

Various advanced features

Setting up Excel to use Macros

To create a macro with the macro recorder, you need to use the commands on the **Developer** tab which, by default, is not displayed in the ribbon.

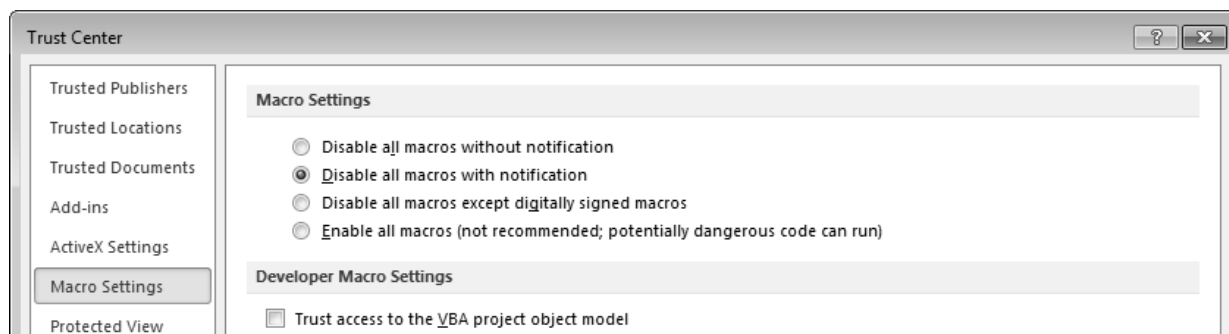
Displaying the developer tab

- ☐ Click the **File** tab, then **Options**.
- ☐ Select the **Customize Ribbon** category, then the **Developer** check box from the **Customize the Ribbon - Main Tabs** list.
- ☐ Click **OK**.

Defining macro security

To record macros, it is advisable to temporarily define the security level to enable all macros. To run macros, you need to perform this operation before even opening the workbook containing the macros you want to run.

- ☐ On the **Developer** tab, in the **Code** group, click the **Macro Security** button.




- ☐ Under **Macro Settings**, click **Enable all macros (not recommended; potentially dangerous code can run)**.
- ☐ Click **OK**.
- ☞ After you have finished working with macros, it is highly recommended that you return to the settings that disable all macros. In the **Trust Center** dialogue box, check **Disable all macros with notification**.

Recording a macro

A macro automates a command or series of frequently used commands and functions. To automate these repetitive tasks, you can record a macro. Microsoft Office Excel uses the Visual Basic application programming language to record macros.

- ☐ Start by specifying if you want to record the macro with relative references or not: on the **Developer** tab, in the **Code** group, click **Use Relative References**.

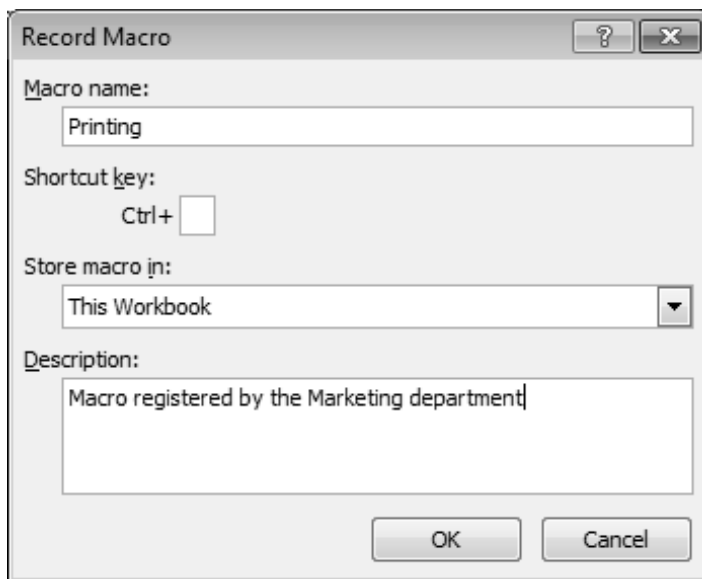
If this button is active, Excel memorises movements in the macro relative to its original position. When you run the macro, it will act on cells relative to the cell from which it is activated. Otherwise, it works with absolute references: regardless of the active cell, it will act on the cells memorised in the macro.


- ☐ On the **Developer** tab, in the **Code** group, click the **Record Macro** button, or click the  button located in the left of the status bar.
- ☐ In the **Macro name** box, enter the name you want to give to the macro.
Spaces are not allowed in a macro name, and the first character of the macro name must be a letter. Following characters can be letters, numbers, or underscore characters. Avoid using names that are also cell references.
- ☐ To assign a shortcut (to run the macro), in the **Shortcut key** box, type any lowercase letter or uppercase letter that you want to use.

- ❑ Open the **Store macro** in list and select the workbook in which you want to store the macro.

New Workbook	to store the macro in a new workbook.
This Workbook	to store the macro in the active workbook.
Personal Macro Workbook	if you want a macro to be available whenever you use Excel. Excel creates a hidden personal macro workbook, called Personal.xlsm and saves the macro in this workbook. It is loaded automatically whenever Excel starts. In Microsoft Windows 2000 and Windows XP, this workbook is saved in C:\Documents and Settings\Username\Application Data\Microsoft\Excel\XLStart. In Microsoft Windows Vista, it is saved in C:\Users\Username\Application Data\Microsoft\Excel\XLStart. In Windows 7, it is saved in C:\Users\Username\AppData\Roaming\Microsoft\Excel\XLStart.


- ❑ To include a description of the macro, enter the text in the **Description** box.



- ❑ Click **OK** to start recording.
- ❑ Perform the actions you want to record.
*Note that, in the 2010 version of Excel, you can now use the **Macro Recorder** to record changes made to the formatting of charts and other objects.*
- ❑ When all the actions have been recorded, click the **Stop Recording** button (**Developer** tab - **Code** group), or click the  button located on the left of the status bar.

Running a macro

- ❑ If the macro has been created in a workbook which is not Personal.xlsm, open it.
- ❑ On the **Developer** tab, in the **Code** group, click the **Macros** button or press **Alt F8**.
- ❑ Use the **Macros in** list to choose the location of the macro to run.
- ❑ Double-click the macro to run it.

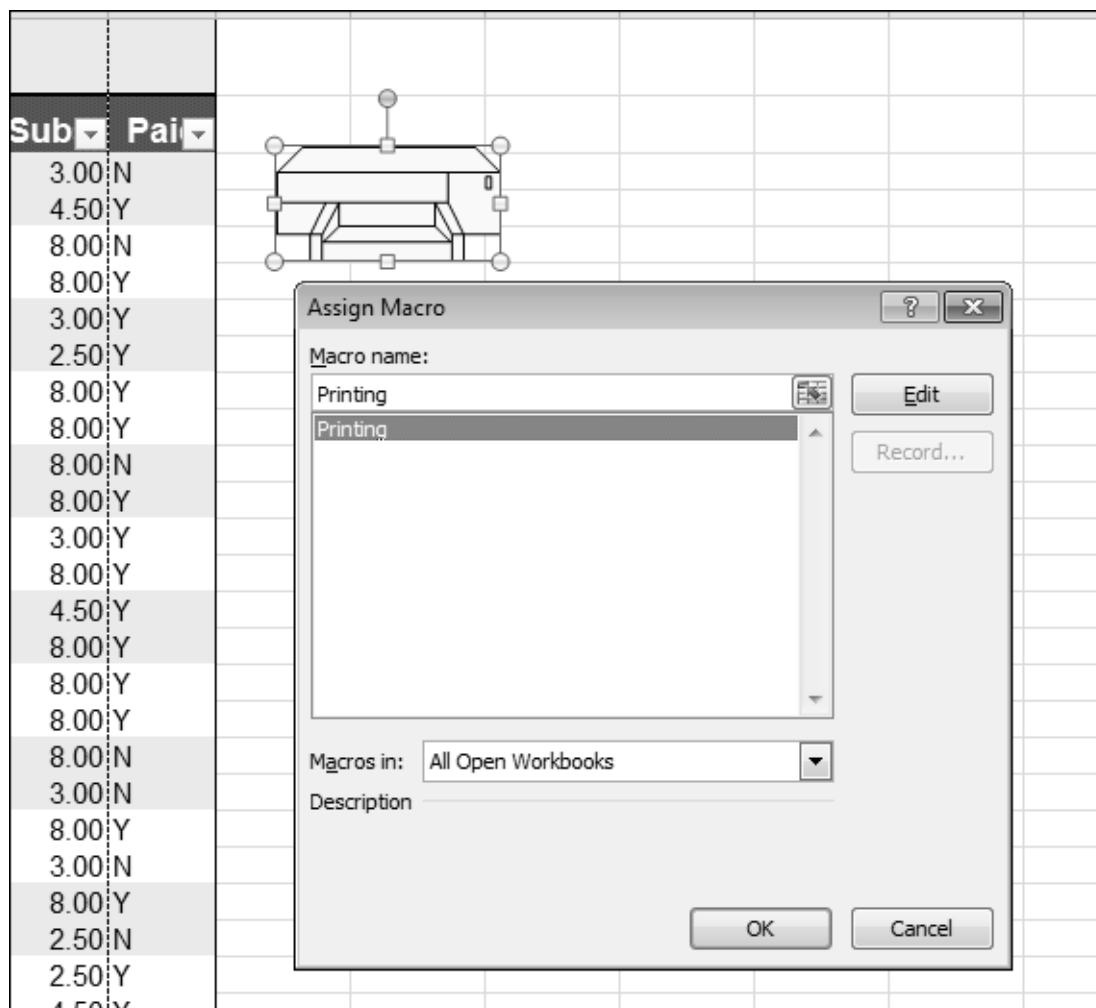
 If a shortcut key has been assigned to the macro when it was created, you can simply use the shortcut.

Running a macro on a graphic object

This technique runs a macro by clicking a graphic object.

- ❑ Right-click the graphic object on which you want to assign an existing macro.
- ❑ Click **Assign Macro**.
- ❑ In the dialogue box that opens, select the macro in the **Macro name** box.

Various advanced features




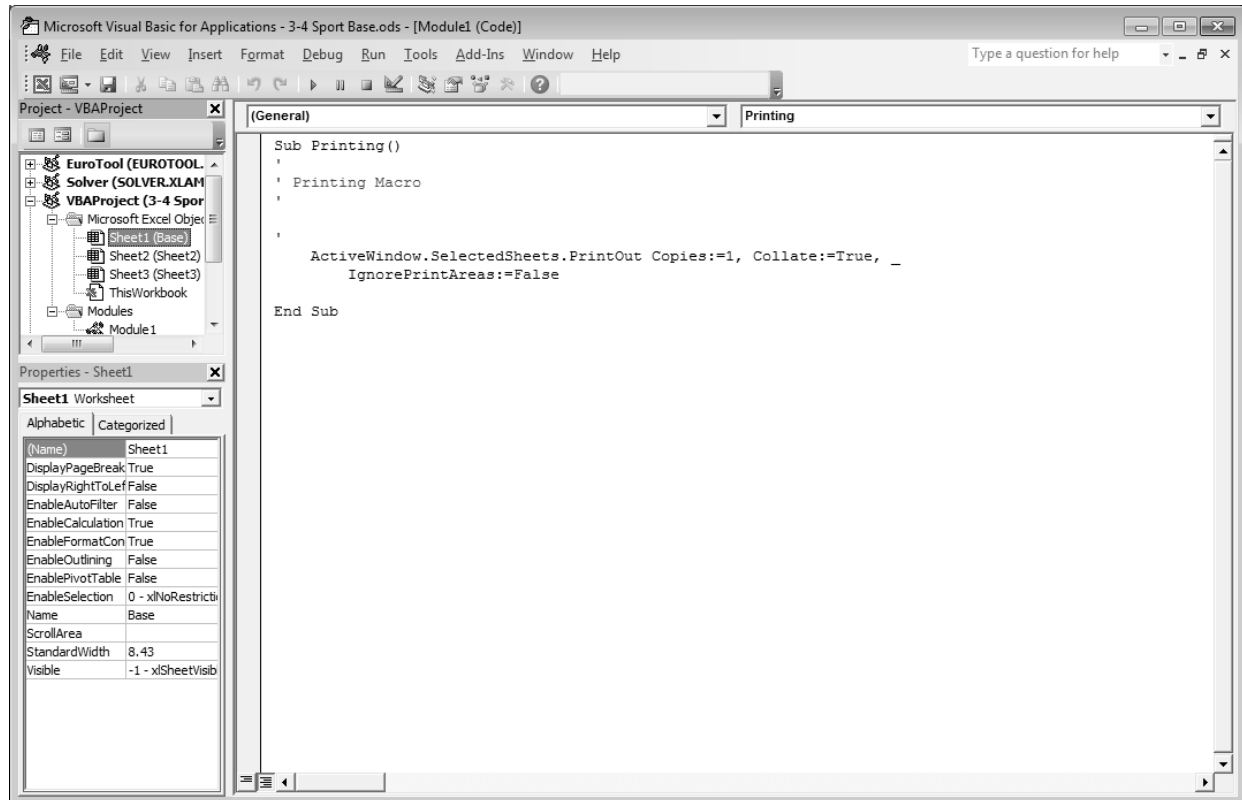
*In this example, a click on the graphic object enables the macro called **Printing** to run.*

☐ Click **OK**.

☞ To select the graphic object without activating the macro, **Ctrl** click on the object.


Modifying a macro

- ☐ If the macro has not been created in Personal.xlsb, open the workbook; if the macro is stored in Personal.xlsb, display it (**View** tab - **Window** group - **Unhide** button ).
- ☐ On the **Developer** tab, in the **Code** group, click the **Macros** button or use **Alt** **F8**.
- ☐ Open the **Macros in** list to select the workbook containing the macro you want to edit.
- ☐ Select the macro then click the **Edit** button.




The macro code appears in a **Microsoft Visual Basic for Applications** window. The code to open a macro is always **Sub** followed by the macro name, while the code to close it is **End sub**.

You do not have to know Visual Basic language to make simple changes in a macro.

- ❑ To change cell references, do the same as for any entry: go to the place required and enter the references you want, using the appropriate keys (**Del** , **←** etc.).
- ❑ To select the instructions, use the drag technique.
- ❑ To copy/move instructions, use **Edit - Copy** or **Edit - Copy** and **Edit - Paste**.
- ❑ When the changes are complete, close the **Microsoft Visual Basic for Applications** window by clicking the close button .

Deleting a macro

- ❑ If the macro has not been created in **Personal.xlsb**, open that workbook; if it has been created in **Personal.xlsb**, display it (**View** tab - **Window** group - **Unhide** button ).
- ❑ On the **Developer** tab, in the **Code** group, click the **Macros** button or press **Alt F8**.
- ❑ If required, open the **Macros** in list to select the workbook containing the macro you want to delete.
- ❑ Select the macro, then click **Delete**.
- ❑ Click **Yes** to confirm the deletion of the macro.

Saving a workbook containing macros

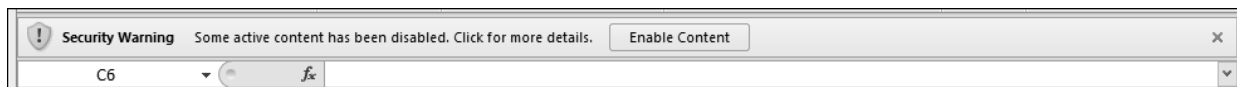
If you want to be able to reuse your macros later in the workbook, you must save it in a special format that takes the macros into account, using the **.xlsm** extension.

- ❑ Activate the relevant workbook.
- ❑ Click the **File** tab, and click **Save As**.
- ❑ In the **Save As** dialogue box which now opens, change the destination folder and file name as appropriate.
- ❑ Open the **Save as type** list and click **Excel Macro-Enabled Workbook**.
- ❑ Click the **Save** button.

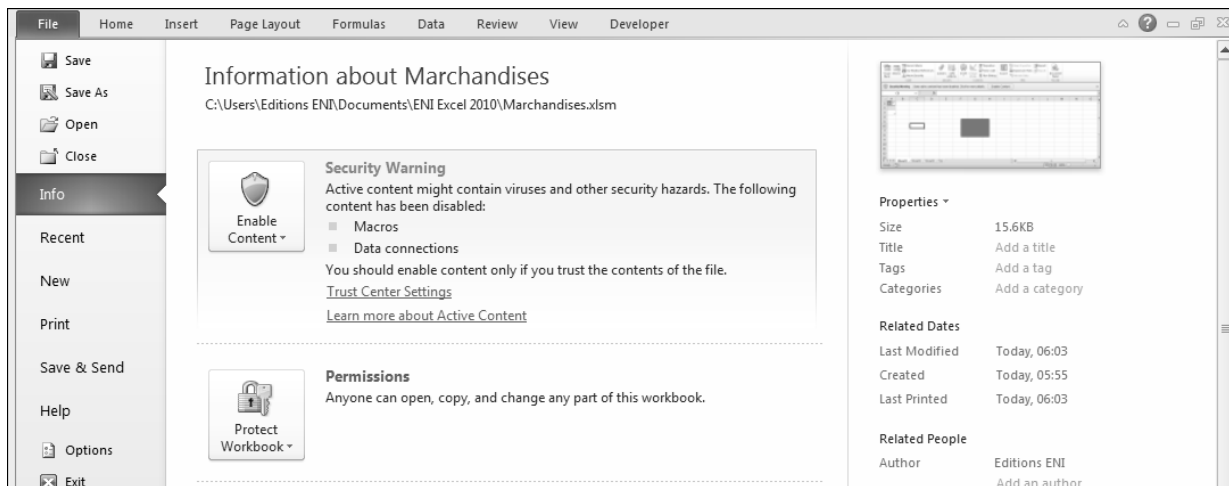
Various advanced features

Enabling macros in the active workbook

Upon opening a file containing macros (file type .xlsm), a **Security warning** message will appear above the formula bar if the macros have not been enabled.



- ☐ If this is the case, to enable the macros, you can click the **Enable Content** button.
- ☐ But you can also open the **File** tab and click the **Info** option.



Click the **Enable Content** button in the **Security Warning** and then click the **Enable All Content** option to enable the content of this workbook indefinitely. To enable the content for this session only, then click on **Advanced Options** and then **Enable content for this session**, then confirm with **OK**.