Recording Macros in Excel

A macro is a customised collection of commands which can be executed on demand to carry out a particular action in Excel. Since macros can include many different commands which are subsequently performed automatically by the computer, they are very useful for performing repetitive or complex tasks and for saving time.

Macros are written in the VBA (Visual Basic for Applications) programming language and can be stored in a regular Excel workbook or a special type of file called an Add-In. You can create a macro by writing the VBA code in Excel's Visual Basic Editor or by actually performing the process in Excel and recording your actions as you are working. This document outlines the process of recording macros.

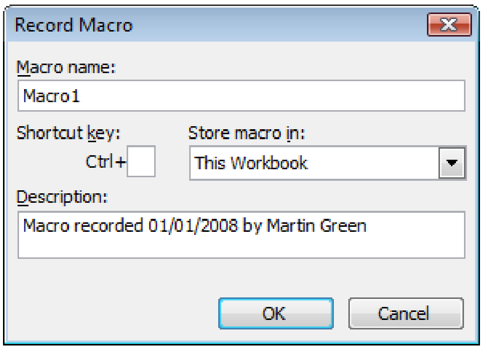
First Plan Your Macro

Before recording a macro plan exactly what you want it to do and practice carrying out the procedure manually. You might find it useful to write down the steps involved so that you don't make any mistakes when making the recording. Think about how you will use the macro and try to generalise the commands so that your macro will work in as many different situations as possible.

Prepare to Record the Macro

Ask Excel to Record a Macro

Before starting the recording think about how the macro will be used. Will the macro act on a particular cell or range of cells?. If so, should you select that cell or range before starting the recording. When you are ready to start recording, open the **Tools** menu then choose **Macro** and **Record New Macro** to open the *Record Macro* dialog box (*Fig. 1*).



Record Macro

Macro name: Macro1

Shortcut key: Ctrl+

Store macro in: This Workbook

Description: Macro recorded 01/01/2008 by Martin Green

Fig. 1 The Record Macro dialog box.

Name the Macro

Enter a suitable name for the macro. The name should not exceed 32 characters, must not include spaces (use an underscore or capitals if necessary e.g. *My First Macro* would not be permitted but *My_First_Macro* or *MyFirstMacro* are both OK). It should not contain any illegal characters and must not start with a number. Try to make the macro name describe what it does so that you can easily identify it later. You can change a macro's name later if you wish.

Define a Shortcut Key

This is optional but you might find it convenient for running the macro later. Click in the small textbox on the dialog box and then type the key you want to use for your keyboard shortcut. When using the keyboard shortcut to run the finished macro you will need to hold down the **[Control]** key whilst typing the key you specified. Since many [Control]+[Letter] shortcuts are already defined for built-in commands it is usually a good idea to enter an upper case letter (so that you will type [Control]+[Shift]+[Letter] to run the macro) so that you do not overwrite the built-in keyboard shortcut.

Decide Where to Store the Macro

The dialog offers three places to store the macro…

- Choose **This Workbook** if you want to store the macro in the current workbook. The macro will be available to use in any workbook as long as the current workbook is open. If you copy the workbook to somewhere else, or give it to someone, the macro will go with it.
- Choose **Personal Macro Workbook** to make the macro available to use in any workbook on the current computer (or, depending on network setup, on any computer when you are logged in). This is useful if only you are going to use the macro.
- Choose **New Workbook** if you don't want to use either of the other options, perhaps because you want to create an Excel Add-In to distribute the macro to other users.

Add a Description

This is an optional but useful feature. You can enter as much text as you like in the text box to describe what the macro does. This description will appear in the dialog box that is used to select a macro to run, and also alongside the macro's code in the Visual Basic Editor.

Record the Macro

Click the **OK** button on the *Record Macro* dialog box to commence recording. When you do this the *Stop Recording* toolbar appears (*Fig. 2*) and the word *Recording* appears in the Status Bar at the bottom of the Excel window. This toolbar has just two buttons: *Stop Recording* and *Relative Reference*.

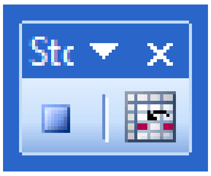


Fig. 2 The Stop Recording toolbar.

Do Only What You Want Excel to Record

The macro recorder will record all your actions in Excel, whether you use the keyboard or the mouse to give commands, until you tell it to stop, so if you make a mistake that will be recorded too. Don't do anything in Excel that you don't want to be part of the finished macro. The macro recorder records commands but not time, so it does not matter how long you take to record the macro because Excel will always replay the commands as fast as it can.

Relative or Absolute References?

If any part of your macro involves selecting cells consider whether, when the macro will be run, the same cell will always be selected at that point (e.g. you will always want the macro to select cell *B27*), or might the cell or cells be different each time (e.g. the first or last cell in a row or column, or the whole row or column, or maybe the current region of cells). It helps to familiarise yourself with the different commands in Excel that are used to select particular cells or ranges because these give non-specific commands to Excel which are useful when recording macros.

By default the macro recorder records *Absolute* references (i.e. the *Relative References* button is not pressed in). This means that if you select a specific cell or range whilst recording Excel will always select the exact same cell or range when the macro is run.

If you click the *Relative References* button the recorder will record cell selections "relative" to the current selection. If you select the cell next to the one that is currently selected, later when the macro is run at that point Excel will always select the cell next to the currently selected one wherever it happens to be.

You can switch Relative References on and off during the recording of a macro as required.

Stopping the Macro

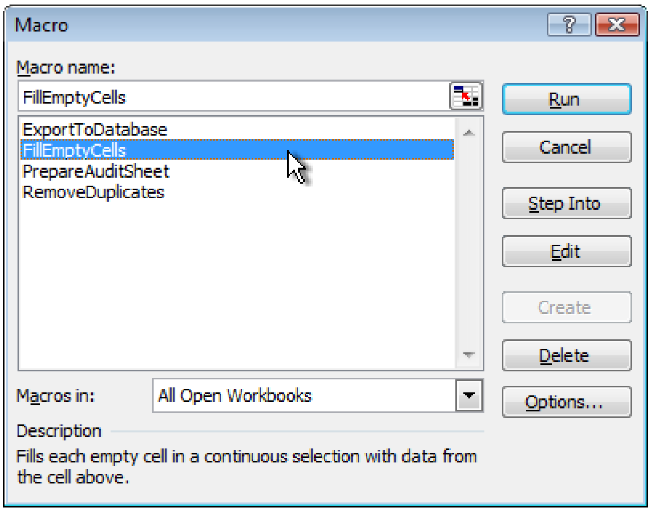
When you finish recording the macro click the **Stop Recording** button. The *Stop Recording* toolbar will disappear and the word *Recording* disappears from the Status Bar. Do not close the toolbar manually (by clicking the cross in its upper-right corner) since this does not stop the recording process. If you do this by mistake you can still stop the recording by going to the **Tools** menu then choosing **Macro** and **Stop Recording**.

What If Something Goes Wrong?

If you make a mistake when recording the macro you can sometimes simply undo your last action and continue without a problem, as long as you are aware you made an error and undo it straight away. In many circumstances the macro recorder understands and will not save the erroneous actions. Often the best course of action is to stop the macro and start again, giving your new macro the same name (Excel will ask your permission to overwrite the previous one). As you become more knowledgeable in VBA you will often be able to continue after making a mistake and correct the code manually.

Running the Macro

There are several different ways to run a macro. Open the **Tools** menu and choose **Macro** then **Macros** or use the keyboard shortcut **[Alt]+[F8]** to open the *Macros* dialog box (*Fig. 3*). This dialog lists all the available macros. Select the name of the macro you require and click the **Run** button to activate it.



Macro name: FillEmptyCells

ExportToDatabase
FillEmptyCells
PrepareAuditSheet
RemoveDuplicates

Macros in: All Open Workbooks

Description
Fills each empty cell in a continuous selection with data from the cell above.

Fig. 3 The Macros dialog box.

If you assigned a keyboard shortcut to your macro you can use this to run the macro instead of opening the dialog. You can also create toolbar buttons and menu items that run macros when selected; or assign a macro to an image placed on a worksheet or a shape drawn on a worksheet using the *Drawing* toolbar (insert the image or draw the shape then right-click on it and choose **Assign Macro**); or assign a macro to a button drawn on a worksheet using the *Forms* toolbar (create the button then right-click on it and choose **Assign Macro**).

Where to Go from Here

Recording macros is an excellent way to learn how to write VBA code. Even experienced VBA programmers sometimes record macros to save time writing code or to remind themselves (or find out) how the code should be written. If you learn some basic coding techniques such as *If Statements* and *Loops* you can introduce conditional operations into your macros (i.e. have the macro do different things in different circumstances) or carry out repetitive actions (i.e. have the macro repeat either part or all of its commands as many times as necessary).

For a more complete guide to this topic you can buy my ebook *Recording Excel Macros* from:

http://www.fontstuff.com/ebooks/ebookrem.htm

You can download a free sample file with examples from the ebook as well as view free video demonstrations at:

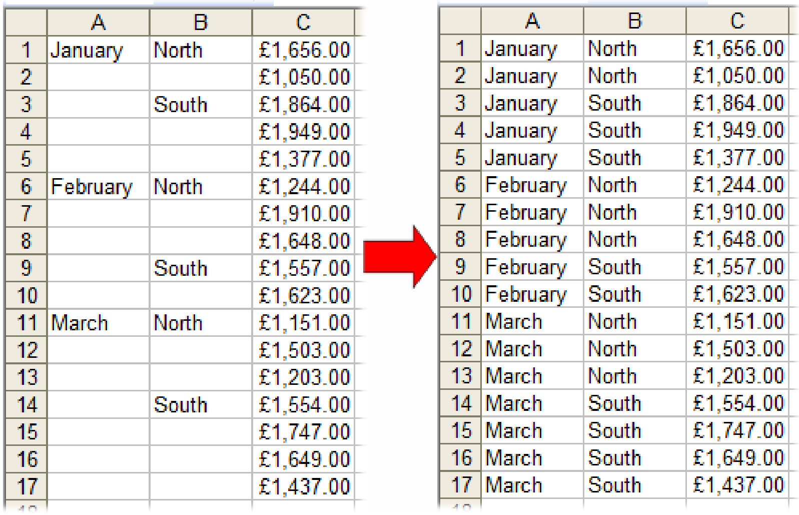
http://www.fontstuff.com/ebooks/ebookextracontent.htm#rem

Once you have learnt the basics of Excel VBA the best reference book for learning and using VBA in Excel is *Excel 2003 Power Programming with VBA* by John Walkenbach. You will find a review of this and other useful books on Excel and VBA at:

http://www.fontstuff.com/bookshelf/excelbooks.htm

Exercise: The Fill Empty Cells Macro

In this exercise you will record a macro that can be used to fill the empty cells in a range (*Fig. 4* left). When you run the macro it will fill each cell in the range with data from the cell above (*Fig. 4* right).



	A	B	C
1	January	North	£1,656.00
2			£1,050.00
3		South	£1,864.00
4			£1,949.00
5			£1,377.00
6	February	North	£1,244.00
7			£1,910.00
8			£1,648.00
9		South	£1,557.00
10			£1,623.00
11	March	North	£1,151.00
12			£1,503.00
13			£1,203.00
14		South	£1,554.00
15			£1,747.00
16			£1,649.00
17			£1,437.00

	A	B	C
1	January	North	£1,656.00
2	January	North	£1,050.00
3	January	South	£1,864.00
4	January	South	£1,949.00
5	January	South	£1,377.00
6	February	North	£1,244.00
7	February	North	£1,910.00
8	February	North	£1,648.00
9	February	South	£1,557.00
10	February	South	£1,623.00
11	March	North	£1,151.00
12	March	North	£1,503.00
13	March	North	£1,203.00
14	March	South	£1,554.00
15	March	South	£1,747.00
16	March	South	£1,649.00
17	March	South	£1,437.00

Fig. 4 The effect of running the Fill Empty Cells macro.

1. Start by preparing a spreadsheet similar to the one on the left in the above illustration (*Fig. 4*). Then select the data and make several copies of it on other worksheets so that you can use them to practice running the macro when you have finished.

2. Select any non-empty cell on the spreadsheet (i.e. one that contains data).

3. Choose **Tools > Macro > Record New Macro**. Enter the name *FillEmptyCells* and choose to save the macro in the **Personal Macro Workbook**. You can also add a shortcut key and a description if you wish. Click the **OK** button to start recording.

4. Choose **Edit > Go To > Special** then select the **Current Region** option and click **OK**.

5. Choose **Edit > Go To > Special** then select the **Blanks** option and click **OK**.

6. Do not click anywhere! On your keyboard type an **Equals** sign then press the **[Up Arrow]** key, then press **[Control]+[Enter]** (i.e. hold down the [Control] key whilst pressing the [Enter] key).

7. Choose **Edit > Go To > Special** then select the **Current Region** option and click **OK**.

8. Choose **Edit > Copy**.

9. Choose **Edit > Paste Special** then select the **Values** option and click **OK**.

10. Press the **[Esc]** (or **[Escape]**) key on your keyboard.

11. Click the **Stop Recording** button on the Stop Recording toolbar.

You have now finished recording the macro. You can test it by going to one of the copies of the spreadsheet you made, selecting a non-empty cell, and running the macro.

Create similar but differently arranged spreadsheets to test the macro to make sure that it works in different circumstances.

Is there any situation that causes the macro to "crash" or otherwise not work properly?