### (About 1036 words)

### Creating and Using Macros in MS Word 365

#### Tom Burt, Vice-President Sun City Summerlin Computer Club <https://www.scscc.club>

#### tomburt8913 (at) cox.net

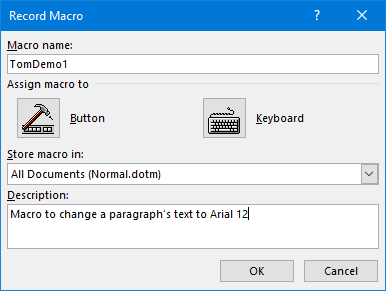
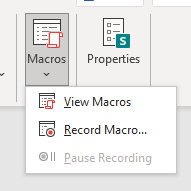
This month we'll take a look at macros, a little-used but powerful feature of all the MS Office programs. A macro is a piece of program code that can be executed whenever you want to operate on a section of a document to perform a defined sequence of changes such as formatting or editing. For example, macros can help insert letterheads, generate pre-formatted tables, applying fonts, highlighting, or styles. In addition, macros make it easy to carry out every day repetitive actions as easily as clicking a single button. For this article, we'll focus on Word macros.

## Creating and Saving Macros using the Macro Recorder

The easiest way to create a macro in Word is to turn on macro recording, perform a sequence of formatting and/or editing steps and then stop recording. The recorded macro can be given a name and have an optional hot-key sequence or a "Quick Access" button assigned to it.

Here are the essential steps:

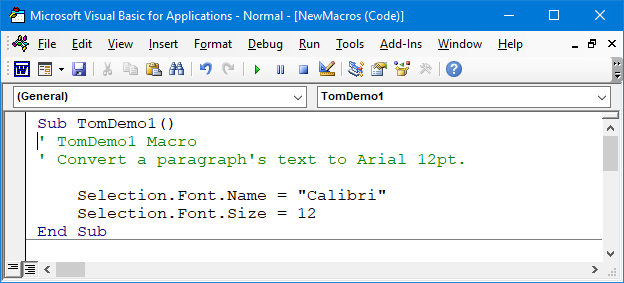
On the View tab of the Word ribbon, go to Macros and click the down (v) widget to open a menu; choose "Record Macro." That will open a small Record Macro dialog. Give the macro a name and optionally fill in the description. You can also click to assign a "Quick Access" button or a hot-key sequence to your macro. Finally, click OK and then begin the sequence of steps to be recorded.



When you are finished, go back to View > Macros, click the down (v) widget, and choose Stop Recording. Your newly recorded macro's name will now appear in the list of macros. You can go to this list, select your macro and then click the Run button to run it. More conveniently, if you assign the macro to a Quick Access button, click the button to run the associated macro.

## Editing Macros Using the Macro Editor

Word generates a series of program statements in the Visual Basic for Applications (VBA) language when you record a macro. A Word document is organized into a hierarchical set of collections such as sections, paragraphs, sentences, words. These elements can be accessed using VBA statements, and their various properties can be changed. You can see the generated VBA code by going to View > Macros > View Macros, selecting your macro, and clicking the Edit button. This will open the VBA macro editor with your macro displayed.



**Word Macro (Visual Basic) Editor**

Let's look at my very simple macro's recorded VBA statements. The green lines preceded by apostrophes are code comments.

Sub TomDemo1()

' TomDemo1 Macro

' Convert a paragraph's text to Arial 12pt.

Selection.Font.Name = "Calibri"

Selection.Font.Size = 12

End Sub

Once a paragraph or sentence has been selected, clicking the Quick Access button to run the macro changes the selection's font name to Calibri and font size to 12 points.

Let's make this macro do a bit more work by also setting the line spacing to 1.5 lines and the text alignment to Justified. The macro would then look like this:

Sub TomDemo1()

' TomDemo1 Macro

' Convert a paragraph's text to Arial 12pt, line spacing 1.5,

' alignment justified.

With Selection

.Font.Name = "Calibri"

.Font.Size = 12

.ParagraphFormat.LineSpacingRule = wdLineSpace1pt5

.ParagraphFormat.Alignment = wdAlignParagraphJustify

End With

End Sub

Note that I've also modified the macro to use a VBA "With" clause to eliminate the need to explicitly specify "Selection." for each property reference. This makes the macro code easier to read and run faster and also simplifies making changes. The "wd" symbols to the right of the = signs are Word internal symbolic constants.

The macro editor is helpful by automatically offering a menu of "Intellisense" choices after you type in a. "Thus, you don't have to remember or learn every possible property of every object or collection.

## Example Macro – Formatting an Article for the Gigabyte Gazette

Now let's look at a more real-world situation. Every month, when I edit the Computer Club's Gigabyte Gazette newsletter, I need to integrate articles and other content from various sources and give them a standard format. One way I simplify this is by using predefined Word styles. I have five: Heading 1, Heading 2, Heading 3, Heading 4, and GGBody.

Here's the VBA code of a macro named GGFormat that applies the GGBody style to each paragraph in a selected area of a document, except those that already have a "Heading x" style.

Sub GGFormat()

' GGFormat Macro - Convert all non-Heading paragraph Styles to "GGBody"

Dim pg As Word.Paragraph

With Selection

' Iterate the document's collection of Paragraph objects

For Each pg In.Paragraphs

' Only change non-Heading paragraphs

If Left(pg.Style, 7) <> "Heading" Then

pg.Style = "GGBody"

End If

Next

End With

End Sub

The "For Each" statement loops through the Selection's collection of paragraph objects, successively referencing each via the pg variable. Each paragraph's Style property is checked to see if it begins with "Heading." If not, the paragraph's Style property is changed to "GGBody."

Now I can open one of my submitter's documents and copy its contents to the desired place in my working copy of this month's Gigabyte Gazette newsletter. Then I select all the body text of the copied article, go to View > Macros > View Macros, and run my GGFormat macro. This formats all the body text to have Style = GGBody but leaves the rest of the document alone.

## A Caution

Macros are very powerful and can sometimes affect places in your document that you did not intend. For example, while experimenting, I had a test document open but also had this document (the May Gigabyte Gazette) open. My GGFormat macro above initially operated on the entire active document rather than the current selection. I expected it just to format my test document, but it formatted both documents – NOT what I wanted when I ran it. So, I modified the GGFormat macro to only operate on the current Selection.

## Learn More

We've barely broken the surface of what can be done with macros in MS Word. Here are some hyperlinks to more in-depth information on Word macros:

<https://support.microsoft.com/en-us/office/create-or-run-a-macro-c6b99036-905c-49a6-818a-dfb98b7c3c9c>

<https://www.pcworld.com/article/2952126/word-macros-three-examples-to-automate-your-documents.html>

<https://www.ionos.com/digitalguide/online-marketing/online-sales/creating-macros-in-word/>

<https://word.tips.net/C0012_VBA_Examples.html>

A picture containing person, person, smiling, older

Description automatically generated