Microsoft Excel - Macros Explained

Macros Explained

Macros or macroinstructions allow you to automate procedures or calculations in Excel.

Macros are usually recorded using the Macro recorder and then played back.

The trigger for playing them back is either a choice, Quick Access Toolbar, a specified short cut key combination or a graphic; the macro is assigned to the graphic object so that when you click the graphic the macro plays back. Graphics are usually called Buttons or Command Buttons.

Macros are written and recorded in the Visual Basic for Applications (VBA) language.

VBA is a high-level, visual-programming version of Basic which has been developed to manipulate an office application, in this case, Excel. A "high-level" language does not mean that it is particularly difficult; quite the opposite, it means that it is quite close to a human language. However, you do not need to know how to program or learn the VBA language to create macros, just use the recorder and play back your recordings.

Macros are stored in the workbook file in modules and are available for use anywhere so long as you have the workbook file open. To actually see the VBA instructions, the code, you need to go to the Visual Basic Editor (VBE) which is the working environment for macros. It is here that you can edit macros, copy macros from one module to another, copy macros between different workbooks and rename the macros.

Note: The workbook must be saved as an “Excel Macro-Enabled Workbook”
The Macros can be stored in the workbook (see above) or in the personal macro workbook.

The XLSart folder is normally located:

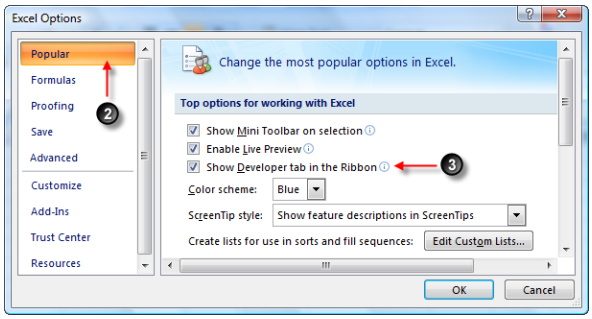
Windows Vista
“c:\Users\user name\AppData\Roaming\Microsoft\Excel

Windows XP
C:\Documents and Settings\user name\Application Data\Microsoft\Excel folder for XP.

Before it is possible to record a new macro we must enable the **Developer Options tab**, this is used to access the developer controls, write code or create Macros.

Enabling the Developer Options Tab.

1. **Click** on the **Office** button → Excel Options.



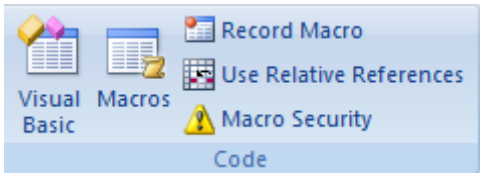
2. **Select** Popular from the left hand pane.

3. **Select** the **Show Developer tab in the Ribbon** then click **Ok**.

Recording Macros.

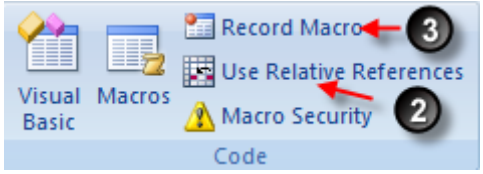
1. **Select** the **Developer** tab

   The Developer tab should display the code section.



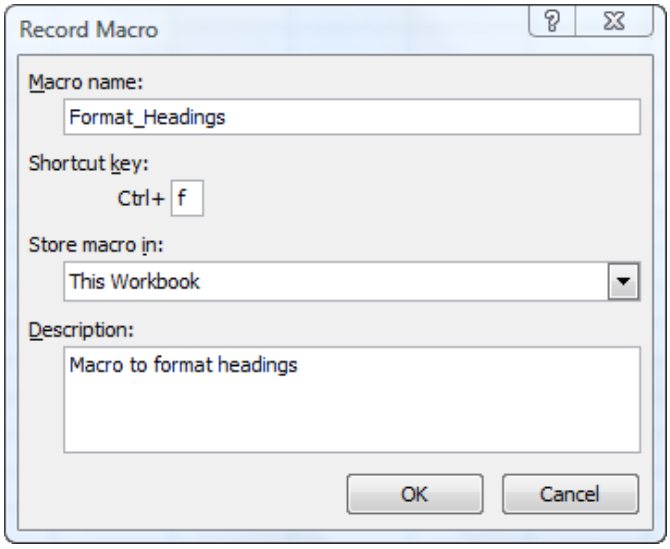
If the code section is not displayed on the Ribbon refer to the previous section:- “Enabling the Developer Options”.

2. To record a macro with actions relative to the initially selected cell, click the **Use Relative References** button.



3. Click **the Record Macro** button.

4. In the Macro name box, enter a name for the macro. The first character of the macro name must be a letter. Other characters can be letters, numbers, or underscore characters. Spaces are not allowed in a macro name; use an underscore character as a word separator. Do not use a macro name that is also a cell reference.



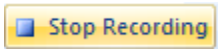
If you want to run the macro by pressing a keyboard shortcut key, enter a letter in the Shortcut key box.

Use any letter key on the keyboard, the assigned key is case sensitive. The shortcut key letter may not be a number or special character such as @ or #.

The shortcut key will override any equivalent default Excel shortcut keys while the workbook that contains the macro is open, choose carefully.
Select the location where you want to store the macro. If you want a macro to be available whenever you use Excel, select Personal Macro Workbook. If the macro is task specific and does something in the active workbook then choose This Workbook. Choose either if you are not sure, you can always copy and paste them later.

Carry out the actions you want to record.

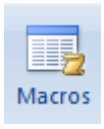
5. To Stop the Macro recording select the Stop Recording button form the **Code** section on the ribbon.



Examining the Recorded Code

You have to go to the Visual Basic Editor to view the recording or rename a macro. Visual Basic Editor; the VBE can be confusing unless you are familiar with the working environment.

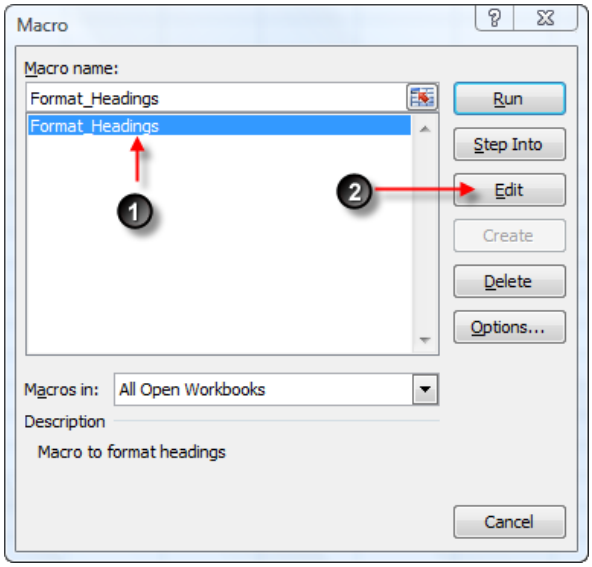
Instead, click on the **Macros** button from the **Code** section on the ribbon.



1. Select the macro name that you wish to view.

2. Click on the Edit button.

   This will take you to the VBE and put you in the right place; your recorded macro will be directly visible.




```
Sub Macro3()
'
' Macro3 Macro
' Macro recorded by me
'

'
    Range("A1").Select
    Selection.Copy
    Range("E7").Select
    ActiveSheet.Paste
    Application.CutCopyMode = False
End Sub
```

The recorded macro will look something like this. Absolute gibberish until you examine it line by line and it starts to make sense.
The code does not reflect the instructions that you gave to Excel through the menus but uses a document object based command set; the part of Excel being addressed and the action it has to perform.

Range("A1") means cell reference A1 and .Select means select it to make it the active selection. Cell A1 is now the Selection and .Copy means copy it. Object.Method. The thing that you are working with and what you want to do to it. Think Yoda. Quite straightforward, it is.
But none of this matters; you do not need to understand the code. Record the macro, play the macro back and if it works then it is fine. You only need a deeper understanding if you really want to gain control of Excel and create a more complicated procedure.

To rename the macro, replace the text after the keyword Sub. You can introduce new code, copy and edit existing code and create white space.

If you want to annotate the code, type-in comments. Comments are plain text and are not instructions. Place an apostrophe at the start of a comment line, it turns green when you move off the line.

Do not just type something in or it will probably cause the code to crash when you run the macro.

Most mistakes that you make will cause an error as you commit them; the code turns red and you are shown an error message.

But it is still possible for the code to contain an error that does not rear its ugly head until you run the macro.

```
Sub CopyValues()

    'Copy the retail summary.
    Range("A1:D50").Select
    Selection.Copy
    Range("E1").Select
    ActiveSheet.Paste

    'Copy the trade summary.
    Range("A100:D150").Select
    Selection.Copy
    Range("E52").Select
    ActiveSheet.Paste

    Application.CutCopyMode = False
End Sub
```

This is a runtime error. For example, you could enter an instruction like this; Range("A1:XX100").Select. It is structurally correct but will cause a runtime error because the range of cells does not exist and therefore can not be selected.
In the illustrations above an original recording has been renamed and the procedure has been extended by copying and editing the recorded code and the entire process annotated by introducing comments into the code. We can not emphasize enough the importance of comments for longer macros, they are only plain text but they document the process. Someone else may have to use and maintain your macros and uncommented macros are as hard to comprehend as a worksheet without labels.
The macro is part of the workbook file and needs to be saved. Either save in the VBE or go back to the Excel worksheet environment and do a normal File, Save. Return to Excel by clicking an Excel shortcut or pressing the shortcut key combination ALT+F11.