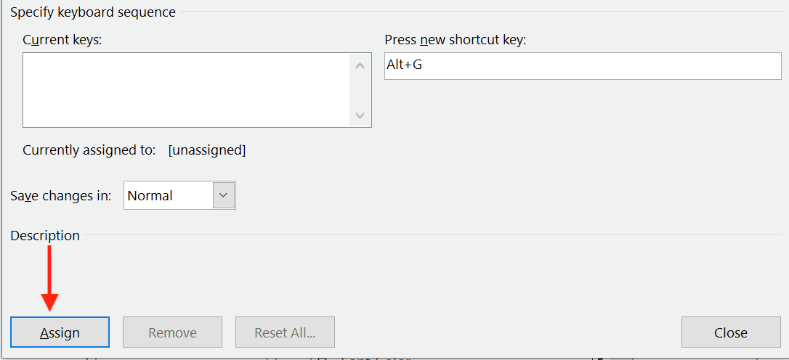
Description automatically generated• **4b.** If **Currently assigned to:** says something other than ‘[unassigned]’, it means that Word has already assigned your chosen keystroke to another function. In this example, **Ctrl**+**G** is already assigned to the **EditGoTo** function. You **can** still choose this shortcut if you want, but note that it will then no longer be available to run the other function.

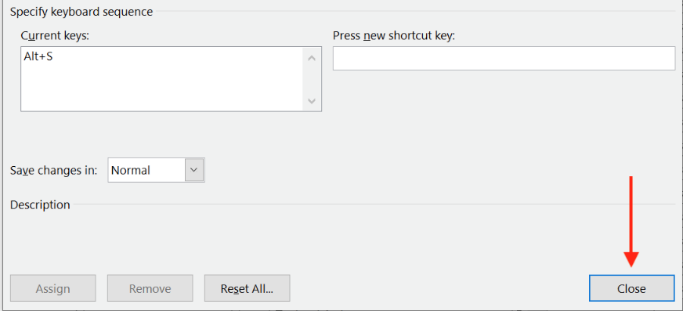
• It’s OK to select a shortcut that is already assigned to another function, as long as you don’t need to use that shortcut to perform its preassigned function. For instance, you would not want to assign Ctrl+C, the shortcut to copy text, to a macro because it is one you likely use often.

5. Once you have decided on a shortcut, remember to click **Assign**. *This is easily overlooked.*

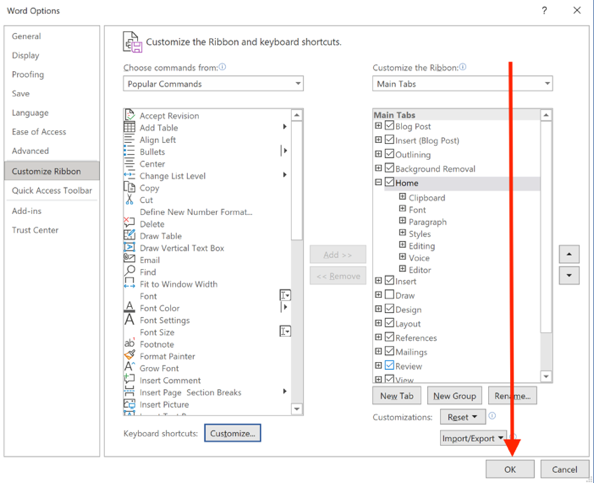
• If you forget, the shortcut will not be assigned and you will have to go through these steps again.

• For the GoogleFetch example, select **Alt+G** – or some other key combination, if you prefer. Then press **Assign**.



6. When you are finished assigning shortcuts, click the **Close** button to close the window.

• It’s also a good idea to write down your shortcuts. It is easy to forget which shortcut you’ve chosen, until you’ve used it regularly. Many editors create a table in Word listing the macro name, its keyboard shortcut, and a brief explanation of what the macro does (in case you ever forget). Ultimately, the goal is to memorize them, of course.

7. Close the Word Options menu by selecting the **OK** button. Now you’re ready to use your keyboard shortcut.

# Practising GoogleFetch using your shortcut

Place your cursor inside the highlighted word in Ex1 and Ex2 and press your keyboard shortcut. If you chose **Alt**+**G**, hold down the **Alt** key and click the **G** key.

**Ex1:** The prime minister travelled to Mackinac Island for the weekend.

**Ex2:** Finish your taco with a spoonful of Mexican crema.

**🡪** **Tip:** When you want to look up more than one word, just select the words and then run the macro. A quick and low-impact way to do that is to click the cursor inside the first word, hold down the Shift key, and click in the last word – as long as some part of each word is selected, the macro will send all of the words to Google.

**Ex3:** Completed in 1925, Briarcliff housed an Aeolian organ and was often the site of organ and other musical recitals.

**Ex4:** Henry Dreyfus designed the iconic Western Electric Model 500 telephone.

# Using macros to speed up copyedits – CaseThisWord

Let’s learn about a macro that speeds up making copyedits. CaseThisWord changes the capitalization of a word. You can run it while you’re editing, for any word whose capitalization needs changing.

First you need to install the macro and assign a keyboard shortcut:

1. Copy the code found on this web page: [www.wordmacrotools.com/macros/C/CaseThisWord](http://www.wordmacrotools.com/macros/C/CaseThisWord)

2. Follow the instructions for installing macros in the [**Installing macros in Word – your first macro**](#_Your_first_macro) section. (**Note**: This time, you ***will*** be able to click **Edit** (step 5a) and doing so will get you straight into VBA to paste in your new macro.)

3. Once you have installed the macro, assign it a shortcut by following the steps in the [**Assigning keyboard shortcuts to macros**](#_Assigning_keyboard_shortcuts) section. You may want to include the letter ‘C’ in the key combination to help you remember this shortcut (C = capitalization). **Shift+Alt**+**C** is a likely candidate. But you can choose whatever key combination you like.

Now you’re ready to practise this macro.

1. In the exercises below, place the cursor inside the word whose capitalization needs to be changed – indicated by grey highlighting.

2. Use your chosen shortcut to run the macro, e.g., if you chose, **Shift+Alt**+**C**, hold down the **Shift** and **Alt** keys and click the **C** key.

**🡪 Tip:** Turn on Track Changes if you want to record the edits the macro makes.

**Ex1:** He built his home immediately East of the original house.

**Ex2:** She travelled to south America to study some of its famous architecture.

**🡪** **Tip:** In the following examples, when you need to change the capitalization of more than one word in a row, you do not need to move the cursor. Just click the shortcut again – as many times as you need.

**Ex3:** Frazier and Bodin were among the few Atlanta firms that designed in an Avant Garde style.

**Ex4:** The designers created a handheld Personal Radiation Detector for law enforcement, fire rescue, and other emergency responders.

# Another speed-editing macro – NumberToText

NumberToText converts a number (numeral) to text. Try it out for yourself.

First, install the macro and assign it a keyboard shortcut:

1. Copy the code for the macro with the number convention you follow:

**UK:** [www.wordmacrotools.com/macros/N/NumberToTextUK](http://www.wordmacrotools.com/macros/N/NumberToTextUK)

**US:** [www.wordmacrotools.com/macros/N/NumberToText](http://www.wordmacrotools.com/macros/N/NumberToText)

(They are only different for larger numbers, e.g., UK ‘three hundred *and* six’, US ‘three hundred six’.)

2. Follow the instructions for installing macros in the [**Installing macros in Word – your first macro**](#_Your_first_macro) section.

3. Once you have installed the macro, assign it a shortcut by following the steps in the [**Assigning keyboard shortcuts to macros**](#_Assigning_keyboard_shortcuts) section. You may want to include the number 1 in the key combination to help you remember this shortcut (1 = a number). **Ctrl**+**Shift**+**1** is a likely candidate. But you can choose whatever key combination you like.

Now you’re ready to practise this macro:

1. In the exercises below, place the cursor in front of the numeral to be changed.

2. Use your chosen shortcut to run the macro, e.g., if you chose, **Ctrl**+**Shift**+**1**, hold down the **Control** and **Shift** keys and click the **1** key. (Note: It will show as Ctrl-!, because Shift-1 is a “!”.)

**Ex1:** Frazier’s professional relationship with Charles Black began 3 years after forming his architecture practice.

**Ex2:** The evacuation time of the platform was found to be below 4 minutes and walking time to a point of safety to be below 6 minutes.

**🡪**  **Tip:** The cursor does not have to be immediately adjacent to the numeral you want to change. The macro scans the upcoming text, jumping to the next numeral.

**Ex3:** They spent the next 15 years contributing to the architectural character of this exclusive Buckhead neighborhood. In all, they built 36 houses.

**Ex4:** The Peachtree elevation contained a similar arrangement as the Pallas Apartments, although much reduced in scale; 2 protruding bays created a recessed central bay and 2 end bays.

# Using macros to check for consistency – ProperNounAlyse

The macros you’ve tried so far are meant to be used as you edit. ProperNounAlyse is meant to be used before you make any copyedits. It analyses an entire document for potential inconsistencies between proper nouns.

In a separate file, it generates a (1) ‘Proper noun queries’ report which lists words that could possibly be variant spellings of one another and (2) a frequency list of all the proper nouns (which the macro defines as any capitalized word). The ‘Proper noun queries’ report uses different coloured highlighting to help you link the pairs, because they are sometimes separated by being in strict alphabetical order (e.g., ‘Bath’ and ‘Both’**\***). You then review the report to see which might be actual mistakes – see the graphical explanation below. You could then add them to your style sheet and correct them in the text before you begin editing.

(\*You know immediately that these are not variant spellings of a proper noun, but the computer sees them as spelling variation similar to ‘Anderson’ and ‘Andersen’; this is an example of computer–human teamwork.)

A screenshot of a computer

Description automatically generated

To try out ProperNounAlyse, you need to supply your own document:

1. Copy the code: [www.wordmacrotools.com/macros/P/ProperNounAlyse](http://www.wordmacrotools.com/macros/P/ProperNounAlyse)

2. Follow the instructions for installing macros in the [**Installing macros in Word – your first macro**](#_Your_first_macro) section.

3. Once you have installed the macro, you *can* assign it a shortcut. However, since you run it only once during a project, using a shortcut might not be worthwhile. Instead, you can run it by opening the Macro dialogue box (select the **Macro** icon on the **View**tab), selecting ProperNounAlyse from the list, and then clicking the **Run** button.

4. Choose a new or recent project. Use the unedited text. Open the document and run ProperNounAlyse.

**🡪 IMPORTANT:** When you run ProperNounAlyse, and other similar analysis macros, you need to give the computer time to work. **Do not touch your keyboard or mouse** while the macro is working. Word may crash.

5. Analysethe results in the ProperNounQueries report and write down the possible inconsistencies. Here are some tips:

• Use the highlighting to identify the word pairs the macro wants you to assess.

• Use your knowledge of proper noun spellings to pick out which word pairs may contain an erroneous spelling.

• Note that not all word pairs are errors. The computer is doing a set of simple comparisons; it is unable to understand variant spellings, such as Europe and European.

6. The next step is to figure out which of the potential inconsistencies are true mistakes. Some inconsistencies may be obvious errors at a glance, but others may need further investigation. You may need to:

• Search the document using Word’s Find box.

• Do some research, such as search Google (GoogleFetch?).

• Consult a dictionary or style guide (using another fetch macro?).

• Query the author.

7. Once you have a list of errors, correct them using Word’s Find and Replace. Don’t forget to add them to your style sheet too.

# Next steps, part 1 – ‘practice makes perfect’

Which of the above macros most piqued your interest? Which seem most useful for the type of editing you do? Choose one or two and start using them in your daily editing.

Here’s a reminder of what the above macros do and when you should use them:

• Run **GoogleFetch** as you are editing when you come across a word(s) you need to look up.

• Run **CaseNextWord** as you are editing when you come across a word whose capitalization needs changing

• Run **NumberToText** as you are editing when you come across a numeral that needs to be spelled out.

• Run **ProperNounAlyse** before you begin editing to analyse the document for inconsistencies with proper nouns, and then you can correct those issues before you begin editing line by line.

When you are comfortable using all the ones that are useful in *your* work, then move to [part 2](#_Next_steps,_part_1) to learn about other macros that are popular among editors.

# An important interlude – always backup your macros!

Very very occasionally, your macros can disappear. This can happen when Word crashes or updates. Save yourself the headache of having to reinstall your macros one by one by creating a backup. Get in the habit of making a backup every time you install new macros.

Follow these steps to make a backup of the macros you’ve installed so far:

1. Open VBA (View–Macros–Edit). Or you can press **Alt+F11**.

2. Use **Ctrl**+**A** to select the code for all the macros,and copy it with **Ctrl**+**C**.

3. Close VBA and open a new Word document.

4. Paste the text into the Word document by using **Ctrl**+**V**.

5. Save the document in an easy-to-find location. *Tip:* Add the date to the filename so you know which is the most recent backup.

6. Create a new backup any time you install a new macro.

**How to Reinstate Your Macros If You Lose Them**

\* This method is for reinstalling all your macros at once. If only one macro disappeared, ignore these steps and reinstall just that one.

1. Open your macros backup document.

2. Copy all the text by selecting it with **Ctrl**+**A** and copying it with **Ctrl+C**.

3. Follow the steps in the [**Installing macros in Word – your first macro**](#_Your_first_macro) to install the macro code you’ve just copied.

*Note:* This method will not back up your keyboard shortcuts, so you will have to reinstall those. Once your macro collection grows, or if you also want to backup your keyboard shortcuts, you can backup your Normal template. This is an intermediate technique. Watch [this video](https://youtu.be/Y6biwUW-rME) for instructions (PC only) and consult these [written instructions](https://www.wordmacrotools.com/pdfs/Appendix_12__Backing_up_the_Normal_Template.pdf) (scroll down for PC).

# Next steps, part 2 – other macros to explore

Once you’ve practised with the above macros enough to be confident using them, it’s time to try out some others. Choose one or two from this list that look useful and start using them in your work. When those become second nature, add one or two more. When you’ve explored all the macros in this section that are useful in *your* work, move to [part 3](#_Next_steps,_part) to learn about how you can find more macros.

Note: The macro names below contain embedded links to the code to install in Word.

Note2: If you don’t understand how some of these macros function, see the[**Next steps, part 3 – how to find more macros**](#_Next_steps,_part) section to learn how to locate the instructions for each macro.

**If you like GoogleFetch, try one of these:**

• [MerriamFetch](https://www.wordmacrotools.com/macros/M/MerriamFetch.txt): Looks up the selected text on Merriam-Webster

• [MerriamUnabridgedFetch](https://www.wordmacrotools.com/macros/M/MerriamUnabridgedFetch.txt): Looks up the selected text on Merriam-Webster (for subscribers)

• [Macquarie](https://www.wordmacrotools.com/macros/M/MacquarieFetch.txt)Fetch: Looks up the selected text on Macquarie

• [ThesaurusFetch](https://www.wordmacrotools.com/macros/T/ThesaurusFetch.txt): Looks up the selected text on thesaurus.com

• [GoogleScholarFetch](https://www.wordmacrotools.com/macros/G/GoogleScholarFetch.txt): Looks up the selected text on Google Scholar

• [PubMedFetch](https://www.wordmacrotools.com/macros/P/PubMedFetch.txt): Looks up the selected text on PubMed

**If you like CaseThisWord or NumberToText, try one of these:**

• [CapperMax](https://www.wordmacrotools.com/macros/C/CapperMax.txt): Applies title case to titles and headers, leaving certain words lowercase ("and," "of," etc.). [CapperMin](https://www.wordmacrotools.com/macros/C/CapperMin.txt) applies sentence case to titles and headers.

• [Comma](https://www.wordmacrotools.com/macros/C/Comma.txt): Changes a punctuation mark to a comma, or adds a comma, and lowercases the next word, if needed

• [DeleteOneWord](https://www.wordmacrotools.com/macros/D/DeleteOneWord.txt): Deletes the current word, but not the punctuation

• [FullPoint](http://wordmacrotools.com/macros/F/FullPoint): Changes a punctuation mark to a full stop (period), or adds a full stop, and capitalizes the next word, if needed

• [ParenthesesAdd](http://wordmacrotools.com/macros/P/ParenthesesAdd): Add parentheses around the selected word or phrase

• [PunctuationOff](http://wordmacrotools.com/macros/P/PunctuationOff): Removes the next punctuation mark

• [PunctuationToHyphen](https://www.wordmacrotools.com/macros/P/PunctuationToHyphen.txt): Changes a dash or space between two words to a hyphen

• [SwapWords](http://wordmacrotools.com/macros/S/SwapWords): Swaps adjacent words (e.g.: changes “It is *this* like” to “It is like *this*”)

• [TextToNumber](https://www.wordmacrotools.com/macros/T/TextToNumber.txt): Converts spelled-out numbers to numerals

**If you like ProperNounAlyse, try one of these:**

• [HyphenAlyse](http://www.wordmacrotools.com/macros/H/HyphenAlyse): Identifies potential inconsistencies in hyphenation

• [WordPairAlyse](https://www.wordmacrotools.com/macros/W/WordPairAlyse.txt): Identifies potential inconsistencies in whether a term is spelled as one word or two separate words

• [CitationAlyse](http://www.wordmacrotools.com/macros/C/CitationAlyse): Creates a list of the citations paired with their references, which you scan for errors; watch [this YouTube video](https://www.youtube.com/watch?v=l5mkFuRps_o) for details

**Other macros popular among editors:**

• [CountRemainder](https://www.wordmacrotools.com/macros/C/CountRemainder.txt): Tells you how many words you’ve edited and how many still to go

• [KeystrokeLister](https://www.wordmacrotools.com/macros/K/KeystrokeLister.txt): Generates a list of all the shortcuts you’ve assigned

• [LanguageToogle](https://www.wordmacrotools.com/macros/L/LanguageToggle.txt): Changes the language setting of the selected text

• [PasteUnformatted](https://www.wordmacrotools.com/macros/P/PasteUnformatted.txt): Pastes text formatted to match the surrounding text

• [TrackOnOffVisible](https://www.wordmacrotools.com/macros/T/TrackOnOffVisible.txt'): Switches tracking off with a visible background so you don’t forget to turn it back on, which you do by running the macro a second time

• [WhatChar](https://www.wordmacrotools.com/macros/W/WhatChar.txt): Provides ASCII and Unicode numbers and names for the selected character

**Two intermediate find-and-replace macros:**

• [FRedit](https://www.wordmacrotools.com/macros/F/FRedit.txt): A powerful global find-and-replace macro to use before you begin editing, to clear the document of routine global errors and inconsistencies (such as misspelled names, double spaces, the formatting of dashes, etc.). Use the [FRedit from Square One](http://www.archivepub.co.uk/documents/FSO) self-tutoring guide to learn how to use it.

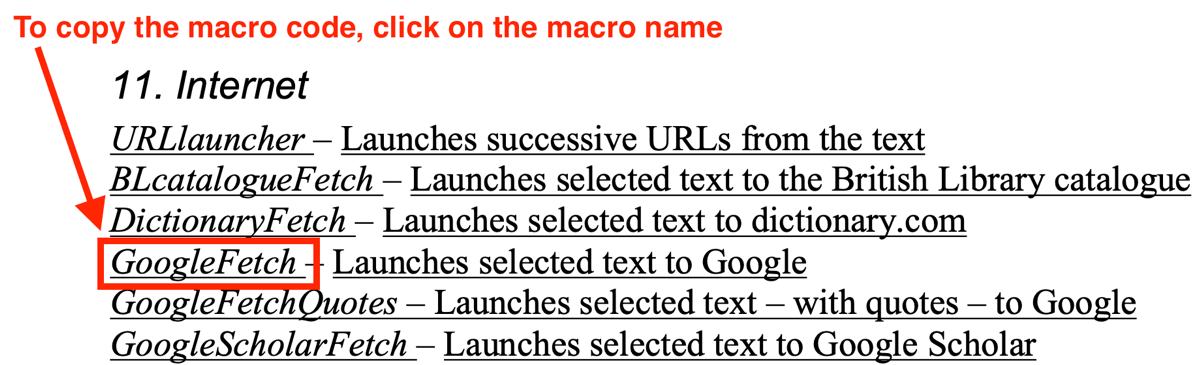
• [MultiSwitch](https://www.wordmacrotools.com/macros/M/MultiSwitch.txt): A situation-dependent replace macro that you use while you are editing. Use the [MultiSwitch from Square One](http://www.archivepub.co.uk/documents/MSSO) self-tutoring guide to learn how to use it.

# Next steps, part 3 – how to find more macros

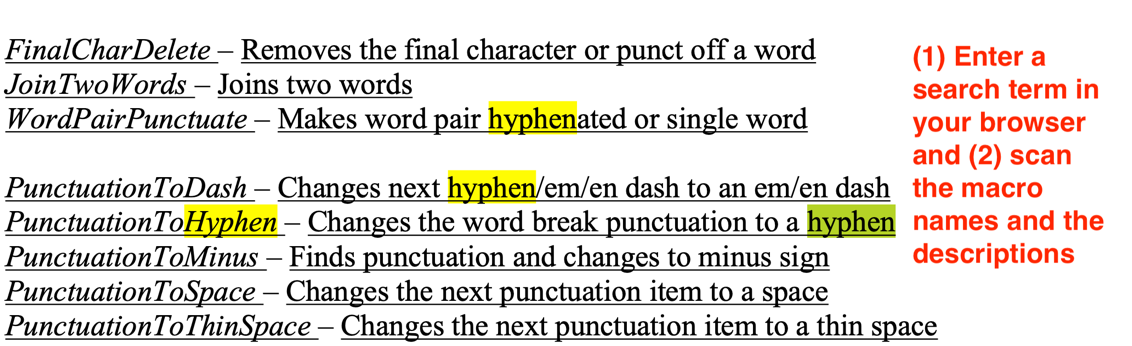
After you’ve become proficient with those macros above that are useful to your work, it’s time to learn how to find more.

When you are finding yourself doing the same repetitive edit (e.g., inserting a hyphen) that’s a sign you should stop and look for a macro to speed up the task.

All the macros are findable from the [Macro Menu](https://www.wordmacrotools.com/pdfs/08_Macro_Menu__complete_macro_tool_list.pdf), where they are organized into categories. The macro names are listed on the left, with a brief description on the right. If you click on the macro name, it opens a new page containing the code for the macro, which you can easily copy.



The easiest way to find a macro is through a search. If you want a macro to help you with hyphens, for example, use your browser’s search function (usually accessed with Ctrl+F) to search for the word ‘hyphen’ on the webpage. Then scroll through the results, looking at the macro names and reading the descriptions until you find one that looks helpful. Then install it and try it out.



If you would like more information about how a given macro works, PunctuationToHyphen, for instance, click on the brief description of the macro on the right. This will open a page that contains an explanation of how the macro works. However, this page showcases not only the macro you are interested in but also similar macros. Therefore, you need to search for the macro you want to learn about. Use your browser’s search function (usually accessed with Ctrl+F) to search for PunctuationToHyphen.

## Good luck!

This should have given you all the skills you need to start using macros. Keep practising and you will see how natural they can be without even thinking about it – just like the other tools you use.

Oh, and there’s an error that people sometimes get which stumps them – see below.

**Questions? Feedback?** Pleaseemail Paul Beverley: [paul@wordmacrotools.com](mailto:paul@wordmacrotools.com)

Maybe buy us a coffee as a thank-you? [ko-fi.com/MrMacros](http://ko-fi.com/MrMacros)

**Want to learn about one new macro each month, plus tips about using macros?** Sign up for the **Macro of the Month** newsletter: <https://macroofthemonth.substack.com>

Lead author: Jennifer Yankopolus, [jennifer@wordcollaborative.com](mailto:jennifer@wordcollaborative.com)

Ably assisted by: Karen Cox

Mistakes added by: Paul Beverley, [paul@wordmacrotools.com](mailto:paul@wordmacrotools.com)

version date: 8 December 2023

**Common Error: “Ambiguous name detected”**

If you’ve seen this error, then you’ll know it’s a **total full stop** to your macro use. It completely stops you running any macros at all.

Graphical user interface, text, application, chat or text message

Description automatically generated

**Don’t worry the solution is simple...** but what’s the cause?

Most likely, you’ve pasted in a ‘new’ macro (in this case called Hyphenate), not realizing that you’ve already got a copy of that macro. But all macros must have a unique name, so Visual Basic is warning you that you have two macros with the same name, i.e., an ‘ambiguous filename’.

The solution is obviously to delete one of them, but the macro list in the View–Macros dialogue box, only lists one ‘Hyphenate’, and it won’t let you delete it anyway!

Here then are the steps to delete the duplicate macro:

1. Open VBA: **Alt+F11** (or the usual View–Macros–Edit).

2. Click on the pulldown menu in the top right of the Code window and scroll to the macro in question. You will see the macro listed twice. In this example, Hyphenate.

Graphical user interface, text, application

Description automatically generated

3. Click on one instance of the macro. That will place your cursor inside the macro.

Graphical user interface, text, application, Word

Description automatically generated

4. Carefully select the entire macro with your mouse, i.e. be sure to select all the text, from **Sub Hyphenate()** to **End Sub**.

Graphical user interface, text, application, Word

Description automatically generated

5. Delete the text.

**Sorted!** (Well, it’s sorted unless you happen to have **another** duplicate macro, but just repeat the same process – VBA will tell you the name of the duplicate macro.)