

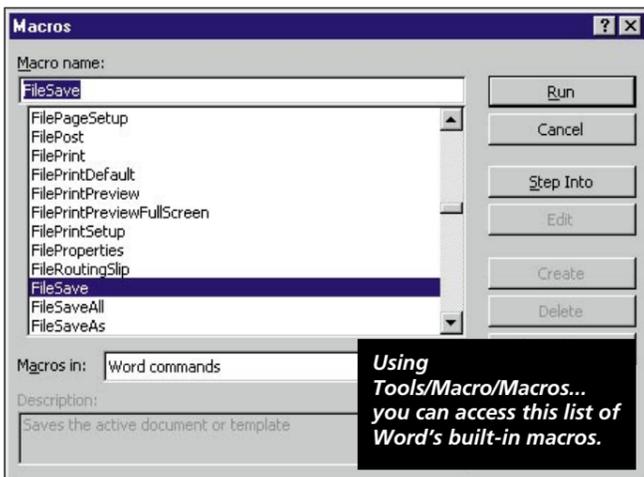
In the second part of this class, Terry Pinnell explains how to assign macros to tool buttons and introduces Visual Basic Editor for use in Excel and other Office 97 applications



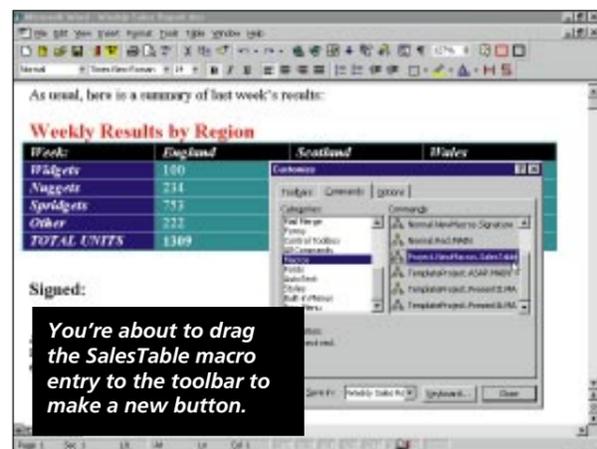
Macros made easy

Last month we covered the basics of using Microsoft Word's macro recorder. In this second and final part of the class we'll first deal with a few other essentials of naming and assigning your macros. Then we'll dip a toe in the waters of the Visual Basic Editor for both Microsoft Word and Excel.

You'll recall that when you recorded your first macro in part one of this class, you were prompted at the start to give it a name. An important point to watch is to avoid giving it the same name as one of Word's built-in macros. To see a list of these, the very building blocks of Word, click 'Macro' on the Tools menu, then click 'Macros...', and open the 'Macros in' drop-down list and choose 'Word Commands'. You'll see the names of nearly a thousand macros. Some, like 'FileSave' and 'Close', will be very familiar, but many are much more specialised. When you're comfortable with the basics, it would be



a good idea to return to this list to try out some of these macros. They are all at your disposal for incorporating into your own macros – which gives you tremendous potential versatility once you've invested the effort. But for this introductory class, just use the list to avoid inadvertent duplication of names, and then close it.

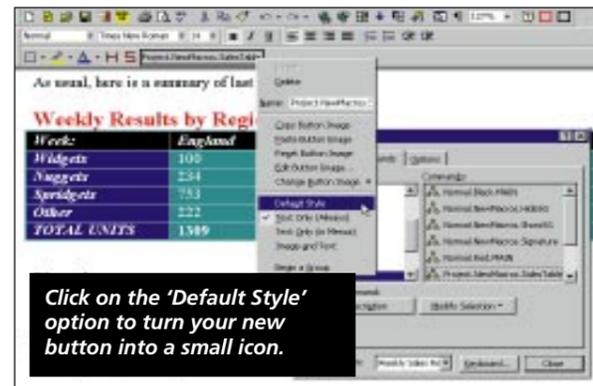


To make it easier to run any macro you can assign it to either a shortcut key or a tool button, instead of having to use the menu. You can make the assignment either before you actually record it, or at any time later; the procedure is very similar in either case. I prefer to get the recording done first and assign it later, if I think it's worth a permanent home on an already crowded toolbar.

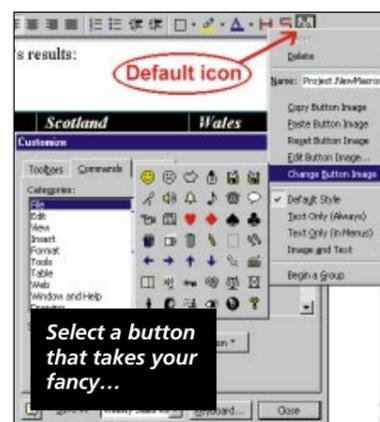
Suppose you want a button to run the table macro that you built in the last class. Unlike your macro for entering a signature, which you assigned to normal.dot so that it worked on all documents, you assigned this SalesTable macro only to

Weekly Sales Report.doc. So first open that, and make sure that the toolbar to which you want to add the new button, say the Formatting toolbar, is visible and has spare room. On the Tools menu, click 'Customize', then the 'Commands' tab. In the 'Categories' box, click 'Macros', about three quarters of the way down. Then, in the 'Commands' box on the right, click the macro's name, 'Project.NewMacros.SalesTable'. Note that the full name usually includes Visual Basic's project name as a prefix, but you don't need to worry about that here – the name you assigned to the macro should always be easy to spot. Now drag that name from the 'Commands' box to the formatting toolbar. Make sure it touches an existing tool button's edge, or it won't 'stick'.

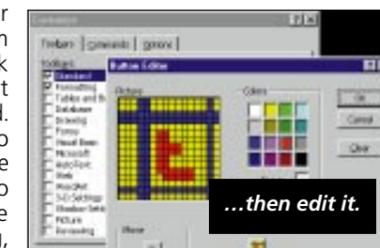
The button appears on the toolbar initially with its full text name, so takes a lot of space, possibly needing continuation below. Right-click on it and select 'Default Style' from the resultant pop-up menu. This turns it into a small standard graphic icon with no text. Close the 'Customize' dialog.



You will probably want to change this icon to something more appropriate; for this, again click 'Customize' on the Tools menu and then right-click the button. You can then choose 'Change Button Image', and select an icon that is to your taste. If there are none that are remotely relevant you can alternatively create your own.



To make your own icon, from the right-click menu select 'Edit Button Image'. Then use the mouse to paint each pixel. Make sure that no one who thinks that you are 'working' is watching, as this looks far too much like having fun. When you are satisfied, click 'OK' and then close 'Customize'. Your personalised button is now ready for use.

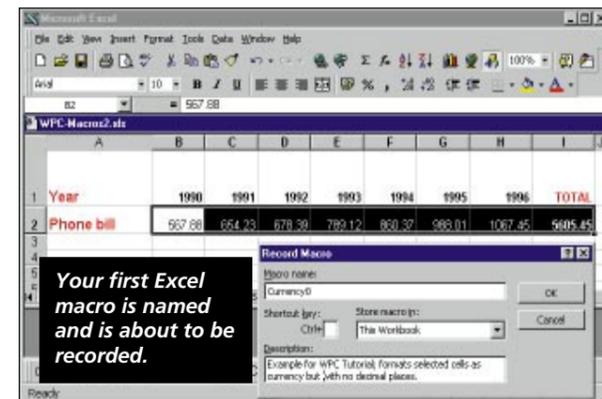


Now let's leave Word, and open Excel. We'll introduce Visual Basic Applications (VBA) gently by recording a simple formatting macro. While many Excel macros would be inappropriate elsewhere, this one could be useful in Word too, although it's chosen mainly for simple clarity.

If you already have a worksheet containing some cells hold-

ing numbers with two decimal places, then open it to use here. Otherwise, open a new workbook and worksheet and enter a few cells with appropriate numbers. Suppose you often want to format such cells as currency, but without the spare pence. It's easy enough using the menu, by setting the decimal places to zero, but it takes up to five clicks. You can get it down to three using the built-in tool buttons, one click on 'Currency' and two on 'Decrease Decimal'. But let's knock up a quick macro to do it with one click.

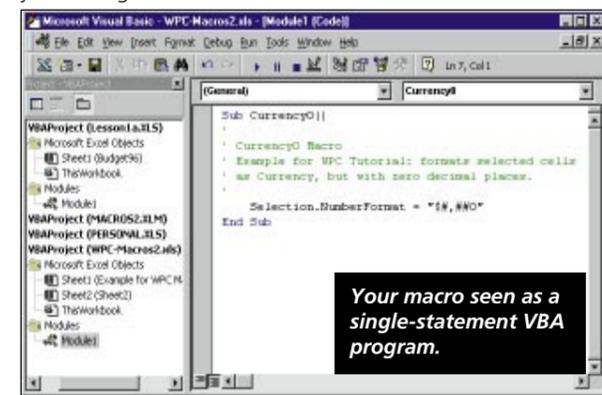
With the cells selected, follow essentially the same procedure as in Word. On the Tools menu, select 'Macro', then click 'Record New Macro'. Type an appropriate name such as 'Currency0' in the 'Macro name' box, and enter a brief description. For now, leave the shortcut field blank and accept the default 'Store macro in' entry as 'This Workbook'.



From Excel's Format menu, choose 'Cells', then the 'Number' tab. In the 'Category' list select 'Currency', change the 'Decimal Places' field from its probable default of 2 to 0, and then click OK. Then click the 'Stop Macro' button – and that's your macro duly recorded. Try it out on other cells if you want.

To see what it looks like in VBA, you should click Tools/Macro/Macros... (or use the handy shortcut Alt+F8), highlight 'Currency0' in the 'Macro name' list, and click 'Edit'. Details might vary a little if you're using an earlier version than Excel 97 as has been used here. But you should now see your macro in the Code window on the right. The lines beginning with apostrophes are comments, the top line beginning with Sub stands for subroutine, and the finishing line is always End Sub. The one statement line that does the work is interpreted from right to left, as: 'Make "\$#,##0" the number format of the selection.'

If you now want to edit the macro without re-recording it, so that it shows, say, one decimal place, then just alter the 0 to 1 and close the VBA application, which automatically saves your changes.



We've barely scratched the surface of the subject, of course, but I do hope that this month's and last month's short tutorials will get you started on some productive and enjoyable macro-making.