

An Introduction to Macros

Using macros enables you to repeat tasks much more efficiently than tediously performing each step over and over. A *macro* is a set of instructions that you use to automate a task. For example, if you want to take each column of numbers, convert them to currency, and then add them together, you can create a simple macro to perform this task. The typical Excel user has a series of tasks that they perform frequently. By creating a macro to perform those tasks, you only require a simple keystroke to repeat the tasks.

You can create macros to perform a task as simple as adding two numbers, or as complex as creating a whole user interface within Excel. To do so, you can employ one, or a combination, of two different methods: You can use the Macro Recorder, or you can manually write a macro using

the Visual Basic Editor. Although many macro users rarely venture past the Macro Recorder, this book shows you how to harness the power of *Visual Basic for Applications*, or VBA, to create more complex macros. No matter how simple or complex a macro, you write them all using VBA.

Macros are a term common to the spreadsheet world. All spreadsheet packages on the market provide the ability to create macros to automate tasks, and Excel is no exception. Although all Microsoft Office products provide the ability to create macros, they are best suited for Microsoft Excel.

As an Excel user, you may have a series of tasks that you perform frequently. By creating a macro to perform complex or repetitive tasks, you can save time by pressing a simple keystroke each time you want to perform the tasks.

Macro History

Macro and Lotus 1-2-3

Macros originated with Lotus 1-2-3 in a fashion similar to the Macro Recorder you find in Excel today. The Lotus macros recorded the keystrokes and enabled you to play them later.

XLM Macro Sheets

Microsoft first entered the macro world with *XLM macro sheets*. XLM macro sheets are just sheets of functions that Excel evaluates in the sequence they exist within the selected macro. Although this macro language was powerful, it was difficult to use. Although Excel still supports XLM macro sheets for compatibility with early versions of Excel, it does not provide the ability to record an XLM macro.

VBA Macros

The addition of Visual Basic for Applications (VBA) macros increased the popularity of Excel within the spreadsheet world. Essentially a subset of the popular Visual Basic language, VBA is familiar to many developers.

Also, VBA brings a much more powerful macro development platform to Excel than the macro development environment in other spreadsheet packages.

Excel Macros

Although Microsoft did not originate the concept of macros and spreadsheets, they have definitely built upon it. The combination of the Macro Recorder and VBA makes macro creation a powerful feature of Excel.

Record and Store Macros

Macro Recorder

The *Macro Recorder* provides a great method for creating a macro without writing VBA code directly. The Macro Recorder holds true to its name. Just like a tape recorder, when you turn it on, it records all the events that occur within Excel. Excel takes the recorded events and creates the VBA code necessary to re-create the events. You can modify all macros you create with the Macro Recorder in the Visual Basic Editor. The Macro Recorder works well for creating simple macros, such as a macro that adds a column of numbers, or changes the layout of the page. But due to the fact that the Macro Recorder creates a macro by recording your actions, it cannot create a complex macro such as one that repeats a process until meeting a specific condition or displays a custom dialog box. More complex Excel macros require the use of VBA.

The Macro Recorder does work well in conjunction with the Visual Basic Editor. For example, if you want to create a macro that sums each column of data in your worksheet, you record the macro that sums a column. You then edit the macro in the Visual Basic Editor to run the macro until Excel process all columns. Combining the use of the Macro Recorder and the Visual Basic Editor simplifies the macro creation by allowing Excel to code part of it for you. See the section "Record a Macro" for more information on recording a macro in Excel.

Macro Storage

The Store macro in option on the Macro dialog box instructs Excel where to store the macros you record. Excel provides three different storage locations for your macros: the current workbook, a new workbook, or the Personal Macro Workbook.

You can store a recorded macro to your current workbook, commonly referred to as the active workbook, by selecting the *This Workbook* option. Use this option if you plan to share the workbook with other users. Storing the macros in the active workbook makes macros available to any user who opens the workbook.

You can record a macro to a *New Workbook*. Excel creates the workbook automatically and adds the new macro to it. If you store a macro in another workbook, you need to open that workbook whenever you want to use that macro. You store macros in separate workbooks when you want to store specific types of macros in different workbooks. For example, you may want to place all macros that perform budget calculations in one workbook.

You can record a global macro by selecting *Personal Macro Workbook*, which serves as a common storage location for macros that you expect to use with other workbooks. Excel stores your Personal Macro Workbook as *Personal.xls* in the *XIStart* folder. This workbook does not exist until you store a macro in it. After you create the workbook, it loads whenever you run Excel as a hidden workbook. Excel hides the Personal Macro workbook so that you are unaware of its being open.

The storage location you select for a macro depends on where you want access to the macro. If you create the macro with the Macro Recorder, you can select the storage location of the macro when you create it on the Record Macro dialog box.

See the section "Record a Macro" for more information about storing macros.

Record a Macro

You can use macros to automate a series of steps. The easiest method for creating a macro involves using the Macro Recorder option, which captures everything you do and saves it in a macro module with the name you specify. After you create the macro, you can run the macro again, modify it, or delete it.

Because the Macro Recorder records every action you perform when you use it, consider planning your steps before creating the macro. Because each macro action takes time to record, when you plan out the macro steps, the macro runs faster and more effectively. When you name a macro, use a name that starts with a letter and has no spaces in it; you can, however, use the underscore character to separate words.

Excel creates the macro with either relative or absolute reference to the cell where you apply it. You can specify the cell reference by selecting the Relative Reference button on the Stop Recording toolbar. If you select Relative Reference, the macro uses *relative* references, meaning that it performs the macro based upon the location of the cell. For example, you can have a macro add the values in the first four cells of a worksheet and place the total sum in a cell you select. With *absolute* positioning, however, the macro records in absolute mode and remembers the specific cells you use to record the macro. For example, the macro remembers always to add the same cells, such as A1 through A5, and place the total sum in cell A6. You can toggle between relative and absolute referencing while recording your macro via the Reference button.

Record a Macro

- 1 Click select the worksheet cell to contain the results of the macro.

If you intend to use other worksheet cells in your macro, make sure the cells contain the desired values.

- 2 Click Tools→Macro→Record New Macro.

The Record Macro dialog box displays.

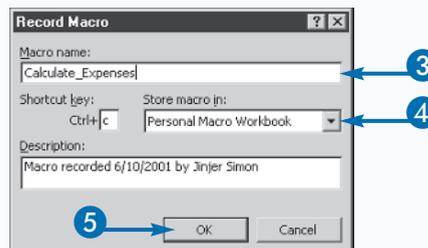
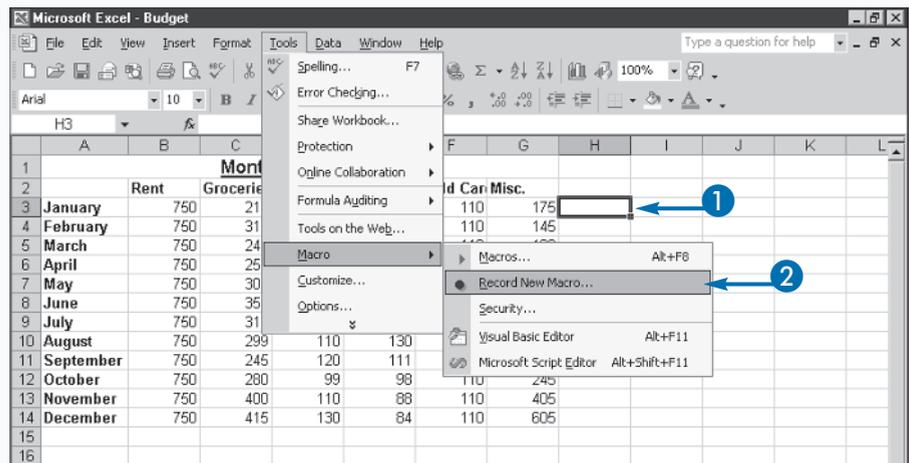
- 3 Type a unique name for the macro.

You can also create a keyboard shortcut for your macro by typing the desired shortcut key in the Shortcut Key field.

- 4 Click here to select a location where you want to store the macro.

Note: See the section "An Introduction to Macros" for more on the three areas to store a macro.

- 5 Click OK.

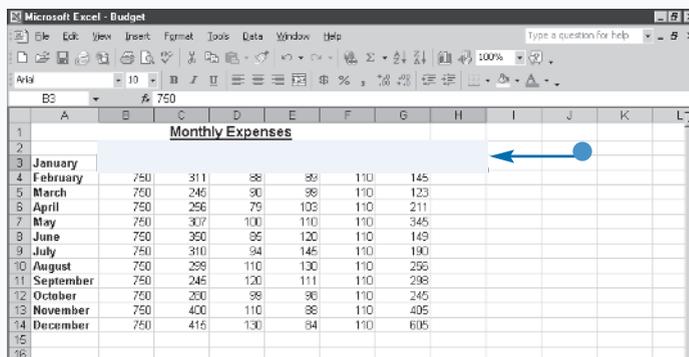
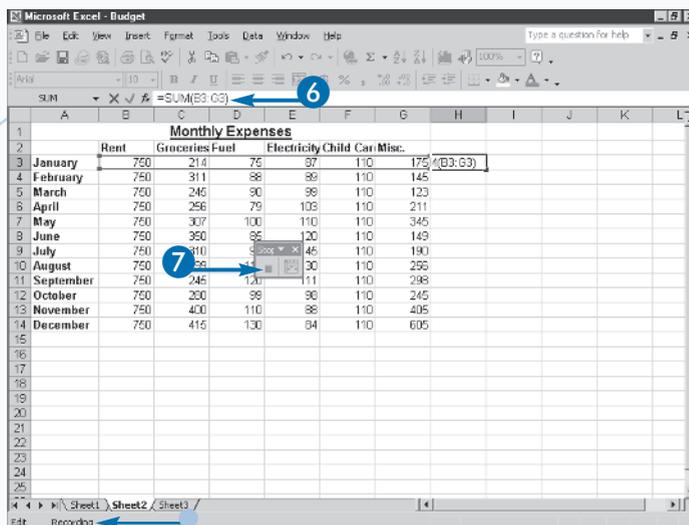


The Stop Recording toolbar appears.

- The status bar reminds you that a macro is recording.

- Press the appropriate key strokes to record the macro.
- When complete, click the Stop Recording button (■).

- Excel records the macro and the Stop Recording toolbar no longer displays on the screen.



Extra

When you create a new macro, you have the option of assigning it to a keyboard shortcut by typing the shortcut in the Keyboard Shortcut box on the Record Macro dialog box. When you do this, the macro runs when you press the Ctrl key and the specified lowercase key simultaneously. If you specify an uppercase letter for the key, you can run the macro by pressing Ctrl+Shift+the specified key. Unfortunately, Excel does not stop you from creating shortcuts that override other predefined Excel shortcut keys. If you specify a shortcut key combination that matches a Microsoft Excel shortcut, your new shortcut overwrites it; each time you press the shortcut keys, your macro — not the Microsoft key combination — runs. For example, the Save command shortcut is Ctrl+s. If you create a macro with a shortcut key of s, your macro runs instead of the Save command when you press Ctrl+s. Excel does use many of the available shortcut keys, so you are bound to overwrite one. Keep in mind that if you use an Excel shortcut, you do not want to create a macro shortcut that overwrites it.

Run a Macro

Excel allows you to run macros in a worksheet that exists either in the current workbook or in any other Excel workbook. However, you can only run a macro from any other workbook as long as the corresponding workbook is open within Excel. When you run a macro, Excel re-creates the recorded steps that you performed to create it, or it runs the VBA code that you created in the Visual Basic Editor. See the section, "Record a Macro" to learn how to record a macro and Chapter 3 for more information on the Visual Basic Editor.

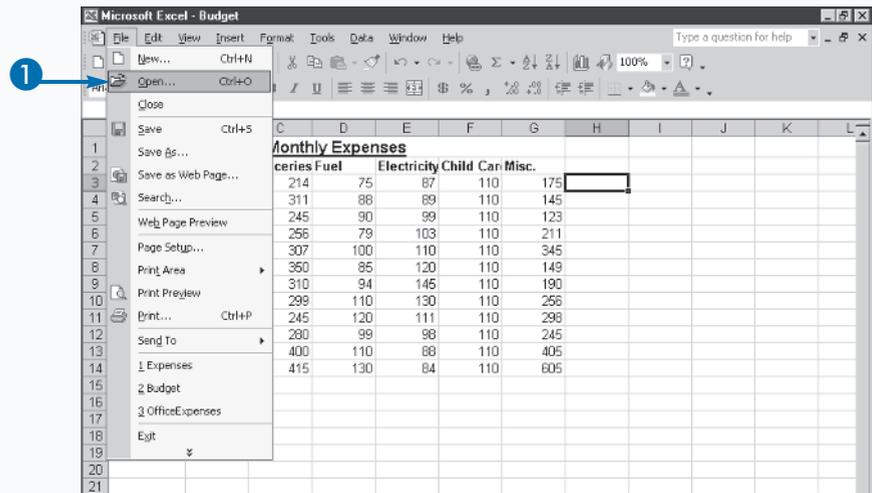
You select macros to run from the Macro dialog box, which lists all currently available macros. Available simply means that Excel can locate the macro in an open workbook. Because Excel only knows about macros in open workbooks, you must open the workbook containing the macro you want to run.

When you create a macro, Excel stores it in one of three locations, the current workbook, a new workbook, or the Personal Macro Workbook. Excel opens the Personal Macro Workbook as a hidden file each time you run Excel, and makes all macros you store there available to run with any workbook. If you store a macro in a separate workbook, you must open the workbook containing the macro in Excel. You can learn more about creating a macro in the section "Record a Macro."

To run a macro from another workbook, you must have a macro from a signed source, or you must set your macro security to either Medium or Low. The default macro security level, High, requires that all macros from other sources be signed. Setting your macro security to Medium or Low lets you run unsigned macros. See section "Set Macro Security" for more information about macro security.

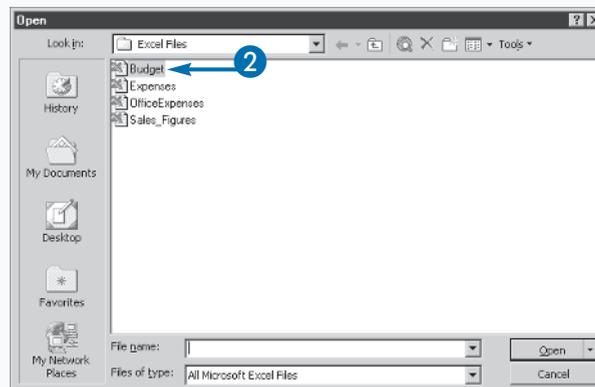
Run a Macro

- 1 Click File→Open.



The Open dialog box displays.

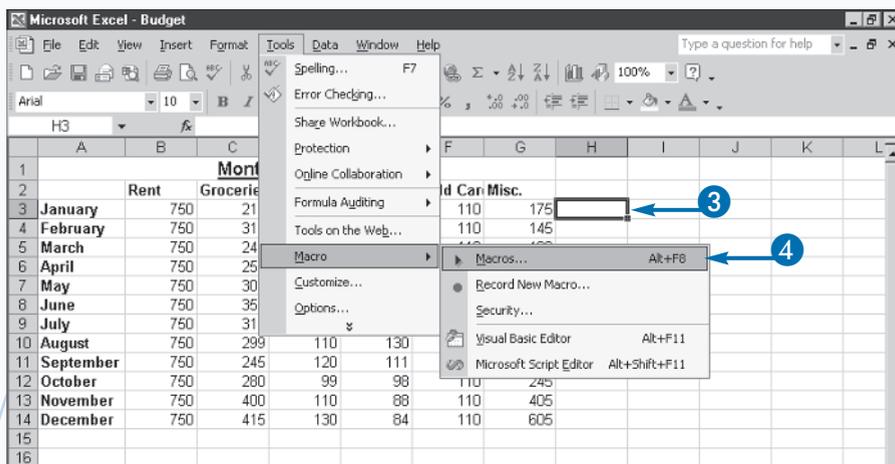
- 2 Click the workbook containing the macro you want to run.



The selected workbook opens.

3 Click the cell where you want the macro to execute.

4 Click Tools→Macro→Macros.

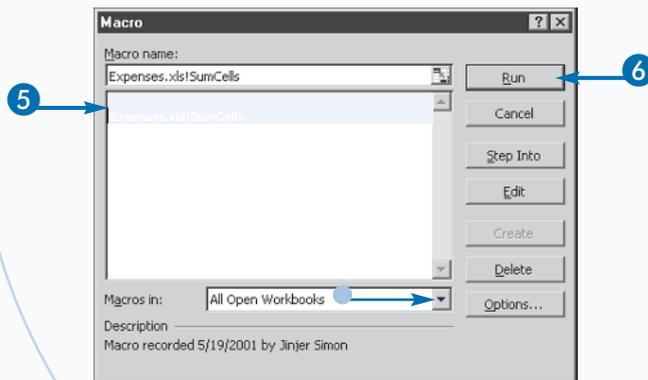


The Macro dialog box displays a list of available macros.

- If the macro is not listed, you can click here and click the location of the macro.

5 Click the macro you want to run.

6 Click Run.



The selected macro executes and makes the appropriate changes to the worksheet.

To run the macro again, repeat steps 3 to 6.

Extra

You can use the Macros In field to limit the number of macros that display on the Macro dialog box. To see the macros in any open workbook, including the Personal Macro Workbook, you can click the  and click the All Open Workbooks option. If you only want to see macros from a specific workbook, select the name of the desired workbook in the Macros In drop-down list. For the global macros stored in the Personal Macro Workbook, you need to select the PERSONAL.XLS option.

Excel differentiates between macros listed in the Macro dialog box by placing the name of the workbook that contains the macro in front of the macro name. For example, Excel lists a macro named Sum_Expenses in the Personal Macro Workbook as PERSONAL.XLS!Sum_Expenses. Because of this nomenclature, two workbooks can have macros with the same name. In other words, if the macro Sum_Cells exists in both the Budget.xls and Expenses.xls workbooks, Excel treats them as two different macros because they are stored in two different locations. The Macro dialog box lists the macros as Budget.xls!Sum_Cells and Expenses.xls!Sum_Cells.

Create and Launch a Keyboard Shortcut

Excel provides the option of keyboard shortcuts to allow you to quickly launch a command, or even a macro, from the keyboard by pressing a combination of keys. With the use of a keyboard shortcut, you can activate a macro by pressing both the Ctrl key and the macro's shortcut key. You assign keyboard shortcuts to a macro during the macro creation, or at any time after you create the macro. See the section "Record a Macro" for more information on creating a macro.

Keyboard shortcuts in Excel are case sensitive. Excel interprets a lowercase s and an uppercase S as two different keys. By using uppercase and lowercase letters, Excel provides more shortcut keys that you can assign to a macro. To execute a macro that has an uppercase letter for the key, such as M, you press Ctrl+Shift+M.

The downside to assigning shortcut keys to a macro is that you have to remember the assigned shortcut. If you forget your shortcut assignment for a selected macro, you can view it in the Macro Options dialog box, which you access via the Macro dialog box.

Excel allows you to assign any key as the shortcut for your macro. If Excel uses the same key as a shortcut key for a standard Excel option, your shortcut definition overrides the Excel definition. For example, when you press Ctrl+O, Excel opens the Open dialog box, enabling you to select a workbook to open. If you create a shortcut key macro of o, your macro executes whenever you press Ctrl+O instead of displaying the Open dialog box. With this in mind, avoid using shortcut keys that you use for other common Excel tasks.

Create a Keyboard Shortcut

- 1 On the Macro dialog box, click the desired macro.

You can click Tools→Macro→Macros to display the Macro dialog box.

Note: See the section "Run a Macro" for more on the Macro dialog box.

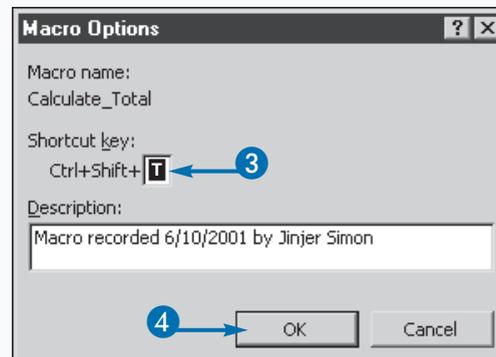
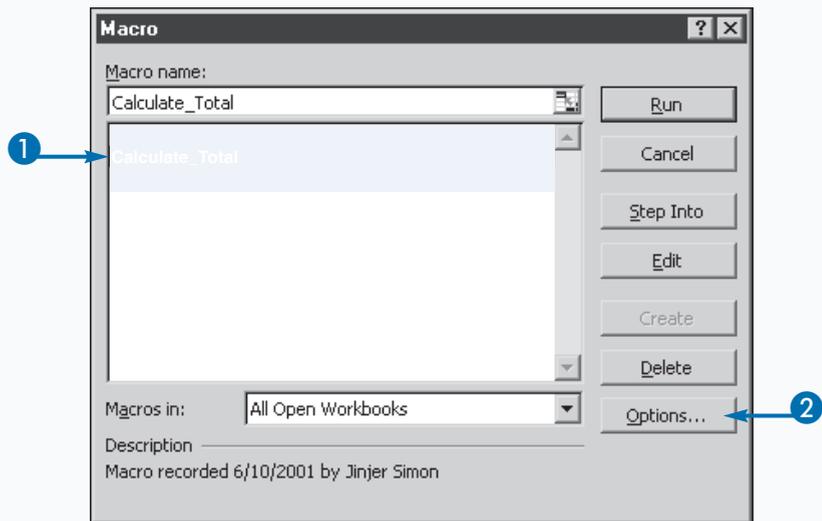
- 2 Click Options.

The Macro Options dialog box displays for the selected macro.

- 3 Type the desired shortcut key in the Shortcut Key box.

When you type an uppercase character in the field, a message appears, reminding you to type **Ctrl+Shift+shortcut key**.

- 4 Click OK to save the shortcut key.



Launch a Keyboard Shortcut

- 1 Click the cell where you want the macro to execute.
- 2 Press Ctrl and the shortcut key to activate the macro.

- The selected macro executes and makes the appropriate changes to the worksheet.

- 3 To run the macro again, repeat steps 1 and 2.

Microsoft Excel - OfficeExpenses

	A	B	C	D	E	F	G	H	I	J	K	
1		Office Expenses										
2		Rent	Advertising	Phone	ISP	Parking	Misc.					
3	January	350	400	200	110	50	175					
4	February	350	350	240	110	50	145					
5	March	350	550	310	110	50	123					
6	April	350	795	250	110	50	211					
7	May	350	800	245	110	50	345					
8	June	350	795	310	110	50	149					
9	July	350	600	340	110	50	190					
10	August	350	910	215	110	50	256					
11	September	350	845	400	110	50	298					
12	October	350	765	450	110	50	245					
13	November	350	900	465	110	50	405					
14	December	350	1005	500	110	50	605					
15												
16												

Microsoft Excel - OfficeExpenses

	A	B	C	D	E	F	G	H	I	J	K
1		Office Expenses									
2											
3	January										
4	February	350	350	240	110	50	145				
5	March	350	550	310	110	50	123				
6	April	350	795	250	110	50	211				
7	May	350	800	245	110	50	345				
8	June	350	795	310	110	50	149				
9	July	350	600	340	110	50	190				
10	August	350	910	215	110	50	256				
11	September	350	845	400	110	50	298				
12	October	350	765	450	110	50	245				
13	November	350	900	465	110	50	405				
14	December	350	1005	500	110	50	605				
15											
16											

Extra

When you use shortcut keys for macros in other workbooks, you may not always receive the correct macro to execute. Excel does not let you assign a shortcut key if a macro in your current workbook uses the shortcut, but it does not check unopened workbooks. If you open workbooks with the same shortcut key, Excel does not know which macro to execute when you select the shortcut. If you use the shortcut keys for a macro and do not receive the anticipated results, you need to verify what you assigned to the shortcut keys.

If you find that the shortcut keys are the same as another available macro, you can reassign a shortcut key to a macro from the Macro dialog box by clicking the macro and then clicking Options to display the Macro Options dialog box. Type the desired shortcut key and click OK. To make the shortcut key modification permanent, save the workbook that contains the macro. Keep in mind, however, that you may find it easier to simply change a lowercase shortcut to uppercase, or vice-versa.

Delete a Macro from a Workbook

You can remove macros from any workbook to reduce your list of available macros. Similar to cleaning a closet, you want to eliminate the stuff you no longer need.

When you delete a macro, Excel removes the actual macro without affecting any changes previously applied to the workbook with that macro. For example, if you use the macro to sum a series of cells, the sum remains the same when you delete the macro. Excel immediately applies the macro changes to the worksheet and then no longer relies on it to maintain any future changes you make.

To delete a macro, you must open the workbook containing the macro because you can only see macros within open workbooks in Excel.

You use the steps in this section to store your macro in unhidden workbooks. If you can open the workbook with the Open command on the File menu, Excel does not hide

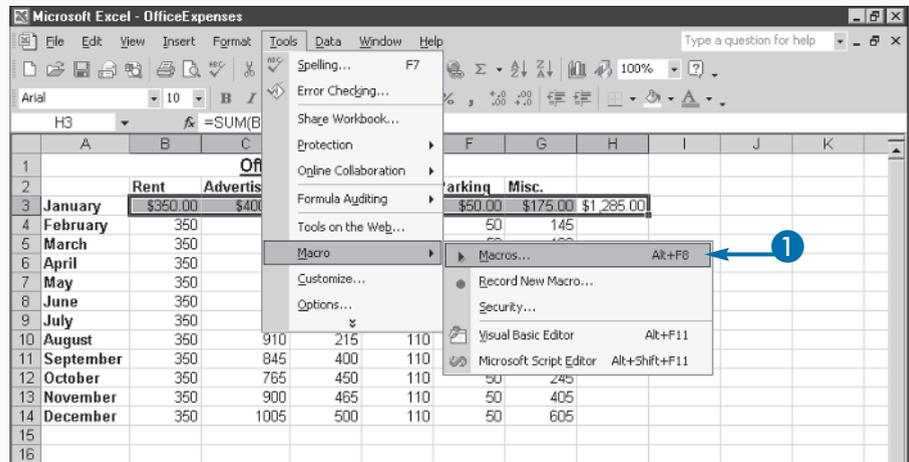
the workbook. You cannot see hidden workbooks when you view the available workbooks in a folder. A good example of a hidden workbook is the Personal Macro Workbook, which loads automatically when you run Excel.

To delete a macro that you store in the Personal Macro Workbook, you need to perform different steps. See the section "Delete from the Personal Macro Workbook" for more information on working with the Personal Macro Workbook or any other hidden workbook.

Remember that you cannot undo the deletion process. If you delete the wrong macro, you can only restore it by recording it again. If you do not want to re-create an unintentionally deleted macro, try closing the workbook without saving it and then reopen the workbook. This eliminates any changes made since your last save, and restores any deleted macros. Of course, you lose any other changes you made to the workbook if you close without saving.

Delete a Macro from a Workbook

- 1 Click Tools→Macro→Macros.

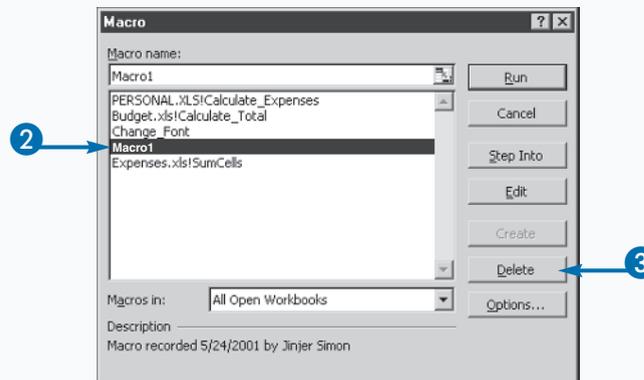


The Macro dialog box displays a list of available macros.

- 2 Click the macro you want to delete.

If the Macro dialog box does not list the macro, click the location of the macro.

- 3 Click Delete.

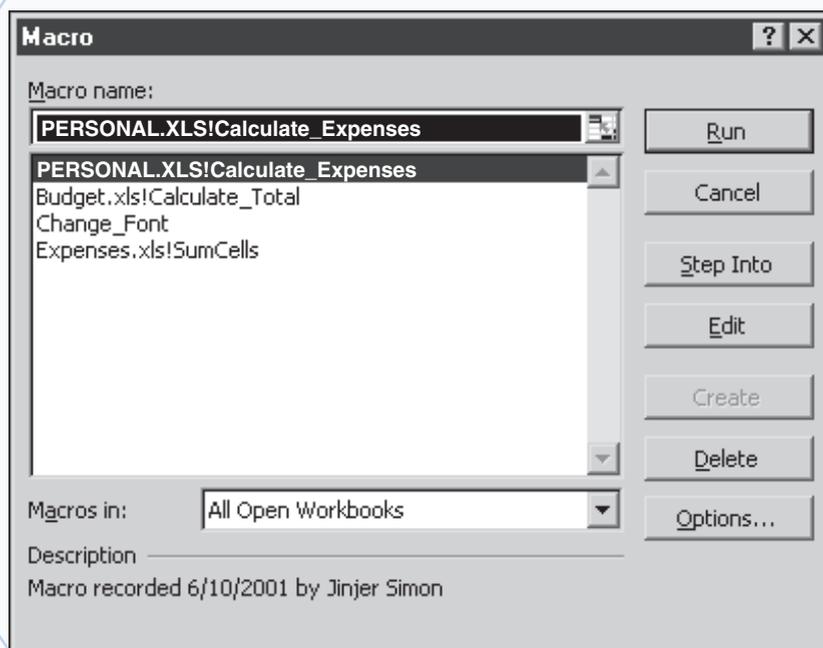
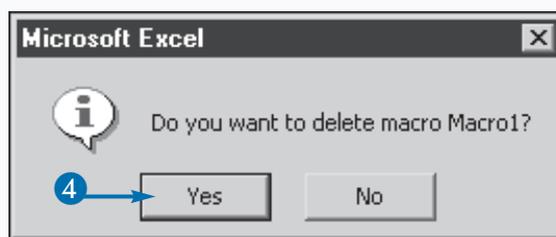


A message box appears asking if you want to delete the macro.

- 4 Click Yes to delete the macro.

If the macro listed is not the one you intended to delete, click No.

Excel deletes the macro from the workbook.



Extra

When you delete a macro, Excel only deletes the macro. If you add the macro to a menu or toolbar button, it retains the macro reference. If you select one of these options after deleting a macro, an error message displays indicating that Excel cannot find the macro. See the sections "Assign a Macro to a Toolbar Button" and "Assign a Macro to a Menu" for more information about assigning macros to toolbars and menus.

To remove menu options and toolbar buttons, click **Tools**→**Customize** to display the Customize dialog box. You can only modify menus and toolbars within Excel when the Customize dialog box displays on the screen. While the Customize dialog box displays, you can right-click the desired icon or menu option and select the **Delete** option to remove it, or you can click the button or menu option and drag it onto the Customize dialog box. Keep in mind that dragging a toolbar or menu option onto the Customize dialog box does not add it to the Customize dialog box.

Delete from the Personal Macro Workbook

You can delete macros that you no longer use from the Personal Macro Workbook. The Personal Macro Workbook stores macros that you want to make available to all workbooks. Excel creates the Personal Macro Workbook when you store your first macro in it. After Excel creates the Personal Macro Workbook, the workbook opens as a hidden file whenever you run Excel. You can only tell that a hidden file exists by viewing the Macro dialog box, where Excel lists the macros used by the Personal Macro Workbook.

Excel stores the Personal Macro Workbook as a file named Personal.xls with a typical path of C:\documents and settings\user_name\Application Data\Microsoft\Excel\XLSTART folder.

If you try to delete a macro out of the Personal Macro Workbook from the Macro dialog box, Excel displays a message box with the message, "Cannot edit a macro in a hidden workbook. Unhide the workbook using the Unhide

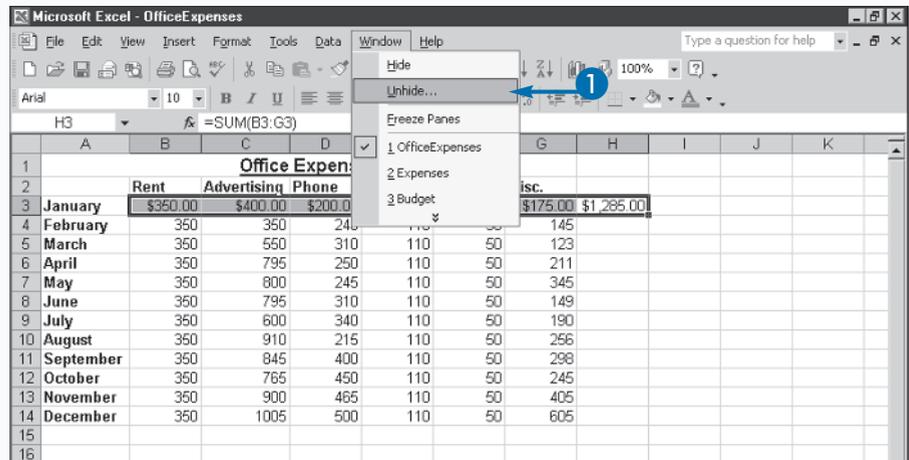
command." By default, Excel does not allow you to delete macros out of hidden workbooks. Because Excel hides the Personal Macro Workbook, you cannot delete the macros in it without first unhiding the workbook using the Excel Unhide command.

After you delete the macro from the workbook, make sure that you hide the workbook again. If you do not hide the Personal Macro Workbook again, the workbook appears as an open workbook. Because you only use this workbook for storing globally used macros, you do not want to make other types of modifications to it. By hiding it, you keep it out of the way and eliminate the possibility of having unwanted changes made to it.

You can also use the Visual Basic Editor to remove macros from the Personal Macro Workbook. See Chapter 2 for more information about removing macros using the Visual Basic Editor.

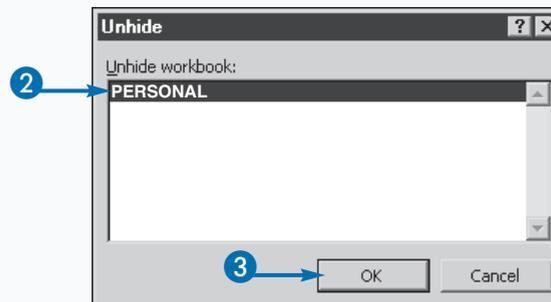
Delete from the Personal Macro Workbook

- 1 Click Window→Unhide.



The Unhide dialog box displays a list of open workbooks that are currently hidden.

- 2 Click PERSONAL.
- 3 Click OK.



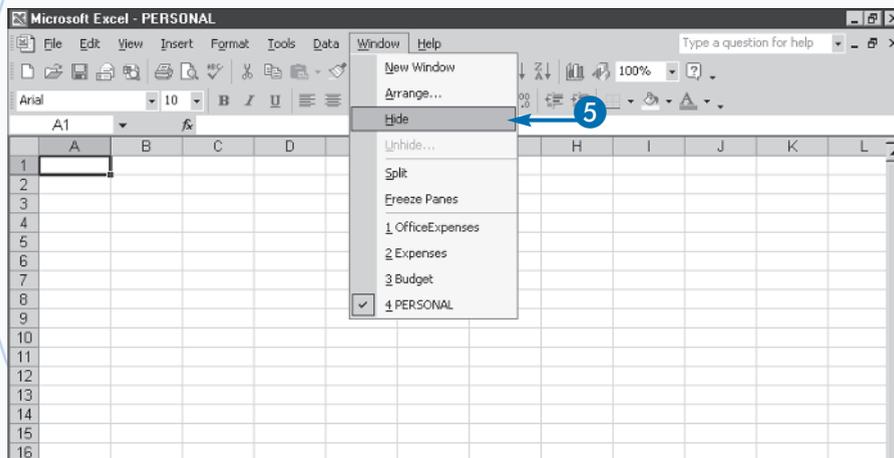
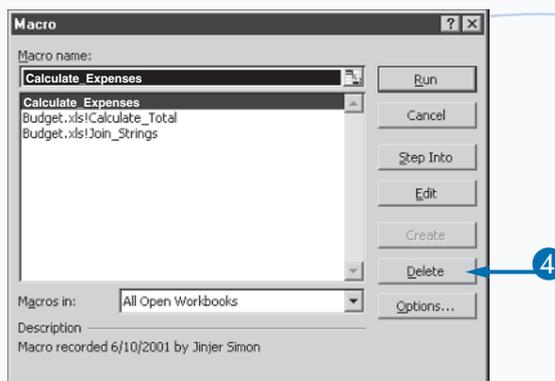
The Personal Macro Workbook is unhidden and displays in the Excel Window.

- 4 Click Delete to remove the macro from the workbook.

Note: For information about deleting a macro, see the section "Delete a Macro from a Workbook."

- 5 After modifications are complete, click Window→Hide.

The workbook is hidden.



Extra

Typically, you do not share the Personal Macro Workbook with other users. Excel creates a different Personal Macro Workbook for each username on a machine. If you have multiple users on your computer with different usernames, Excel creates a different Personal Macro Workbook for each user. You can share a Personal Macro Workbook between different users, even on the same computer, by copying the workbook. You can use Windows Explorer and copy the workbook from one user to another if you want to make the macros in that workbook available to other users. To do so, make sure you copy the PERSONAL.XLS file to the C:\documents and settings\user_name\Application Data\Microsoft\Excel\XLSTART folder from the user workbook you want to share to each user's corresponding folder. Keep in mind, you can only have one PERSONAL.XLS file for each user. If a user already has a Personal Macro Workbook, you can overwrite it with the new one. Of course, if you overwrite an existing Personal Macro Workbook, Excel no longer makes any macros you store in the workbook available. To eliminate potential problems, you should rename the existing workbook so that a user can still access it if necessary.

Assign a Macro to a Toolbar Button

You can assign any macro to an Excel toolbar. Excel uses toolbars to provide quick access to commonly used commands. You can make macros more accessible by creating a button on a toolbar to execute a macro. By doing this, you also do not have to remember the shortcut key that launches the macro. If you create a macro toolbar button, each time you want to run the macro, you simply click the appropriate button.

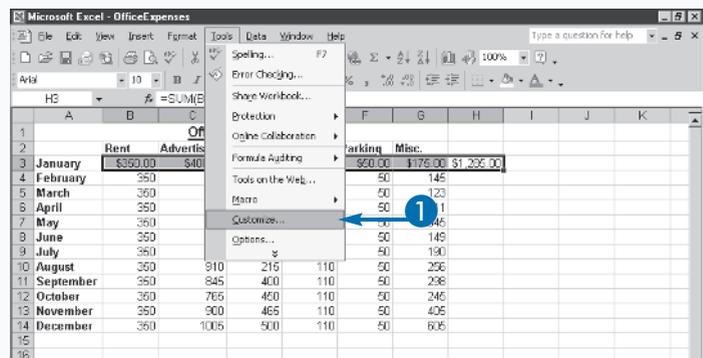
When you add a button to a toolbar, it remains on that toolbar for all workbooks that you open in Excel. In other words, even if the active workbook does not have access to the macro because you closed the corresponding workbook, the Toolbar button still displays. For that reason, you should assign a macro that exists in your Personal Macro Workbook to a toolbar button to make the macro available

from all workbooks. Remember that the Personal Macro Workbook stores commonly used macros, and opens as a hidden file each time you run Excel. Excel always keeps the Personal Macro Workbook open and, therefore, makes any macros you have in the Workbook always available for use by other workbooks that you open. For more on macro storage, see the section "An Introduction to Macros."

You can add buttons to any of the existing Excel toolbars, or you can create new toolbars for your buttons. By creating a separate toolbar for your macros, you keep your custom macros together in one location and you avoid ruining existing toolbars. You can select the toolbars to display in Excel on the Customize dialog box. Of course, you need to display a toolbar in Excel before you can add buttons to it. You also display the Customize dialog box to add buttons to toolbars.

Assign a Macro to a Toolbar Button

1 Click Tools→Customize.



The Customize dialog box displays options for updating menus and toolbars.

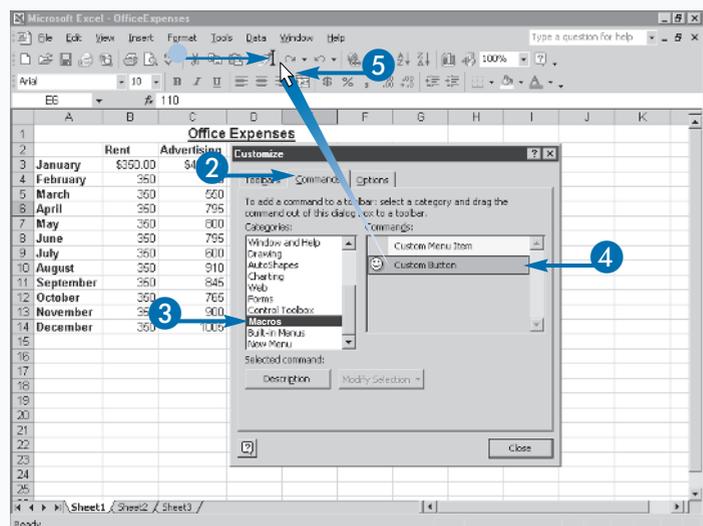
2 Click the Commands tab.

3 Click Macros.

4 Click the Custom Button option.

5 Drag the option to the desired location on the toolbar.

- As you drag the button across the toolbar, Excel inserts a line to indicate the location.



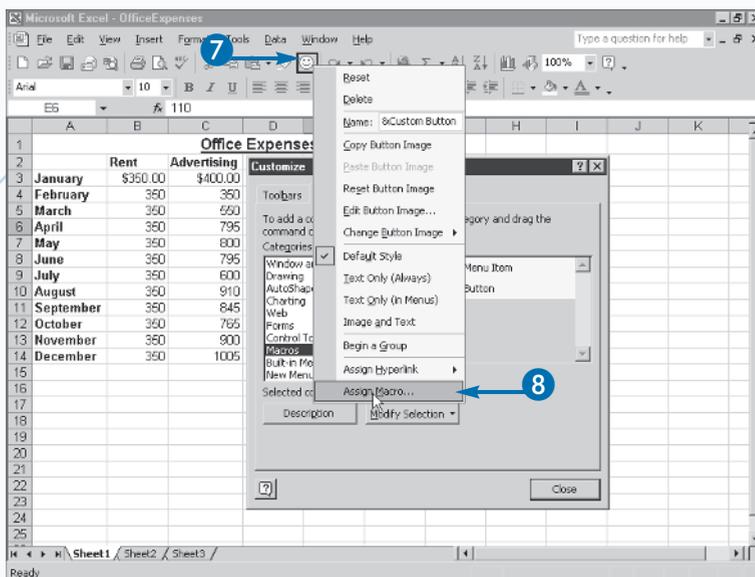
- 6 Release the mouse button.

The button appears on the toolbar.

- 7 Right-click the toolbar button to display a menu of options.

Note: You must have the Customize dialog box open to customize the button on the toolbar.

- 8 Click Assign Macro.

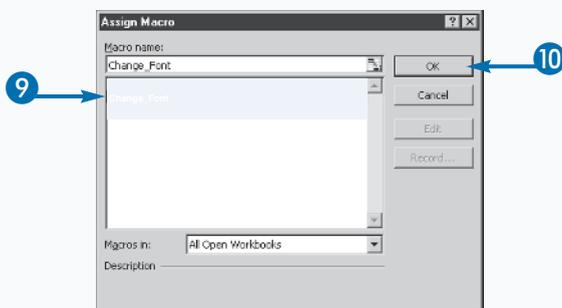


The Assign Macro dialog box displays a list of currently available macros.

- 9 Click the name of the macro you want to assign to the new button.

- 10 Click OK.

The macro runs each time you select the button.



Apply It

Chances are you do not want to keep the default smiley face button image Excel inserts on the toolbar. You can change the image by using one of two different options on the menu that appears when you right-click the mouse over the toolbar button when you have the Customize dialog box open. If you want to select an image from a list of existing images, you can click the Change Button Image option. If you do not like the images on the Change Button Image menu, you can also create your own button image on the Button Editor dialog box. To change the button image, click the Edit Button Image option on the menu.

Changing the button image is similar to a paint-by-number exercise you did as a kid. You have 16 different colors you can use to create the new image. Click the desired color and then click the pixel of the image that you want to modify. You can also move the image within the window by clicking and dragging it. But the maximum size of the button image is 16x16 pixels or the contents of the window. When you close the dialog box, the button image updates.

Assign a Macro to a Menu

You can assign a macro to any existing Excel menu. If you do not want to use existing menus, you can even create a new menu. By assigning a macro to a menu, you make the macro as accessible as any menu option. Assigning macros to menus eliminates the need to remember the shortcut key required to launch the macro.

When you add a macro to a menu, it remains on the menu for all workbooks that you open in Excel. For that reason, you should assign a macro that exists in your Personal Macro Workbook to a menu to ensure that all workbooks can access the macros. The Personal Macro Workbook stores commonly used macros for the current user, and opens as a hidden file each time you run Excel. Because the Personal Macro Workbook is always open, any workbook

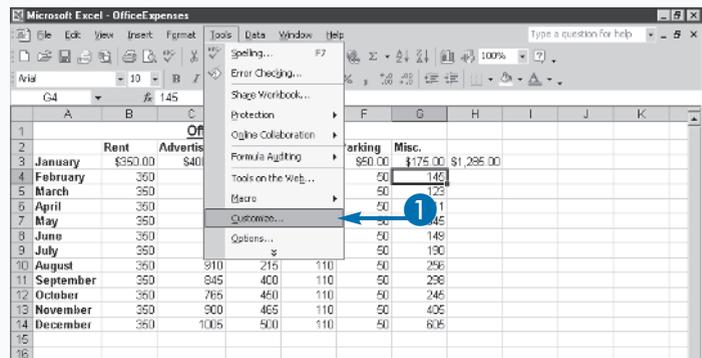
can use all of the macros it contains. To learn more about the Personal Macro Workbook, see the section "An Introduction to Macros."

You can assign the macro to any available menu; however, to keep your macros easy to find, you may want to place them all on one custom menu. You can create a new Excel menu using the Customize dialog box. Of course, whatever menu you decide to use as a home for your macro must exist on the Excel window before you can add the macro option to it.

You add options to a menu by dragging them onto the menu from the Customize dialog box. In fact, you can modify menus only while the Customize dialog box displays. You can remove menu options in a similar fashion by dragging them from the menu back to the Customize dialog box.

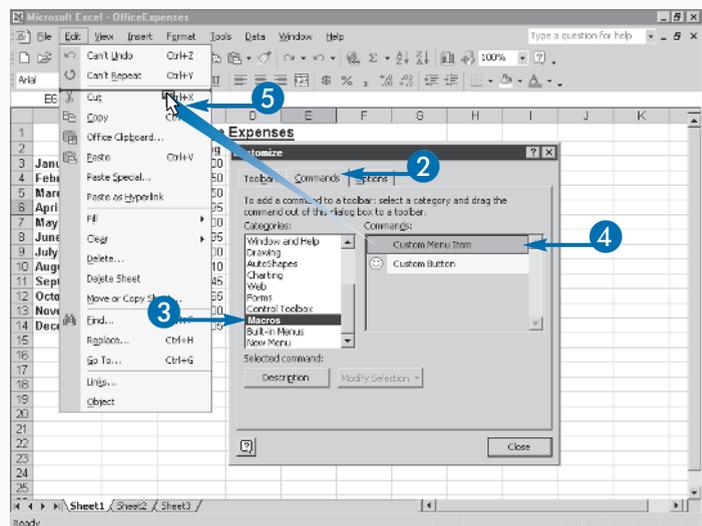
Assign a Macro to a Menu

- 1 Click Tools→Customize.



The Customize dialog box appears.

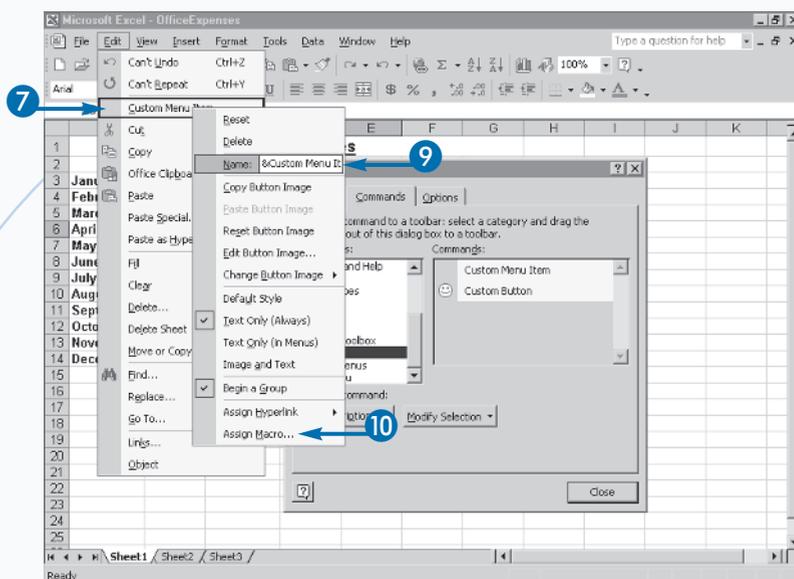
- 2 Click the Commands tab.
- 3 Click Macros.
- 4 Click the Custom Menu Item option.
- 5 Drag the item to the desired menu.



The menu expands and a line indicates your position in the menu.

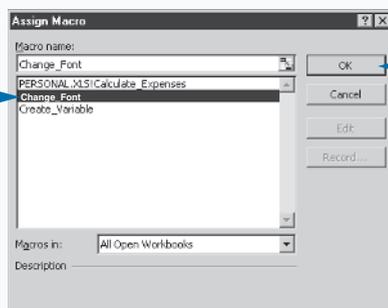
- 6 Release the mouse button.

- 7 Right-click the menu option.
- 8 Click Name.
- 9 Type the desired name for the macro menu option in the field.
- 10 Click Assign Macro.



The Assign Macro dialog box appears.

- 11 Click the name of the macro you want to assign to the new menu option.
- 12 Click OK.



The macro runs each time you select the menu option.

Extra

You can create a custom menu for macros you place on a menu. This keeps all the macro references in one location and prevents clutter on the existing Excel menus. To create a new menu, open the Customize dialog box as described in the steps on this page. On the Customize tab select the New Menu option as the desired category. A New Menu option displays as the available command. Click the New Menu option and drag it to the desired menu location. After you add the menu, you can right-click it, and change the name, which makes it ready to receive your macros.

When you name a menu option, you can also create a shortcut key that corresponds to the menu option. Similar to the shortcuts you create for macros, the menu option shortcut launches whatever command you assign to the menu option. Also, these shortcuts launch with the Alt key. To create a menu shortcut, you need to type a & before the character in the menu item name that corresponds to the shortcut key. For example, if you want Alt+T to launch the menu option "Determine Total," you place the & before the letter T: "Determine &Totals."

Set Macro Security

Due to the increasing problem with computer viruses, specifically macro viruses, by default, Excel disables all macros in worksheets that you open, except those with a signature from a trusted source. You can Excel open all macros regardless of source, or prompt you before opening unsigned macros, by modifying the macro security level.

Digital signatures, which a creator uses to verify a macro's safety, remain attached to a macro or other file so long as no one modifies the macro or file. Macro modifications require you, as the creator, to reattach the signature. A macro with a valid digital signature confirms the macro's origins and that no one altered it.

Depending how you use Excel — and whether you open workbooks from other sources — you may want to modify the security type that Excel uses to open workbooks containing macros. The three security settings include:

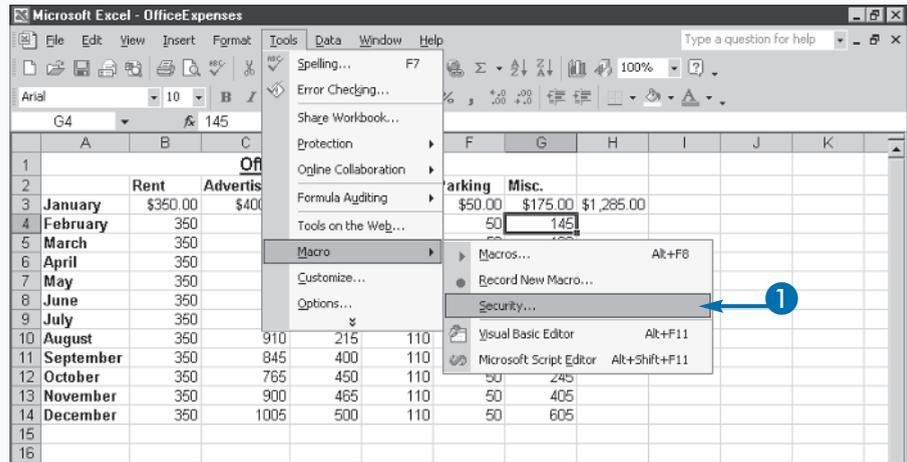
- **High:** The default, selecting this level disables all unsigned macros, even ones you create. You have the option of selecting macros from other trusted sources when you run Excel.
- **Medium:** With this level you can specify whether you want to run macros from trusted and unsigned sources when you load Excel. Select this level if you want to eliminate the hassle of signing the macros you create.
- **Low:** Excel automatically loads all workbooks and macros without checking to see if they are from trusted sources. With this setting, the only protection from macro viruses is a good virus scanner.

To eliminate the hassle and expense of acquiring a digital certificate, you can personally sign your macros by running SelfCert.exe, an Office XP program. Creating and attaching your personal signature indicates that you certify the security of a macro, identifies macros you create, and distinguishes your macros from other macros.

Set Macro Security

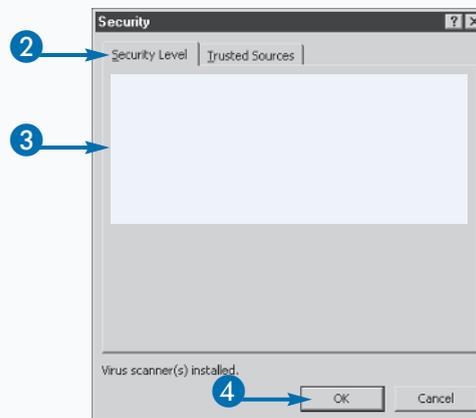
SET SECURITY

- 1 Click Tools→Macro→Security.



The Security dialog box displays.

- 2 Click the Security Level tab.
- 3 Click the desired security level (changes to).
- 4 Click OK.



Excel assigns a security level.

CREATE A PERSONAL SECURITY CERTIFICATE

- 1 Open Microsoft Windows Explorer.
- 2 Click the Office10 subfolder of the Microsoft Office folder.

Note: If you performed a typical Office XP installation, Excel locates your program files in C:\Program Files\Microsoft Office.

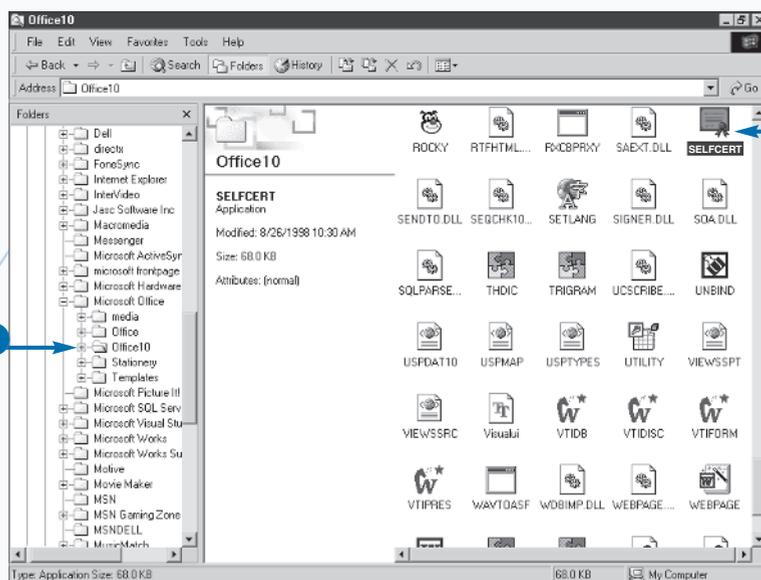
- 3 Double-click the SelfCert.exe program file.

The Create Digital Certificate dialog box displays.

- 4 Type your name.
- 5 Click OK.

Excel creates a digital certificate.

Note: See Chapter 2 to assign the certificate to your macros.



Extra

Assigning a certificate you create with SelfCert.exe to a project indicates the project is self-signed and not authenticated. This option works well for personal workbooks. However, if you plan to distribute your workbook to other users, you probably want to consider acquiring a true digital signature file. When you use a commercial digital signature file, the digital ID attaches to the macro. The Digital ID remains with the macro, and — if someone alters the macro in any way — notifies the user when the user should not trust the macro. This ensures that a macro you create is not altered to harm another person's machine.

The most common location for obtaining a digital certification is from VeriSign, Inc. Of course, to obtain a commercial certification, you have to submit an application and pay the appropriate fee. You can find out more about obtaining a digital certification for your macro at www.verisign.com. Another company that you can contact for a digital ID is Thwate Consulting. You can find out about their digital signature options at www.thwate.com.