

IN THIS CHAPTER

- » Understanding how macros do what they do
- » Recording macros for instant playback
- » Using the relative option when recording macros
- » Running the macros you've recorded
- » Changing the macro security settings
- » Assigning your macros to the Ribbon and Quick Access toolbar

Chapter **1**

Recording and Running Macros

Macros enable you to automate almost any task that you can undertake in Excel 2019. By using Excel's macro recorder to record tasks that you perform routinely, you not only speed up the procedure considerably (because Excel can play back your keystrokes and mouse actions much faster than you can perform them manually), but you are also assured that each step in the task is carried out the same way each and every time you perform the task.

Excel's macro recorder records all the commands and keystrokes that you make in a language called Visual Basic for Applications (VBA), which is a special version of the BASIC programming language developed and refined by the good folks at Microsoft for use with all their Office application programs. You can then use Excel's Visual Basic Editor to display and make changes to the macro's VBA code.

In this chapter, you find out how to use Excel's macro recorder to record, test, and play back macros that you use to automate repetitive tasks required when building and using your Excel worksheets and charts. In the next chapter, you find out how to use Excel's Visual Basic Editor to debug and edit the macros that you record, as well

as to create complex macros that run custom functions and set up and run custom Excel applications, complete with their own pull-down menus and dialog boxes.

Macro Basics

You can create macros in one of two ways:

- » Use Excel's macro recorder to record your actions as you undertake them in a worksheet.
- » Enter the instructions that you want followed in VBA code in the Visual Basic Editor.

Either way, Excel creates a special *module* sheet that holds the actions and instructions in your macro. The macro instructions in a macro module (whether recorded by Excel or written by you) are stored in the Visual Basic for Applications programming language.

You can then study the VBA code that the macro recorder creates and edit this code in the Visual Basic Editor, which you open by clicking the Visual Basic command button on the Developer tab (when this optional tab is displayed on the Ribbon) or by pressing Alt+F11.

Recording macros

With Excel's macro recorder, you can create many of the utility-type macros that help you to perform the repetitive tasks necessary for creating and editing your worksheets and charts. When you turn on the macro recorder, the macro recorder records all your actions in the active worksheet or chart sheet as you make them. Note that the macro recorder doesn't record the keystrokes or mouse actions that you take to accomplish an action — only the VBA code required to perform the action itself. This means that mistakes that you make while taking an action that you rectify won't be recorded as part of the macro; for example, if you make a typing error and then edit it while the macro recorder is on, only the corrected entry shows up in the macro without the original mistakes and steps taken to remedy them.

The macros that you create with the macro recorder can be stored either as part of the current workbook, in a new workbook, or in a special, globally available Personal Macro Workbook named `PERSONAL.XLSB` that's stored in a folder called `XLSTART` on your hard disk. When you record a macro as part of your Personal Macro Workbook, you can run that macro from any workbook that you have open. (This is because the `PERSONAL.XLSB` workbook is secretly opened whenever you

launch Excel, and although it remains hidden, its macros are always available.) When you record macros as part of the current workbook or a new workbook, you can run those macros only when the workbook in which they were recorded is open in Excel.

When you create a macro with the macro recorder, you decide not only the workbook in which to store the macro but also what name and shortcut keystrokes to assign to the macro that you are creating. When assigning a name for your macro, use the same guidelines that you use when you assign a standard range name to a cell range in your worksheet. When assigning a shortcut keystroke to run the macro, you can assign

- » The Ctrl key plus a letter from A to Z, as in Ctrl+Q
- » Ctrl+Shift and a letter from A to Z, as in Ctrl+Shift+Q

You can't, however, assign the Ctrl key plus a punctuation or number key (such as Ctrl+1 or Ctrl+/) to your macro.

The Ribbon's View tab contains a Macros command button to which a drop-down menu containing the following three options is attached:

- » **View Macros:** Opens the Macro dialog box where you can select a macro to run or edit (Alt+WMV).
- » **Record Macro:** Opens the Record Macro dialog box where you define the settings for your new macro and then start the macro recorder; this is the same as clicking the Record Macro button on the Status bar (Alt+WMR).
- » **Use Relative References:** Uses relative cell addresses when recording a macro, making the macro more versatile by enabling you to run it in areas of a worksheet other than the ones originally used in the macro's recording (Alt+WMU).

Excel 2019 also enables you to add a Developer tab to the Ribbon. This tab contains a Record Macro and Use Relative References button that you can use in recording your macros. To add the Developer tab to the Excel 2019 Ribbon, you follow these two steps:

1. Click **File** ⇨ **Options** and then click the **Customize Ribbon** tab or press **Alt+FTC**.
Excel opens the Customize the Ribbon pane within the Excel Options dialog box.
2. Click the **Developer** check box in the **Main Tabs** list on the right side of the **Customize the Ribbon** pane in the Excel Options dialog box and then click **OK**.

You can then turn on the macro recorder by doing any of the following:

- » Click the Record Macro button on the Excel Status bar (to the immediate right of the Ready indicator)
- » Click the Record Macro option on the Macros drop-down button on the View tab (Alt+WMR)
- » Click the Record Macro command button on the Developer tab (Alt+LR)

No matter how you turn on the macro recorder, once you do, it records all your actions in the active worksheet or chart sheet as you make them.

To see how easy it is to create a macro with the macro recorder, follow along with these steps for creating a macro that enters the company name in 12-point, bold type and centers the company name across rows A through E with the Merge and Center feature:

1. Open the Excel workbook that contains the worksheet data or chart you want your macro to work with.

If you're building a macro that adds new data to a worksheet (as in this example), open a worksheet with plenty of blank cells in which to add the data. If you're building a macro that needs to be in a particular cell when its steps are played back, put the cell cursor in that cell.

2. Click the Record Macro button on the Status bar or select the Record Macro option on the Macros command button on the View tab or press Alt+WMR.

The Record Macro dialog box opens, similar to the one shown in Figure 1-1, where you enter the macro name, define any keystroke shortcut, select the workbook in which to store the macro, and enter a description of the macro's function.

3. Replace the Macro1 temporary macro name by entering your name for the macro in the Macro Name text box.

When naming a macro, you must not use spaces in the macro name and it must begin with a letter and not some number or punctuation symbol. For this example macro, you replace Macro1 in the Macro Name text box with the name Company_Name.



REMEMBER

Next, you can enter a letter between A and Z that acts like a shortcut key for running the macro when you press Ctrl followed by that letter key. Just remember that Excel has already assigned a number of Ctrl+letter keystroke shortcuts for doing common tasks, such as Ctrl+C for copying an item to the Clipboard and Ctrl+V for pasting an item from the Clipboard into the

worksheet. If you assign the same keystrokes to the macro that you're building, your macro's shortcut keys override and, therefore, disable Excel's ready-made shortcut keystrokes.

- 4. (Optional) Click the Shortcut Key text box and then press the letter of the alphabet that you want to assign to the Ctrl key combination that can run the macro.**

For this example macro, you simply press Shift+C to assign Ctrl+Shift+C as the shortcut keystroke (so as not to disable the ready-made Ctrl+C shortcut).

Next, you need to decide where to save the new macro that you're building. Select Personal Macro Workbook from the Store Macro In drop-down list box to be able to run the macro anytime you like. Select This Workbook (the default) when you need to run the macro only when the current workbook is open. Select New Workbook if you want to open a new workbook in which to record and save the new macro.

- 5. Select the Personal Macro Workbook, New Workbook, or This Workbook item from the Store Macro In drop-down list to indicate where to store the new macro.**

For this example macro, select the Personal Macro Workbook so that you can use it to enter the company name in any Excel workbook that you create or edit.



TIP

Next, you should document the purpose and functioning of your macro in the Description list box. Although this step is purely optional, it is a good idea to get in the habit of recording this information every time you build a new macro so that you and your coworkers can always know what to expect from the macro when any of you run it.

- 6. (Optional) Click the Description list box and then insert a brief description of the macro's purpose in front of the information indicating the date and who recorded the macro.**

Now you're ready to close the Record Macro dialog box and start recording your macro.

- 7. Click OK to close the Record Macro dialog box.**

When you do this, the Record Macro dialog box closes. Click the Use Relative References option on the Macros command button on the View tab or the Use Relative References command button on the Developer tab (when it's displayed) when you want the macro recorder to record the macro relative to the position of the current cell. Doing this often makes a macro more versatile as it enables you to run the macro in areas in the worksheet other than the cells used in its original recording.

8. (Optional) Click the **Use Relative References** option on the **Macros** command button on the **View** tab (Alt+WMU) or click the **Use Relative References** command button on the **Developer** tab (Alt+LU) if you want to be able to play back the macro anywhere in the worksheet.

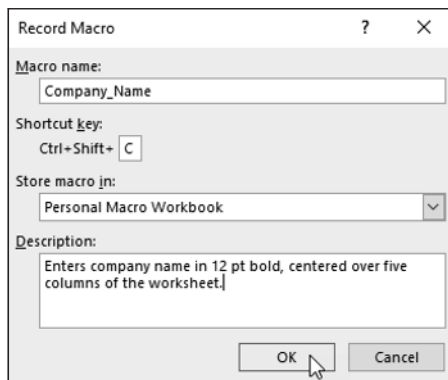
9. Select the cells, enter the data, and click the Excel commands required to perform the tasks that you want recorded just as you normally would in creating or editing the current worksheet, using either the keyboard or the mouse or a combination of the two.

For the example macro, all you do is type the company name and click the **Enter** button on the **Formula** bar to complete the entry in the current cell. Next, click the **Bold** button and then click 12 on the **Font Size** drop-down list on the **Formatting** toolbar. Finally, drag through cells A1:E1 to select this range and then click the **Merge and Center** button, again on the **Formatting** toolbar.

After you finish taking all the actions in Excel that you want recorded, you're ready to shut off the macro recorder.

10. Click the **Stop Recording** button on the **Excel Status bar** or **Stop Recording** option on the **Macros** button's drop-down menu on the **View** tab or the **Stop Recording** button in the **Code group** of the **Developer** tab.

FIGURE 1-1: Getting ready to record the **Company_Name** macro in the **Record Macro** dialog box.



TIP

The **Record Macro/Stop Recording** button to the **Status bar** (to the immediate right of the **Ready** status indicator) is by far the most direct and easiest way to turn the macro recorder on and off. When using it to record a new macro, as soon as you click **OK** in the **Record Macro** dialog box to actually start recording your actions, this **Record Macro** button on the **Status bar** changes to a **Stop Recording** button, and you can click it when you are done recording all the actions to be included in the macro.

Running a macro

After recording a macro, you can run it by doing any of the following:

- » Click the View Macros option on the Macros command button on the View tab or press Alt+WMV.
- » Click the Macros command button on the Developer tab of the Ribbon when it's displayed or press Alt+LPM.
- » Press Alt+F8.

Excel then opens the Macro dialog box, which is similar to the one shown in Figure 1-2. As this figure shows, Excel lists the names of all the macros in the current workbook and in your Personal Macro Workbook (provided you've created one) in the Macro Name list box. Click the name of the macro that you want to play and click the Run button or press Enter.

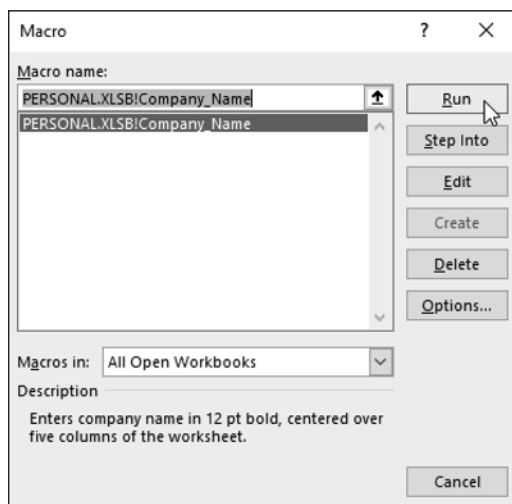


FIGURE 1-2: Selecting the macro to run in the Macro dialog box.

If you assigned a shortcut keystroke to the macro, you don't have to bother opening the Macro dialog box to play the macro: Simply press Ctrl plus the letter key or Ctrl+Shift plus the letter key that you assigned and Excel immediately plays back all of the commands that you recorded.



REMEMBER

Before testing a new macro, you may need to select a new worksheet or at least a new cell range within the active worksheet. When recording cell references in a macro, the macro recorder always inserts absolute references in the macro sheet unless you click the Relative Reference button on the Stop Recording toolbar

before you start choosing the commands and taking the actions in the spreadsheet that you want recorded as part of the macro. This means that your macro enters its data entries or performs its formatting in the same area of the active worksheet (unless the code in the macro itself causes the macro to first select a new area or select a new sheet in the active workbook).

If you run your macro in a worksheet that already contains data in the cells that the macro uses, you run the risk of having existing data and/or formatting overwritten during the macro's execution. Keep in mind that, although you can use the Undo feature to reverse the very last action performed by your macro, most macros perform a series of actions, so you may end up using multiple levels of Undo before you are able to successfully reconstruct your spreadsheet.

Assigning Macros to the Ribbon and the Quick Access Toolbar

Instead of running a macro by selecting it in the Macro dialog box or by pressing shortcut keys you assign to it, you can assign the macro to a custom tab on the Ribbon or a custom button on the Quick Access toolbar and then run it by clicking that custom button.

Adding your macros to a custom tab on the Ribbon

To assign a macro to a custom tab on the Excel 2019 Ribbon, you follow these steps:

- 1. Click File⇒Options and then click the Custom Ribbon button in the Excel Options dialog box (or press Alt+FTC).**
Excel displays the Customize the Ribbon pane in the Excel Options dialog box.
- 2. Click Macros in the Choose Commands From drop-down list box on the left.**
Excel lists the names of the all the macros created in the current workbook and saved in the PERSONAL .XLSB workbook in the Choose Commands From list box.
- 3. Click the name of the custom group on the custom tab to which you want to add the macro in the Main Tabs list box on the right.**

If you haven't already created a custom tab and group for the macro or need to create a new one, you then need to follow these steps:

- a. *Click the New Tab button at the bottom of the Main Tabs list.*
Excel adds both a New Tab (Custom) and New Group (Custom) item to the Main Tabs list while at the same time selecting the New Group (Custom) item.
- b. *Click the New Tab (Custom) item you just added to the Main Tabs.*
- c. *Click the Rename button at the bottom of the Main Tabs list box and then type a display name for the new custom tab before you click OK.*
- d. *Click the New Group (Custom) item right below the custom tab you just renamed.*
- e. *Click the Rename button and then type a display name for the new custom group before you click OK.*

4. **In the Choose Commands From list box on the left, click the name of the macro you want to add to the custom group now selected in the Main Tabs list box on the right.**
5. **Click the Add button to add the selected macro to the selected custom group on your custom tab and then click the OK button to close the Excel options dialog box.**

After you add a macro to the custom group of a custom Ribbon tab, the name of the macro then appears on a button sporting a generic icon (a programming diagram chart) when you select the custom tab on the Ribbon. All you have to do to run the macro is to click this macro command button.

Adding your macros to custom buttons on the Quick Access toolbar

To assign a macro to a custom button on the Quick Access toolbar, you follow these steps:

1. **Click the Customize Quick Access Toolbar button at the end of the Quick Access toolbar and then click More Commands from the drop-down menu to open the Quick Access Toolbar tab of the Excel Options dialog box.**
2. **Select Macros from the Choose Commands From drop-down list box.**
Excel lists the names of all the macros created in the current workbook and saved in the PERSONAL.XLSB workbook in the Choose Commands From list box.
3. **Click the name of the macro to add to a custom button on the Quick Access toolbar in the Choose Commands From list box and then click the Add button.**
4. **Click OK to close the Excel Options dialog box.**

After you close the Excel Options dialog box, a custom button with a generic macro icon (picturing a standard command flowchart icon) appears on the Quick Access toolbar. To see the name of the macro assigned to this custom macro button as a ScreenTip, position the mouse pointer over the button. To run the macro, click the button.



REMEMBER

Keep in mind that you can also assign macros to the graphic objects you add to your worksheet, including shapes you draw, clip art, and pictures you import. To assign a macro to a graphic object, right-click it and then click the Assign Macro option from its shortcut menu. Then click the name of the recorded macro to run in the list box of the Assign Macro dialog box and click OK. Thereafter, when you position the mouse pointer over the graphic object, the pointer becomes a hand with a pointing index finger (just like when you assign a hyperlink to a graphic — see Book 4, Chapter 2), indicating that you can click it to run the macro.

Macro Security

Excel 2019 uses a system called Microsoft Authenticode that enables developers to authenticate their macro projects or add-ins created with Visual Basic for Applications by a process referred to as *digital signing*. When you run a macro in your worksheet that's not saved in the trusted locations on your computer, such as the Templates and XLSTART folder in your user area on the computer, Excel checks to see whether the macro is digitally signed and that the signature is both valid and current. The macro's developer must have a certificate issued by a reputable authority or a trusted publisher.

If the program cannot verify a macro's digital signature (perhaps because it doesn't have one) or the trustworthiness of its macro publisher, the program then displays a security alert on the message bar underneath the Excel Ribbon. This alert area contains an Enable Content and a Trust Center command button. You can then click the Enable Content button to ignore the alert and go ahead and run the macro, assuming that you can vouch for the macro's publisher and are sure that running the macro poses no security risk to your computer. You click the Trust Center command button in the security alert on the message bar to open the Trust Center dialog box, where you can add to the trusted locations on your computer system and change the macro security settings.



REMEMBER

You can also open the Macro Settings tab of the Trust Center dialog box by clicking the Macro Security command button on the Developer tab (Alt+LAS) or clicking the Trust Center Settings button on the Trust Center tab of the Excel Options dialog box (Alt+FTT).

By default, Excel selects the **Disable All Macros with Notification** option button on the **Macro Settings** tab of the **Trust Center** dialog box. When this setting is selected, all macros that are not saved in one of the trusted locations are automatically disabled in the worksheet, but you do get a security alert each time you try to run one of these macros that enables you to ignore the alert and go ahead and run the macro by clicking the **Enable Content** button.

The **Macro Settings** tab of the **Trust Center** dialog box also contains these other option you can select:



WARNING

- » **Disable All Macros without Notification** to disable all macros not saved in one of your computer's trusted locations and all security alerts so that you and the other users of the worksheet have no way to ignore the alert and run the macro. Select this option when you don't trust someone else's macros and want to make it impossible to run a macro carrying a computer virus.
- » **Disable All Macros with Notification** (the default) to control the disabling of macros not saved in one of your computer's trusted locations and security alerts. When you select this setting, worksheet users can't ignore the alert and run the macro. Select this option when you want to maintain control over running potentially untrustworthy macros.
- » **Disable All Macros except Digitally Signed Macros** to automatically enable digitally signed macros from a publisher that you've indicated is trustworthy and to disable all macros that are not digitally signed without notification. When you select this option and try to run a digitally signed macro that's not from a publisher you've indicated is trustworthy, Excel displays an alert in the message bar with a **Trust All Documents from this Publisher** button that you can select, thereby adding the publisher to the trusted list.
- » **Enable All Macros (Not Recommended; Potentially Dangerous Code Can Run)** to throw all caution to the wind and allow all macros to run in any worksheet you open — this is one option you never ever want to select, because it could cause serious damage to your machine!
- » **Trust Access to the VBA Project Object Model** to grant access to Visual Basic for Applications object model (see Book 8, Chapter 2) from an automation client.

To change the trusted locations on your computer, you need to click the **Trusted Locations** tab in the **Trust Center** dialog box. You can then use these options to change the location settings:

- » **Add New Location:** Use this command button to open the Microsoft Office Trusted Location dialog box, where you select a new folder on your computer as a trusted location either by entering its directory pathname in the **Path** text

box or selecting it with the Browse button. Click the Subfolders of This Location Are Also Trusted check box if you want all subfolders within the designated folder to be included as trusted locations.

- » **Allow Trusted Locations on My Network (Not Recommended):** Click this check box so that you can designate folders to which you have access on your local network as trusted locations using the Add New Location command button (as described in the immediately preceding bullet point).
- » **Disable All Trusted Locations:** Click this check box to immediately disable all the folders currently designated as trusted locations and allow only macros from publishers designated as trustworthy to run in Excel.