

Creating Macros in Microsoft Excel 2003

This short course will introduce you to macros in Excel 2003. You'll learn how to create your own macros using the macro recorder, make them available across workbooks, and create shortcuts and toolbars for easy access. You'll also learn how to edit macros using the Visual Basic Editor and how to find online sources of prewritten macros.

Even if you have some Excel experience under your belt, the term macro might still sound a little scary. If so, you're in the right place! This knowledge file explains what macros are, why they're good to have, how to create and edit them, and how to set them up to be easy to use.

What's a Macro?

As you're planning your day, you probably put together a to-do list, if not on paper then at least in your head, right? For example, take out the trash, put a new liner in the trash can, bring in the mail, and so on.

Imagine what your life would be like if you had a robot that could execute your to-do list each day, with a single command from you. You could say, "Process my list!" and everything would get done exactly the same way you had done it on previous days.

Well, welcome to the future, with Excel as your robot. A macro is a to-do list that you give to Excel to perform. When you run a macro, Excel goes through your list step by step, processing each instruction automatically. With a macro, you can automate complex processes to save yourself the time and tedium of repeating them.

For example, suppose that you frequently apply the following formatting to cells:

1-point dashed, dark blue border around the outside of the cell
Red, bold, Arial 13.5-point text in the cell
Yellow patterned background in the cell

Looking at this list, you probably see immediately that the formatting specifications represent a lot of work. A dashed border is nonstandard, as is a yellow patterned background, so both of those involve opening dialog boxes. And because 13.5 is a nonstandard font size, you need to enter that manually too. What a pain!

The good news is that you can create a macro that will apply all of that formatting in a single step.

Ways to Create a Macro

There are two ways to create a macro: record it or write it.

Recording a macro is the simplest and most popular method. You turn on the Record Macro feature, and from that point on, Excel records every keystroke and every mouse

movement and places them in the macro's list of commands. Then when you turn the Record macro feature off, it stops recording.

If you're familiar with VBA (Visual Basic for Applications), a programming language, you can write your own macros from scratch by entering the commands manually in the Visual Basic Editor utility. This method requires that you know the correct syntax for each command, though, so it's not for beginners.

One good way to familiarize yourself with the proper command syntax for a macro is to record one, and then open it in the Visual Basic Editor to examine how Excel constructed it. Recording Your First Macro

Let's start by creating that sample macro described in the preceding section. Here's what the macro should apply:

1-point dashed, dark blue border around the outside of the cellRed, bold, Arial 13.5-point text in the cellYellow patterned background in the cell

We'll call the macro CustomFormat1. Follow these steps to create it.

Select Tools > Macro > Record New Macro. The Record Macro dialog box shown in

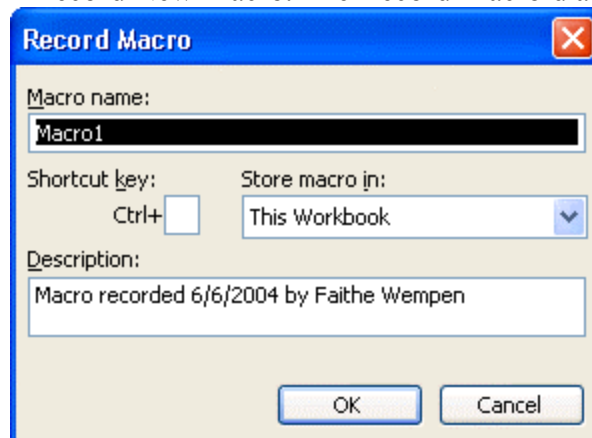


Figure 1.

Figure 1: The Record Macro dialog box.

In the Macro name text box, type CustomFormat1 to replace the default name. In the Shortcut key text box, enter M. (You can assign other shortcut keys if you like; M is just for this example.) If you assign a key to a macro that's already in use in Excel, the macro will take priority for as long as the workbook is open to which it's attached. So, for example, if you use C (Ctrl + C), you can't use Ctrl + C to copy text to the Clipboard. If you accidentally assign a key to the macro that you later decide you want to use for its original purpose, you can edit the macro properties. You learn how to do that later in this knowledge file.

In the Store macro in text box, open the drop-down list and select Personal Macro Workbook. This step enables the macro to be used in other workbooks besides the current one.

In the Description text box, replace the default text with the following: Applies 1-point dashed dark blue border, red bold Arial 13.5 point text, yellow patterned background. Click OK. The Macro toolbar appears in the center of the screen.

Now you apply formatting to the current cell. Select Format > Cells, and then click the Border tab. Open the Color drop-down menu and click the dark blue color.

In the Style area, select a dashed style. Click the Outline button. Click the Font tab, and then under Font style, select Bold.

Open the Color drop-down menu and select red. In the Size text box, type 13.5. Click the Patterns tab, and in the top section of the Colors list, click the bright yellow color.

Open the Pattern drop-down menu and select a pattern (any). Click OK.

Click the Stop Recording button (square) in the Macro toolbar.

Your macro is recorded and ready for testing.

Take Good Notes Running a Macro

There are several ways to run your macro. The easiest way is to simply press the shortcut key combination you specified when you created it. In our example, the shortcut key combination was Ctrl + M, so do the following:

Click in a different cell, one that's not currently formatted the way you set up your macro. Press Ctrl + M. The formatting is applied.

If you did not assign a shortcut key combination to the macro, use the following method instead to run it:

Select Tools > Macro > Macros. The Macro dialog box appears. Click the desired macro (if there's more than one), as shown in Figure 2, and then click Run.

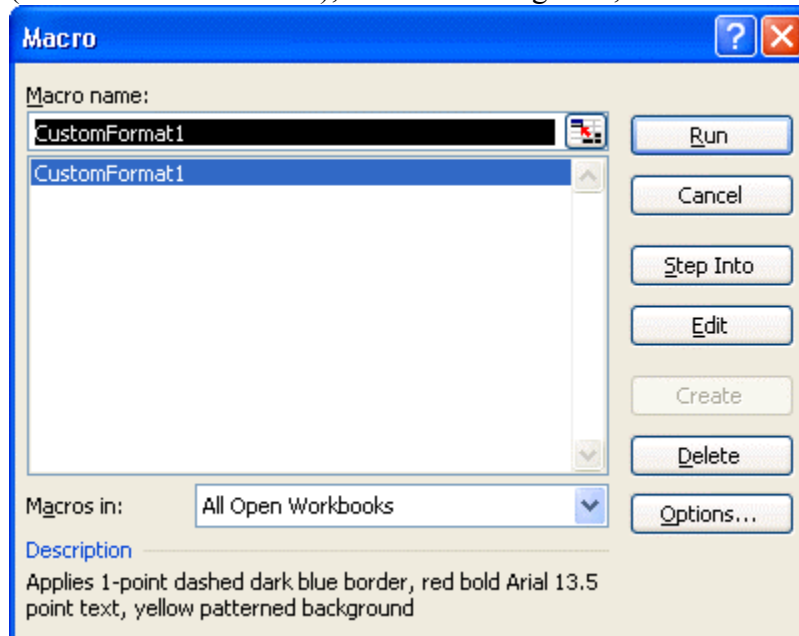


Figure 2: Run a macro from the Macro dialog box.

You'll learn some other ways to run macros later in the course, but first let's answer a couple of common questions.

Where Are Macros Stored?

When you created the macro, you specified where it would be stored. If you decide to store it the current workbook only, that's where it is. Whenever you use the current workbook, it's there; if you open another workbook, it isn't.

However, if you decided to store it in the Personal Macro Workbook, the macro is stored in a workbook called Personal.xls at this location:

```
\Documents and Settings\yourname\Application Data\Microsoft  
Excel\XLSTART\Personal.xls
```

This is a hidden workbook by default, but you want to unhide because you'll edit it later in this knowledge file.

Although Personal.xls is hidden, it does not mean that it's hidden in the same sense as a hidden file or folder in the file system. It's hidden in Excel, so it does not appear on the list of open workbooks in the Window menu.

To unhide the Personal.xls workbook, follow these steps:

Close Excel. After you've saved your current workbook, you're prompted to save your changes to the personal macro workbook. Click Yes. This creates Personal.xls. Restart Excel. The Personal.xls workbook is automatically opened (but hidden). Select Window > Unhide. The Unhide dialog box appears. Click OK to unhide Personal.xls. (Optional) Open the Window menu to confirm that it now appears there, unhidden.

How Can I Change the Shortcut Key?

A macro has two basic properties to it: description and shortcut key. You can edit these without changing the content of the macro -- that is, without changing how it operates. Changing the name of the macro is a bit more difficult, and you'll find out about that later in this knowledge file.

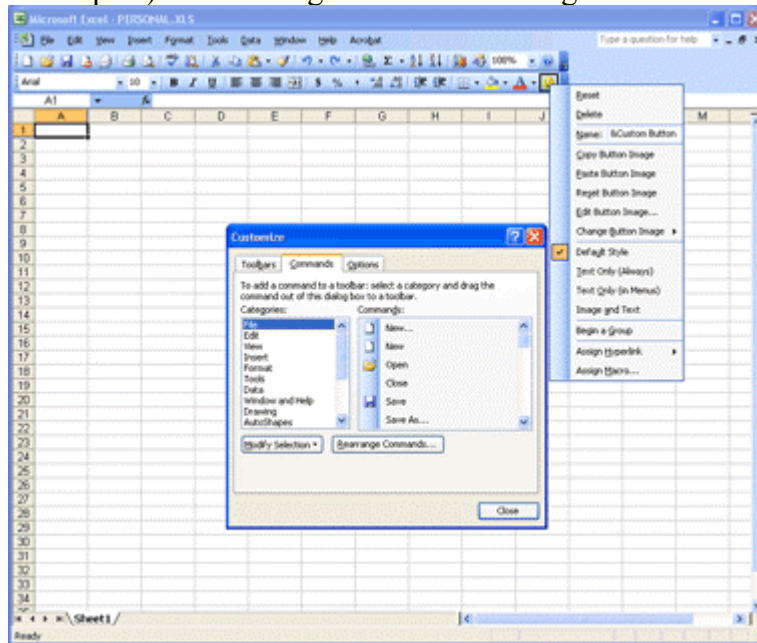
To change a macro's properties, follow these steps:

Select Tools > Macro > Macros. The Macros dialog box appears. Select the desired macro, and then click Options. In the Macro Options dialog box, change the shortcut key assignment if desired. To remove the shortcut key, delete the contents of the Shortcut Key text box. Change the description if desired. Click OK.

Assigning a Macro to a Menu or Toolbar

Many people like to create an entry on a toolbar or menu for a macro, so they don't have to remember its shortcut key or go through the Macro dialog box. This is fairly easy to do. To add your newly created macro to both a toolbar and a menu, follow these steps:

Select Tools > Customize. The Customize dialog box appears. Click the Commands tab. In the Categories list, select Macros. Drag the Custom Button to any existing toolbar. A smiley face button appears there. Right-click the smiley face button (on the toolbar) to open the menu shown in Figure 3. (This menu is applicable only while the Customize dialog box is open.) Select Assign Macro. The Assign Macro dialog box



appears.

Figure 3: Right-click a button to display its menu.

([enlarge image](#))

Select the macro from the list, and then click OK.

(Optional) To change the button image, right-click the smiley face button and select Change Button Image and pick a different picture. (Alternatively, you can select Edit Image and draw your own icon for it.)

(Optional) To make the button appear as text rather than graphics, right-click it and select Text Only (Always).

(Optional) To change the button's text, right-click it and enter new text in the Name box. (For our example, you might enter CustomFormat1.) The & (ampersand) character indicates which letter should be the underlined letter in the text. You can hold down the Alt key and type the underlined letter as a shortcut for selecting that command. This is primarily an issue on menus rather than toolbars.

Hold down the Ctrl key, drag the button from the toolbar to the Tools menu, and pause over the Tools menu until it appears. Then drag and drop it at the bottom of the Tools menu. (Because you held down Ctrl when you dragged, it makes a copy.) Click the Close button in the Customize dialog box. Check the Tools menu to confirm that your macro is there and that it works. Click the macro's button on the toolbar to confirm that it works from there. (Optional) Reopen the Customize dialog box and drag the macro off the toolbar and off the menu to remove it from those locations. Creating a Macros Menu or Toolbar

If the preceding section piqued your interest in creating customized menus and toolbars, let's explore that a little further. You can actually create entire new menus and toolbars, and use them to hold large quantities of macros.

For example, use the following steps to create a My Macros toolbar, and a Macros menu:

Select Tools > Customize. On the Toolbars tab, click New. Type My Macros and click OK. A new, empty toolbar appears (floating). You can drag it to any side of the screen (top, bottom, right, or left) to dock it if desired.

Click the Commands tab, and create a button for your macro on the new toolbar just as you did in the preceding section.

On the Commands tab, select New Menu from the Categories list.

Drag New Menu from the Commands list to the menu bar, which is to the right of the existing menu names.

Right-click the new menu name (on the menu bar) and in the Name text box, type &Macros. (Remember, the & character indicates the underlined letter.) Create a menu command for your macro on the new menu, as you did in the preceding section.

Click Close to close the Customize dialog box.

Get Organized Examining and Editing a Macro

Excel uses the Visual Basic Editor for macro editing. If you've used the Visual Basic programming language before, this environment will seem very familiar to you. If you haven't, don't panic -- this knowledge file doesn't delve too deep into it.

Examining a macro in the Visual Basic Editor can be a great teaching tool for understanding how Excel macros work. So even if you don't need to edit your macro, you might find following along with this section to be worthwhile.

Let's edit the macro you created earlier. Suppose you want to make the following changes to it:

Rather than 13.5 points, you would like the text to be 15 points. You do not want the text to be bold.

Follow these steps to edit the macro:

From the Window menu, select Personal.xls. Select Tools > Macro > Macros. Click the desired macro, and then click Edit. The Visual Basic Editor window appears, as shown in

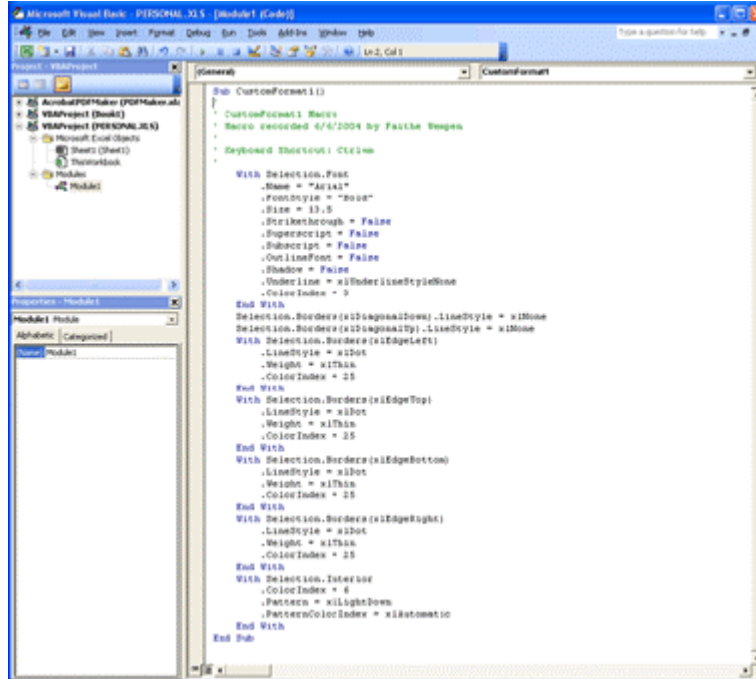


Figure 4.
Figure 4: The Visual Basic Editor.
([enlarge image](#))

Locate the `.Size = 13.5` line and change the number to 15. Locate the `.FontStyle = "Bold"` line and delete it. Take a look at each of the remaining lines in the macro, and try to determine what each of them does. (This is explained in detail in the next section.) When you're finished, close the Visual Basic Editor window. When prompted to save changes, click Yes.

(Optional) Hide Personal.xls again. To do so, view it and then select Window > Hide.
Examining a Typical Macro

Table 1 breaks down the version of the macro created earlier in the lessons and explains each line. Depending on the selections you made, this version may be slightly different from yours, but the basics should be the same.

The Solution for Growing Files Sharing Macros with Other Workbooks

You've already seen one of the best ways of sharing a macro with other workbooks: Simply save it in the Personal.xls workbook, which is loaded automatically each time Excel starts. This method makes the macro(s) available to all workbooks.

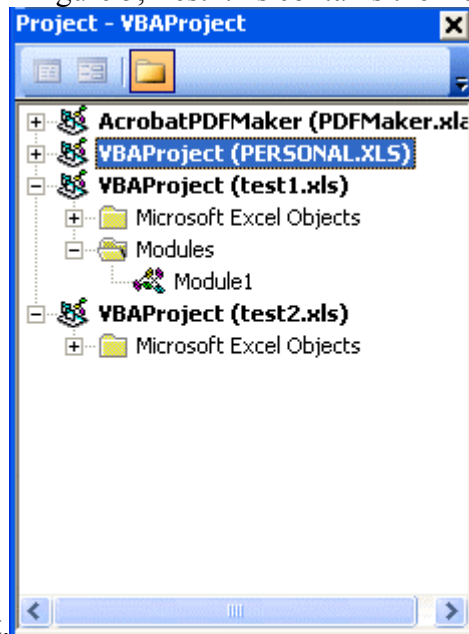
Why is this? Because macros do not have to be in the active workbook to be usable -- they just need to be in any open workbook. So because Personal.xls is set to open itself automatically -- and then hide -- every macro in it is always available.

There may be times, however, when you want to limit a macro to certain workbooks only. For example, you might have a macro that inputs a formula that your company uses to calculate pay, and you don't want that formula to be common knowledge. In cases like this, you should create the macro in the current workbook only, and then copy it to other workbooks as needed, as described in the following section.

Copy a Macro between Workbooks

When you copy a macro, you're actually copying the entire macro module for that workbook, which might contain more than one macro. To copy a macro module, follow these steps:

Open both workbooks -- the one containing the macro, and the one in which you want to place it. Start in the workbook containing the macro and select Tools > Macro > Visual Basic Editor. In the upper-left pane, expand the folder trees for both workbooks so that all modules are visible. For example, in Figure 5, Test1.xls contains the macro and



Test2.xls is where you want to copy it. Figure 5: Copy a macro between workbooks through the Visual Basic Editor. Drag-and-drop the Module1 entry from the source workbook onto the destination workbook's name. Close the Visual Basic Editor. The macro module has been copied.

Macro Security

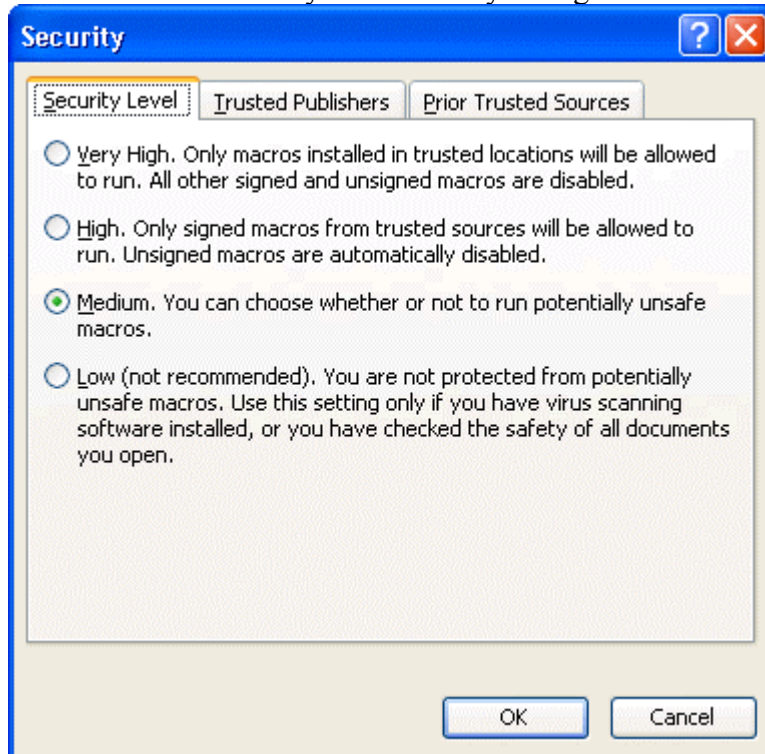
Macros are a security risk because they're VB applications, and VB applications can be used to spread viruses. Therefore, when you set security to High to block potentially harmful macros in Excel, you also block your useful macros.

Excel 2003 has several features for helping you manage security settings. Much of this is beyond the scope of this knowledge file, but here are the basics.

Changing Excel's Security Setting

First things first -- decide how safe you want to be in Excel versus how much macro functionality you want to allow. To change your Excel security setting, follow these steps:

Select Tools > Macro > Security. The Security dialog box shown in Figure 6



appears.

Figure 6: Security dialog box.

Select a security setting. Medium is the best choice for most systems. Click OK.

These security settings refer to signed and unsigned macros. A signed macro is one created by someone who signed it using a security certificate; professional macro publishers do this. For example, any macros you download from Microsoft are signed. The signature verifies the identity of the original author and also verifies that the macro has not been altered since it was signed. It's usually safe to run signed macros.

The settings also refer to trusted locations. For example, your own hard disk is a trusted location. But if a macro executes a command that goes out to a Web site and downloads a script to run, that's pretty fishy, and it's probably up to no good.

If you take a more advanced course on macros and security, you'll learn more about signing macros and setting up trusted locations, but for now just be aware of what those terms mean.

Adding a Macro Publisher to Your List of Trusted Sources

If you get a lot of signed macros from a certain source, you might want to set up that source as a trusted publisher so you can run the macros even when you have your security settings set to High.

To add a publisher, follow these steps:

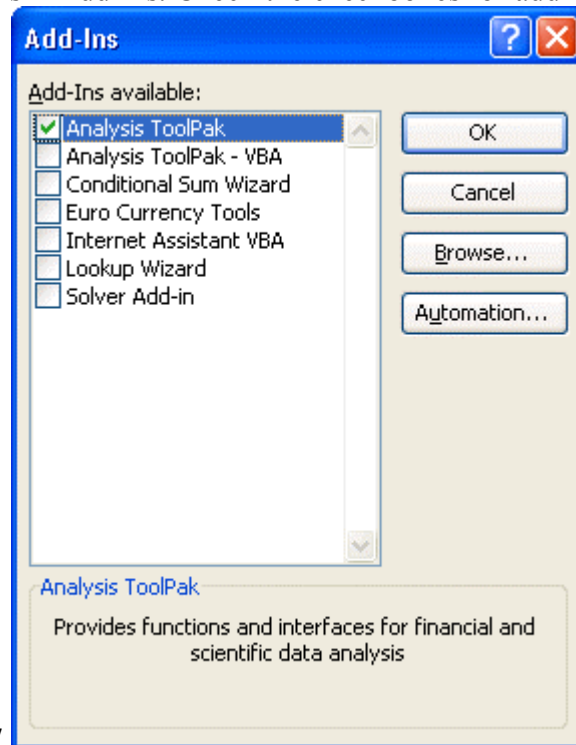
Set the Macro Security setting to Medium or High if it is not already so. Open the file or load the add-in containing the macro(s). A Security Warning dialog box appears. Click Always trust macros from this publisher. Click OK. If you do not see the Security warning box, the macros are not digitally signed, you do not have the security setting at Medium or High, or this publisher is already trusted. Share Documents Securely Add-Ins

Add-ins are related to macros; they're modules that extend Excel's functionality by adding new capabilities, new dialog boxes, new Wizards, and so on. Microsoft provides several add-ins with Excel, and all you need to do is enable them.

Why doesn't Microsoft make all these add-ins part of Excel so that you don't have to mess with enabling them? Because most people don't need them, and having them loaded can make Excel take longer to start up and longer to perform some operations.

To enable an installed add-in, follow these steps:

Select Tools > Add-ins. Check the checkboxes for add-ins you want to enable, as shown



in Figure 7.

Figure 7: Enable add-ins.

Click OK.

After you've enabled an add-in, you'll probably notice extra commands on certain menus or extra buttons on certain toolbars. For example, when you enable the Analysis Toolpak from Figure 7, a Data Analysis command appears on the Tools menu that was not there before. The add-in not only contains the macros and other components, but also the menu customization.

If you download add-ins from the Internet, they typically come with Setup utility programs that place them on the list of available add-ins (shown in Figure 7). But in case you need to add one, you would click the Browse button in Figure 7 to locate it. Sources of Macros and Add-Ins

The Internet is a rich source of macros and add-ins, and most are free for the taking. Here are some places to begin looking:

[Microsoft Office online](#) : In Excel, select Help > Microsoft Office Online, or open a Web browser and go to [Microsoft Office Online](#). Click Excel, and then browse for downloadable macros and add-ins. [MVPs.org](#) : A jumping-off place for many Web sites produced by industry experts associated with Microsoft products ("Microsoft MVPs"). Although there are many pages on this Web site devoted to advanced Excel techniques, visit the top level to get a sense of the total offerings. [excel-vba.com](#) : A Web site with lots of great information about writing your own macros in VBA. This Web site is designed primarily for people with programming experience. [David McRitchie's Excel Pages](#) : Here's a great Web site put together by one of the Excel MVPs, with a lot of macros and information about Excel in general. Moving On

In this short course, you learned how to record your own macros, and how to store them for convenient access and edit them in the Visual Basic Editor. If you're interested in learning how to write your own macros from scratch, you should look into an introductory course in VB. Such a course will help you understand how macro modules are structured so that you can begin deconstructing recorded macros and figuring out what command syntax to use for writing your own. It will also give you the knowledge you'll need to write fancier macros that include dialog boxes.

Good luck with your continued work with Excel macros!