12 Automating Repetitive Tasks by Using Macros

In this chapter, you will learn how to

- ✔ Enable and examine macros.
- ✔ Create and modify macros.
- ✔ Run macros when a button is clicked.
- ✔ Run macros when a workbook is opened.

Many tasks you perform in Microsoft Excel 2010 are done once (for example, entering sales data for a particular day or adding formulas to a worksheet) or can be repeated quickly by using tools in Excel (for example, changing the format of a cell range). However, there are probably one or two tasks you perform frequently that require a lot of steps to accomplish. For example, you might have several cells in a worksheet that contain important data you use quite often in presentations to your colleagues. Instead of going through a lengthy series of steps to highlight the cells with the important information, you can create a macro, which is a recorded series of actions, to perform the steps for you. After you have created a macro, you can run, edit, or delete it as needed.

In Excel, you run and edit macros by using the items available in the Macros group on the View tab. You can make your macros easier to access by creating new buttons on the Quick Access Toolbar, to which you can assign your macros. If you run a macro to highlight specific cells in a worksheet every time you show that worksheet to a colleague, you can save time by adding a Quick Access Toolbar button that runs the macro to highlight the cells for you.

Another handy feature of Excel macros is that you can create macros that run when a workbook is opened. For example, you might want to ensure that no cells in a worksheet are highlighted when the worksheet opens. You can create a macro that removes any special formatting from your worksheet cells when its workbook opens, enabling you to emphasize the data you want as you present the information to your colleagues.

In this chapter, you'll learn how to open, run, create, and modify macros; create Quick Access Toolbar buttons and shapes that enable you to run macros with a single mouse click; define macro security settings; and run a macro when a workbook is opened.

Practice Files Before you can complete the exercises in this chapter, you need to copy the book's practice files to your computer. The practice files you'll use to complete the exercises in this chapter are in the Chapter12 practice file folder. A complete list of practice files is provided in "Using the Practice Files" at the beginning of this book.

Enabling and Examining Macros

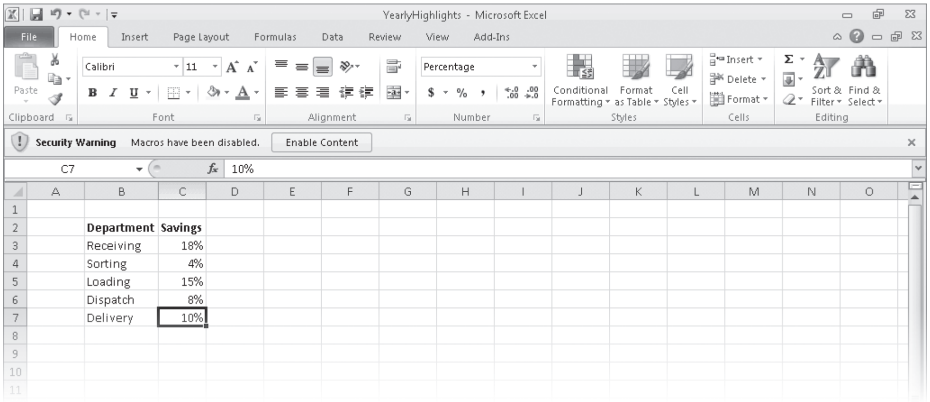
It's possible for unscrupulous programmers to write viruses and other harmful programs by using the Microsoft Visual Basic for Applications (VBA) programming language, so you need to be sure that you don't run macros from unknown sources. In addition to running protective software such as Windows Defender, you can also change your Excel macro security settings to control when macros can be run. After you're sure a macro is safe, you open it in the Visual Basic Editor to examine its code.

Macro Security in Excel 2010

In versions of Excel prior to Excel 2007, you could define macro security levels to determine which macros, if any, your workbooks would be allowed to run, but there was no workbook type in which all macros were disallowed. Excel 2010 has several file types you can use to control whether a workbook will allow macros to be run. The following table summarizes the macro-related file types.

Extension	Description
.xlsx	Regular Excel 2010 workbook; macros are *disabled*
.xlsm	Regular Excel 2010 workbook; macros are *enabled*
.xltx	Excel 2010 template workbook; macros are *disabled*
.xltm	Excel 2010 template workbook; macros are *enabled*

When you open a macro-enabled workbook, the Excel program-level security settings might prevent the workbook from running the macro code. When that happens, Excel displays a security warning on the Message Bar.



Security Warning Macros have been disabled.

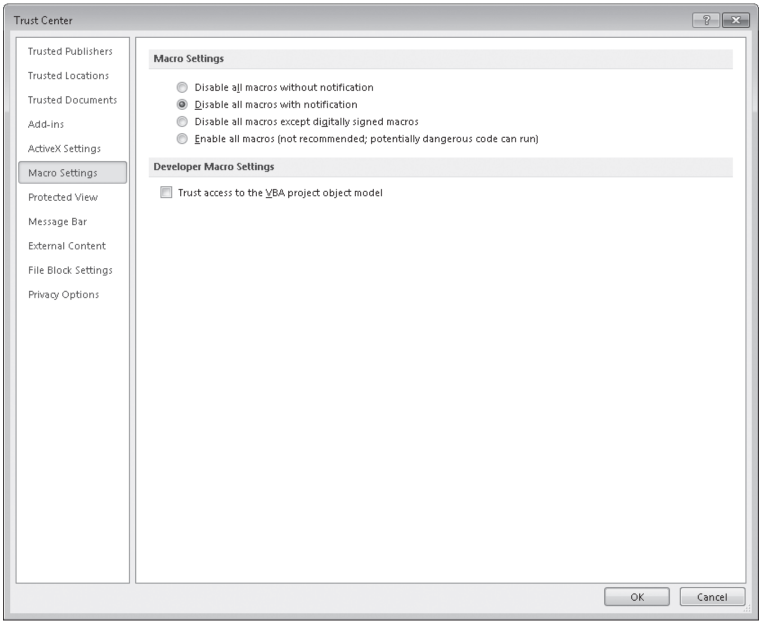
C7 10%

Department	Savings
Receiving	18%
Sorting	4%
Loading	15%
Dispatch	8%
Delivery	10%

Troubleshooting **The appearance of buttons and groups on the ribbon changes depending on the width of the program window. For information about changing the appearance of the ribbon to match our screen images, see "Modifying the Display of the Ribbon" at the beginning of this book.**

Clicking the Enable Content button lets the workbook use its macros. Always take the time to verify the workbook's source and consider whether you expected the workbook to contain macros before you enable the content. If you decide not to enable the macros in a workbook, click the Close button at the right edge of the Message Bar.

You can change your program-level security settings to make them more or less restrictive; to do so, click the File tab, click Options, and then, in the Excel Options dialog box, click the Trust Center category. On the page that appears, click the Trust Center Settings button to display the Trust Center dialog box.



Macro Settings

○ Disable all macros without notification
● Disable all macros with notification
○ Disable all macros except digitally signed macros
○ Enable all macros (not recommended; potentially dangerous code can run)

Developer Macro Settings

☐ Trust access to the VBA project object model

The Excel default macro security level is Disable All Macros With Notification, which means that Excel displays a warning on the Message Bar but allows you to enable the macros manually. Selecting the Disable All Macros Without Notification option does exactly what the label says. If Consolidated Messenger's company policy is to disallow all macros in all Excel workbooks, its employees would select the Disable All Macros Without Notification option.

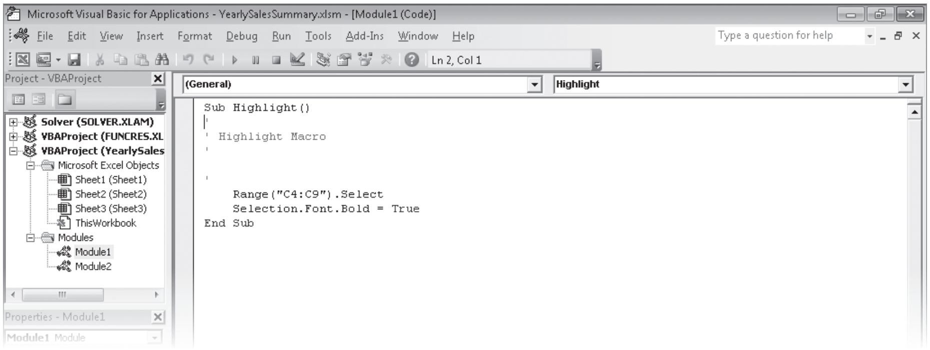
Important Because it is possible to write macros that act as viruses, potentially causing harm to your computer and spreading copies of themselves to other computers, you should never choose the Enable All Macros security setting, even if you have virus-checking software installed on your computer.

Examining Macros

The best way to get an idea of how macros work is to examine an existing macro. To do that, display the View tab. In the Macros group, click the Macros button, and then click View Macros.

Tip In the Macro dialog box, you can display the macros available in other workbooks by clicking the Macros In box and selecting a workbook by name or selecting All Open Workbooks to display every macro in any open workbook. If you select either of those choices, the macro names that are displayed include the name of the workbook in which the macro is stored. Clicking This Workbook displays the macros in the active workbook.

The Macro dialog box displays a list of macros in your workbook. To view the code behind a macro, you click the macro's name and then click Edit to open the Visual Basic Editor.



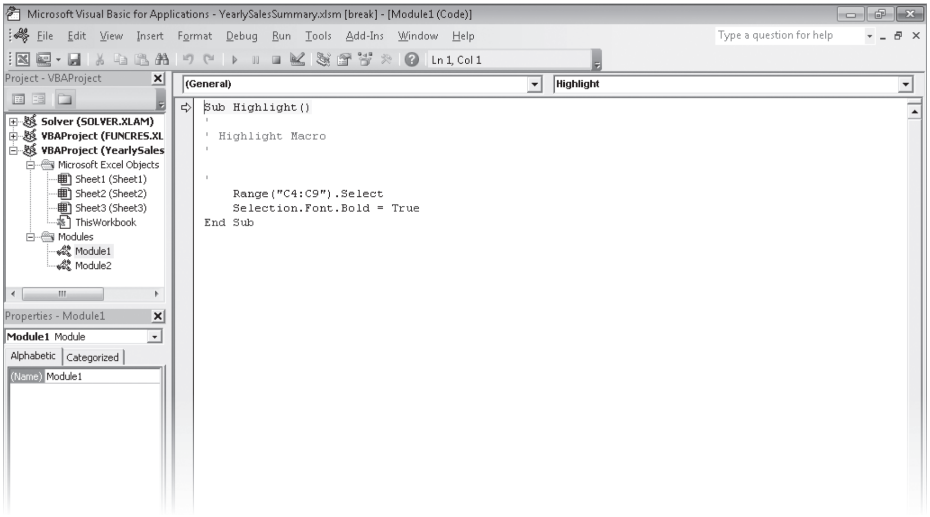
```
Sub Highlight()
'
' Highlight Macro
'

'
    Range("C4:C9").Select
    Selection.Font.Bold = True
End Sub
```

Keyboard Shortcut You can also open and close the Visual Basic Editor by pressing Alt+F11.

See Also To see a complete list of keyboard shortcuts, see "Keyboard Shortcuts" at the end of this book.

Excel macros are recorded by using VBA. Consider, for example, the code for a macro that selects the cell range C4:C9 and changes the cells' formatting to bold. The first line of the macro identifies the cell range to be selected (in this case, cells C4:C9). After the macro selects the cells, the next line of the macro changes the formatting of the selected cells to bold, which has the same result as clicking a cell and then clicking the Bold button in the Font group on the Home tab.

To see how the macro works, you can open the Macro dialog box, click the name of the macro you want to examine, and then click Step Into. The Visual Basic Editor opens, with a highlight around the instruction that will be executed next.



```
Sub Highlight()
'
' Highlight Macro
'

'
    Range("C4:C9").Select
    Selection.Font.Bold = True
End Sub
```

To execute an instruction, press F8. The highlight moves to the next instruction, and your worksheet then changes to reflect the action that resulted from executing the preceding instruction.

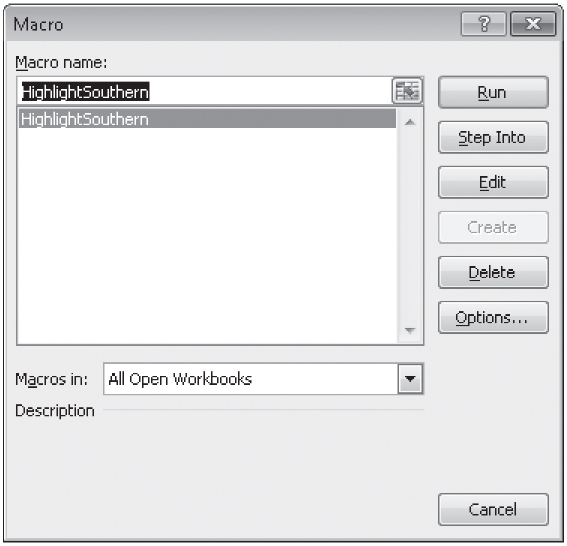
You can run a macro without stopping after each instruction by opening the Macro dialog box, clicking the macro to run, and then clicking Run. You'll usually run the macro this way; after all, the point of using macros is to save time.

In this exercise, you'll examine a macro in the Visual Basic Editor, move through the first part of the macro one step at a time, and then run the entire macro without stopping.

SET UP You need the VolumeHighlights_start workbook located in your Chapter12 practice file folder to complete this exercise. Start Excel, open the VolumeHighlights_start workbook, click the Enable Content button on the Message Bar (if necessary), and save the workbook as *VolumeHighlights*. Then follow the steps.

1. On the **View** tab, in the **Macros** group, click the **Macros** arrow and then, in the list that appears, click **View Macros**.

   The Macro dialog box opens.



Macro name: HighlightSouthern

HighlightSouthern

Macros in: All Open Workbooks

2. Click the **HighlightSouthern** macro, and then, to display the macro code, click **Edit**.

   The Visual Basic Editor opens, with the code for the HighlightSouthern macro displayed in the Module1 (Code) window.

3. In the Visual Basic Editor window, click the **Close** button.

   The Visual Basic Editor closes, and Excel displays the VolumeHighlights workbook.

4. In the **Macros** list, click **View Macros**.

   The Macro dialog box opens.

5. Click the **HighlightSouthern** macro, and then click **Step Into**.

   The macro appears in the Visual Basic Editor, with the first macro instruction highlighted.

6. Press the F8 key.

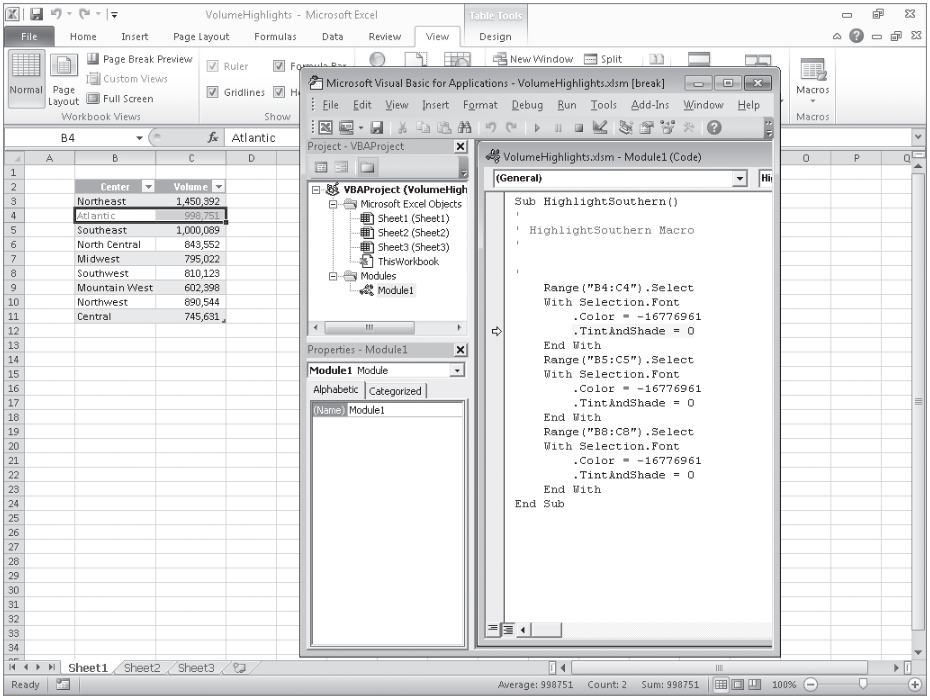
   Excel highlights the next instruction.

7. Press F8 again.

   The macro selects the Atlantic row in the table.

8. Press F8 twice.

   The macro changes the Atlantic row's text color to red.



B4 Atlantic

Center	Volume
Northeast	1,450,392
Atlantic	998,751
Southeast	1,000,089
North Central	843,552
Midwest	795,022
Southwest	810,123
Mountain West	602,398
Northwest	890,544
Central	745,631

```
Sub HighlightSouthern()
'
' HighlightSouthern Macro
'

'
    Range("B4:C4").Select
    With Selection.Font
        .Color = -16776961
        .TintAndShade = 0
    End With
    Range("B5:C5").Select
    With Selection.Font
        .Color = -16776961
        .TintAndShade = 0
    End With
    Range("B8:C8").Select
    With Selection.Font
        .Color = -16776961
        .TintAndShade = 0
    End With
End Sub
```

9. Click the Visual Basic Editor **Close** button.

   A warning dialog box opens, indicating that closing the Visual Basic Editor will stop the debugger.

10. Click **OK**.

    The Visual Basic Editor closes.

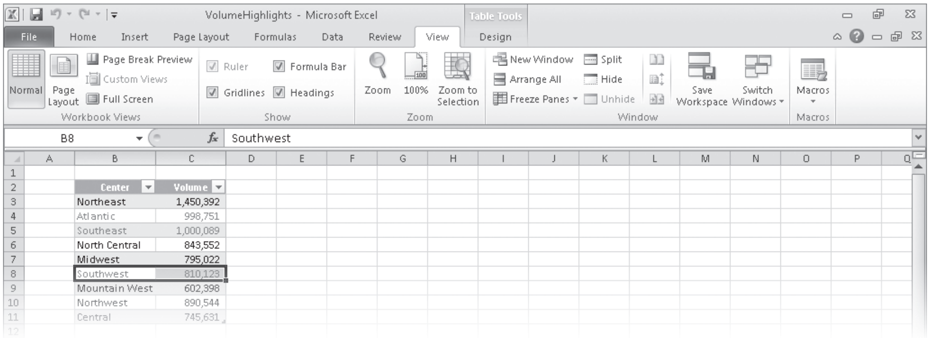
11. In the **Macros** list, click **View Macros**.

    The Macro dialog box opens.

12. Click the **HighlightSouthern** macro.

13. Click **Run**.

    The Macro dialog box closes, and Excel runs the entire macro.



	A	B	C	D
1				
2		Center	Volume	
3		Northeast	1,450,392	
4		Atlantic	998,751	
5		Southeast	1,000,089	
6		North Central	843,552	
7		Midwest	795,022	
8		Southwest	810,123	
9		Mountain West	602,398	
10		Northwest	890,544	
11		Central	745,631	
12				

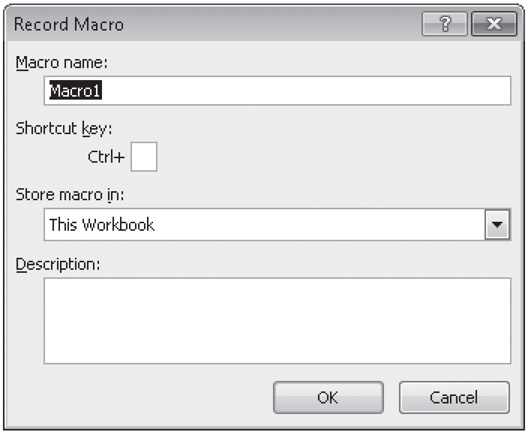
14. On the Quick Access Toolbar, click the **Save** button.

    Excel saves your work.

CLEAN UP Close the VolumeHighlights workbook.

Creating and Modifying Macros

The first step of creating a macro is to plan the process you want to automate. Computers today are quite fast, so adding an extra step that doesn't affect the outcome of a process doesn't slow you down noticeably, but leaving out a step means you will need to re-record your macro. After you plan your process, you can create a macro by clicking the View tab and then, in the Macros group, clicking the Macros arrow. In the list that appears, click Record Macro. When you do, the Record Macro dialog box opens.



Macro name: Macro1

Shortcut key: Ctrl+

Store macro in: This Workbook

Description:

After you type the name of your macro in the Macro Name box, click OK. You can now perform the actions you want Excel to repeat later; when you're done, in the Macros list, click Stop Recording to add your macro to the list of macros available in your workbook.

To modify an existing macro, you can simply delete the macro and re-record it. Or if you just need to make a quick change, you can open it in the Visual Basic Editor and add to or change the macro's instructions. To delete a macro, open the Macro dialog box, click the macro you want to delete, and then click Delete.

See Also For more information about using the Visual Basic Editor, press Alt+F11 to display the Visual Basic Editor, and then press F1 to display the Visual Basic Help dialog box.

In this exercise, you'll record, edit, save, and run a macro that removes the bold formatting from selected cells.

SET UP You need the YearlySalesSummary_start workbook located in your Chapter12 practice file folder to complete this exercise. Open the YearlySalesSummary_start workbook, click the Enable Content button on the Message Bar (if necessary), and save the workbook as *YearlySalesSummary*. Then follow the steps.

1. On the **View** tab, in the **Macros** group, click the **Macros** arrow and then, in the list that appears, click **Record Macro**.

   The Record Macro dialog box opens.

2. In the **Macro name** box, delete the existing name, and then type **RemoveHighlight**.
3. Click **OK**.

   The Record Macro dialog box closes.

4. Select the cell range **C4:C7**.

   The text in these cells is currently bold.

5. On the **Home** tab, in the **Font** group, click the **Bold** button.

6. On the **View** tab, in the **Macros** list, click **Stop Recording**.

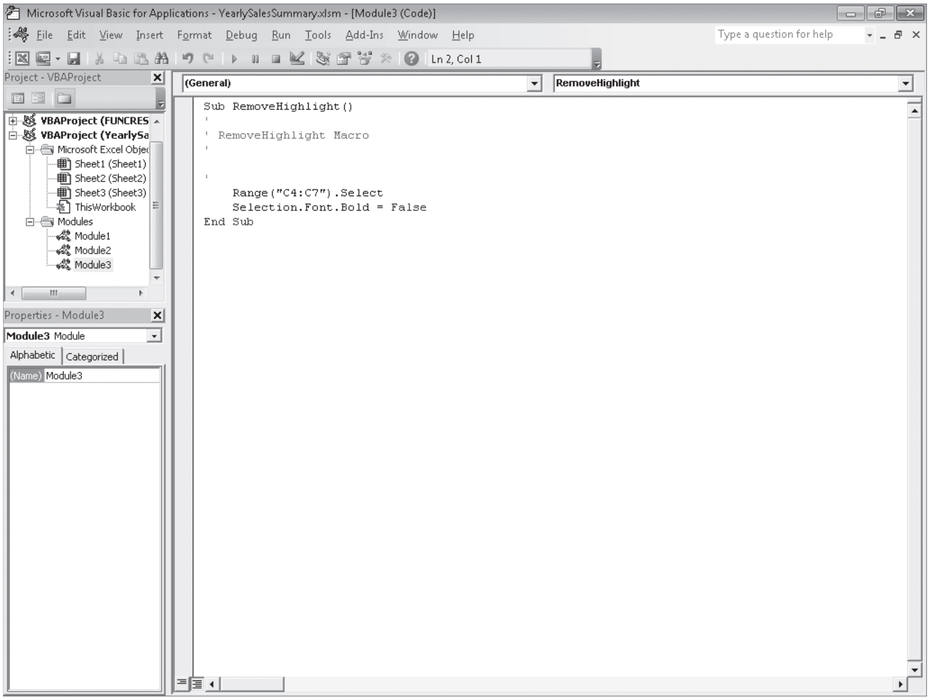
   Excel stops recording the macro.

7. In the **Macros** list, click **View Macros**.

   The Macro dialog box opens.

8. In the **Macro name** area, click **RemoveHighlight**, and then click **Edit**.

   The Visual Basic Editor starts.



```
Sub RemoveHighlight()
'
' RemoveHighlight Macro
'

'
    Range("C4:C7").Select
    Selection.Font.Bold = False
End Sub
```

9. Edit the line of code that currently reads **Range("C4:C7").Select** so that it reads **Range("C3:C9").Select**.

This macro statement selects the cell range C3:C9, not the incorrect range C4:C7.

10. On the **Standard** toolbar of the Visual Basic Editor, click the **Save** button to save your change.

11. On the title bar of the Visual Basic Editor window, click the **Close** button.

    The Visual Basic Editor closes.

12. Select cells **C3:C9**, format them as bold, and then click cell **C9**.

    Excel formats the cells' contents in bold and then removes the highlight from cells C3:C9.

13. In the **Macros** list, click **View Macros**.

    The Macro dialog box opens.

14. Click **RemoveHighlight**, and then click **Run**.

    The bold formatting is removed from cells C3:C9.

15. On the Quick Access Toolbar, click the **Save** button.

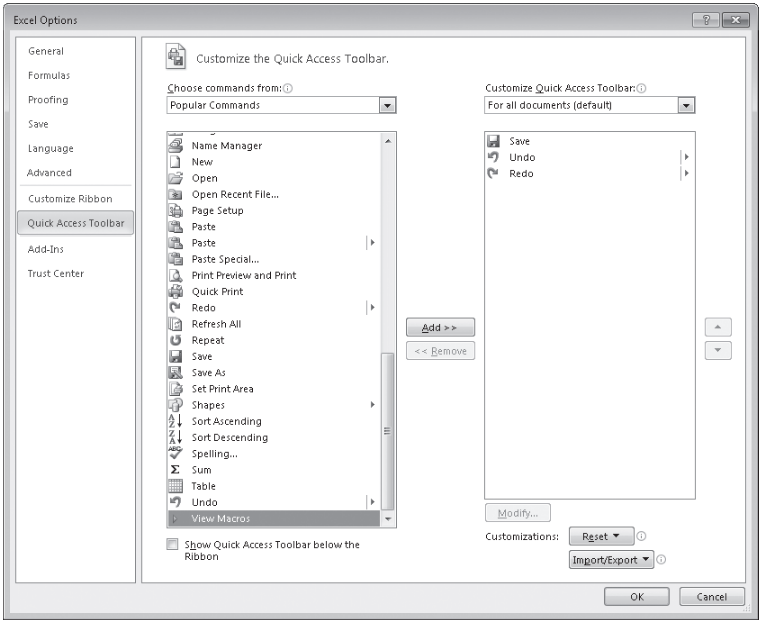
    Excel saves your workbook.

CLEAN UP Close the YearlySalesSummary workbook.

Running Macros When a Button Is Clicked

The ribbon enables you to discover the commands built into Excel quickly. However, it can take a few seconds to display the View tab, open the Macro dialog box, select the macro you want to run, and click the Run button. When you're in the middle of a presentation, taking even those few seconds can reduce your momentum and force you to regain your audience's attention. Excel offers several ways for you to make your macros more accessible.

If you want to display the Macro dialog box quickly, you can add the View Macros button to the Quick Access Toolbar. To do so, click the Customize Quick Access Toolbar button at the right edge of the Quick Access Toolbar, and then click More Commands to display the Customize The Quick Access Toolbar page of the Excel Options dialog box.



Choose commands from: Popular Commands

Customize Quick Access Toolbar: For all documents (default)

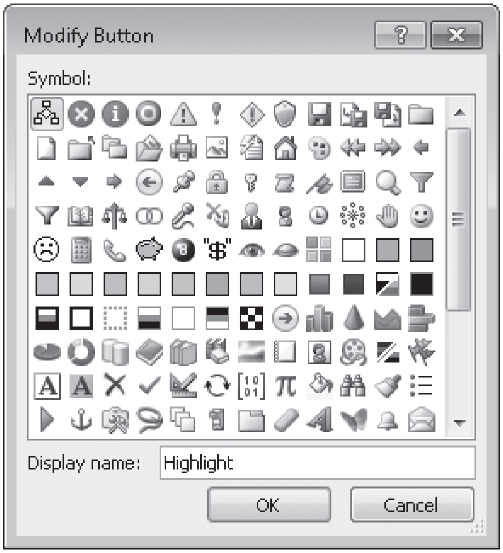
□ Show Quick Access Toolbar below the Ribbon

See Also For more information about customizing the Quick Access Toolbar, see “Customizing the Excel 2010 Program Window” in Chapter 1, “Setting Up a Workbook.”

When you display the Popular Commands command group, you’ll see that the last item in the command pane is View Macros. When you click the View Macros item, click the Add button, and then click OK, Excel adds the command to the Quick Access Toolbar and closes the Excel Options dialog box. Clicking the View Macros button on the Quick Access Toolbar displays the Macro dialog box, which saves a significant amount of time compared to displaying the View tab and moving the mouse to the far right edge of the ribbon.

If you prefer to run a macro without having to display the Macro dialog box, you can do so by adding a button representing the macro to the Quick Access Toolbar. Clicking that button runs the macro immediately, which is very handy when you create a macro for a task you perform frequently. To add a button representing a macro to the Quick Access Toolbar, click the Customize Quick Access Toolbar button at the right edge of the Quick Access Toolbar, and then click More Commands to display the Customize The Quick Access Toolbar page of the Excel Options dialog box. From there, in the Choose Commands From list, click Macros. Click the macro you want represented on the Quick Access Toolbar, click Add, and then click OK.

If you add more than one macro button to the Quick Access Toolbar or if you want to change the button that represents your macro on the Quick Access Toolbar, you can select a new button from more than 160 options. To assign a new button to your macro, click the macro item in the Customize Quick Access Toolbar pane and click the Modify button to display your choices. Click the symbol you want, type a new text value to appear when a user points to the button, and then click OK twice (the first time to close the Modify Button dialog box and the second to close the Excel Options dialog box).



Display name: Highlight

Finally, you can have Excel run a macro when you click a shape in your workbook. Assigning macros to shapes enables you to create “buttons” that are graphically richer than those available on the Quick Access Toolbar. If you’re so inclined, you can even create custom button layouts that represent other objects, such as a remote control. To run a macro when you click a shape, right-click the shape, and then click Assign Macro on the shortcut menu that opens. In the Assign Macro dialog box, click the macro you want to run when you click the shape, and then click OK.

Important When you assign a macro to run when you click a shape, don’t change the name of the macro that appears in the Assign Macro dialog box. The name that appears refers to the object and what the object should do when it is clicked; changing the macro name breaks that connection and prevents Excel from running the macro.

In this exercise, you'll add the View Macros button to the Quick Access Toolbar, add a macro button to the Quick Access Toolbar, assign a macro to a workbook shape, and then run the macros.

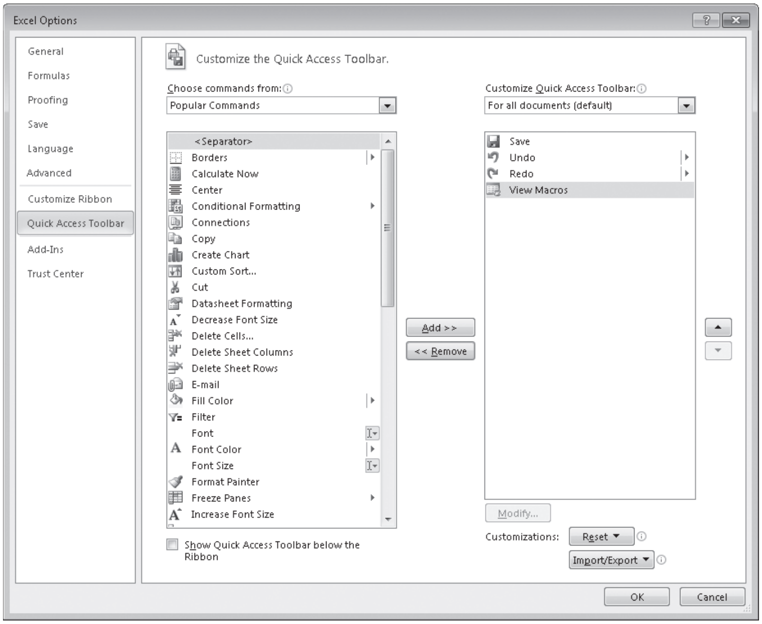
SET UP You need the PerformanceDashboard_start workbook located in your Chapter12 practice file folder to complete this exercise. Open the PerformanceDashboard_start workbook, click the Enable Content button on the Message Bar (if necessary), and save the workbook as *PerformanceDashboard*. Then follow the steps.

1. On the Quick Access Toolbar, click the **Customize Quick Access Toolbar** button, and then click **More Commands**.

   The Customize The Quick Access Toolbar page of the Excel Options dialog box opens, displaying the Popular Commands category in the Choose Commands From pane.

2. In the list of available commands, click **View Macros**.
3. Click **Add**.

   The View Macros command appears in the Customize Quick Access Toolbar pane.



Excel Options

Customize the Quick Access Toolbar.

Choose commands from: Popular Commands

Customize Quick Access Toolbar: For all documents (default)

4. In the **Choose commands from** list, click **Macros**.

   The available macros appear in the pane below.

5. In the **Choose commands from** pane, click **SavingsHighlight**.

   Troubleshooting If macros in the workbook are not enabled, the SavingsHighlight macro will not appear in the list.

6. Click **Add**.

   The SavingsHighlight macro appears in the Customize Quick Access Toolbar pane.

7. In the **Customize Quick Access Toolbar** pane, click the **SavingsHighlight** command.

8. Click **Modify**.

   The Modify Button dialog box opens.

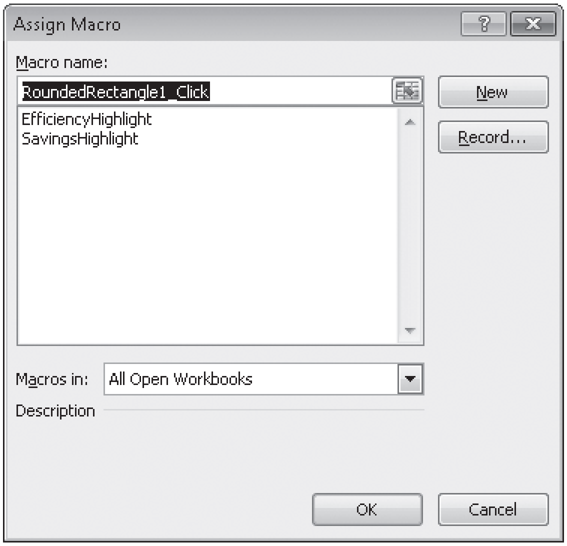
9. Click the blue button with the white circle inside it (the fourth button from the left on the top row).

10. Click **OK** twice to close the **Modify Button** dialog box and the **Excel Options** dialog box.

    The Excel Options dialog box closes, and the View Macros and SavingsHighlight buttons appear on the Quick Access Toolbar.

11. On the worksheet, right-click the **Show Efficiency** shape, and then click **Assign Macro**.

    The Assign Macro dialog box opens.



Assign Macro

Macro name: RoundedRectangle1_Click

EfficiencyHighlight
SavingsHighlight

Macros in: All Open Workbooks

Description

12. Click **EfficiencyHighlight**, and then click **OK**.

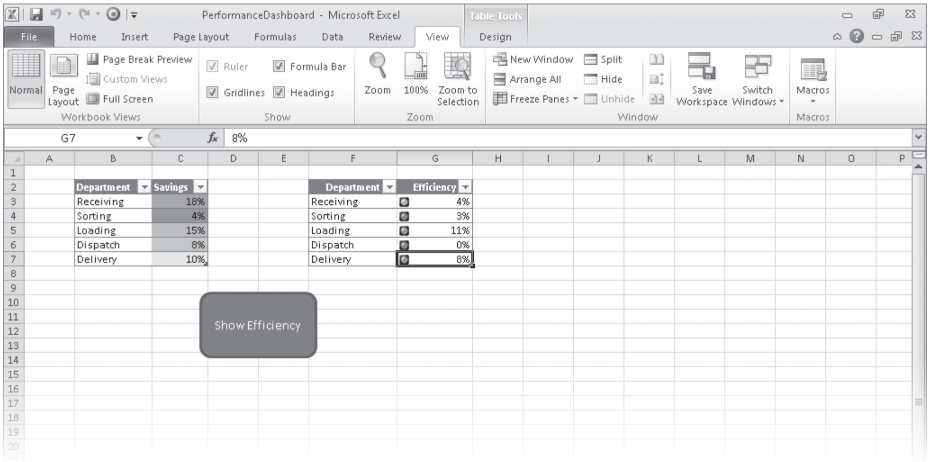
    The Assign Macro dialog box closes.

13. On the Quick Access Toolbar, click the **SavingsHighlight** button.

    Excel runs the macro, which applies a conditional format to the values in the Savings column of the table on the left.

14. Click the **Show Efficiency** shape.

    Excel runs the macro, which applies a conditional format to the values in the Efficiency column of the table on the right.



Department	Savings
Receiving	18%
Sorting	4%
Loading	15%
Dispatch	8%
Delivery	10%

Department	Efficiency
Receiving	● 4%
Sorting	● 3%
Loading	● 11%
Dispatch	● 0%
Delivery	● 8%

15. On the Quick Access Toolbar, click the **Save** button to save your work.

CLEAN UP Close the PerformanceDashboard workbook.

Running Macros When a Workbook Is Opened

One advantage of writing Excel macros in VBA is that you can have Excel run a macro whenever a workbook is opened. For example, if you use a worksheet for presentations, you can create macros that render the contents of selected cells in bold type, italic, or different typefaces to set the data apart from data in neighboring cells. If you close a workbook without removing that formatting, however, the contents of your workbook will still have that formatting applied when you open it. Although this is not a catastrophe, returning the workbook to its original formatting might take a few seconds to accomplish.

Instead of running a macro manually, or even from a toolbar button or a menu, you can have Excel run a macro whenever a workbook is opened. The trick of making that happen is in the name you give the macro. Whenever Excel finds a macro with the name Auto_Open, it runs the macro when the workbook to which it is attached is opened.

Tip If you have your macro security set to the Disable With Notification level, clicking the Options button that appears on the Message Bar, selecting the Enable This Content option, and then clicking OK allows the Auto_Open macro to run.

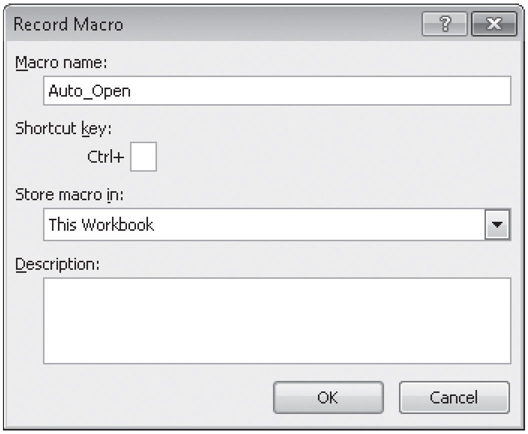
In this exercise, you'll create and test a macro that runs whenever someone opens the workbook to which it is attached.

SET UP You need the RunOnOpen_start workbook located in your Chapter12 practice file folder to complete this exercise. Open the RunOnOpen_start workbook, click the Enable Content button on the Message Bar (if necessary), and save the workbook as *RunOnOpen*. Then follow the steps.

1. On the **View** tab, in the **Macros** group, click the **Macros** arrow and then, in the list that appears, click **Record Macro**.

   The Record Macro dialog box opens.

2. In the **Macro name** box, delete the existing name, and then type **Auto_Open**.



Record Macro

Macro name: Auto_Open

Shortcut key: Ctrl+

Store macro in: This Workbook

Description:

3. Click **OK**.

   The Record Macro dialog box closes.

4. Select the cell range **B3:C11**.

5. On the **Home** tab, in the **Font** group, click the **Bold** button twice.

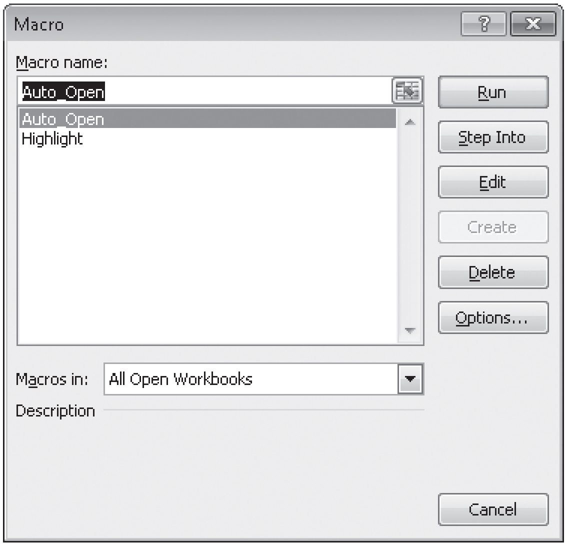
   The first click of the Bold button formats all the selected cells in bold; the second click removes the bold formatting from all the selected cells.

6. Click cell **C11** and then, in the **Macros** list, click **Stop Recording**.

   Excel stops recording your macro.

7. In the **Macros** list, click **View Macros**.

   The Macro dialog box opens.


Macro

Macro name: Auto_Open

Auto_Open
Highlight

Macros in: All Open Workbooks

Description

8. Click **Highlight**, and then click **Run**.

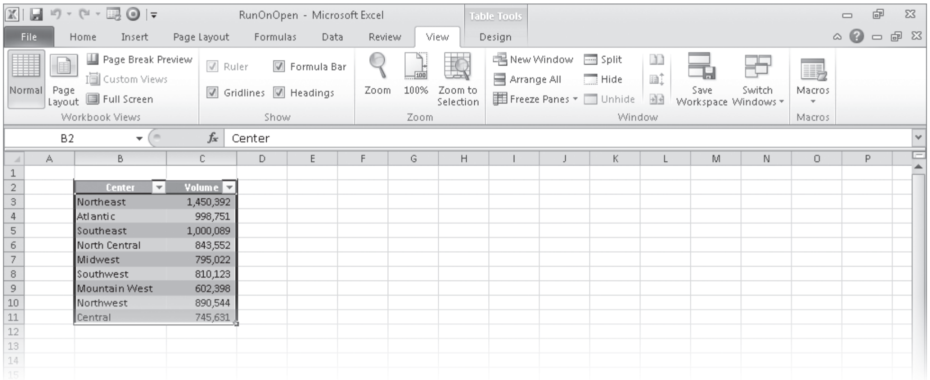
   The contents of cells C4, C6, and C10 appear in bold type.

9. On the Quick Access Toolbar, click the **Save** button to save your work.

10. Click the **Close** button to close the **RunOnOpen** workbook.

11. Click the **File** tab and then, in the **Recent Documents** list, click **RunOnOpen.xlsm**. If a warning appears, click **Enable Content**, and then click **OK** to enable macros.

    RunOnOpen opens, and the contents of cells C4, C6, and C10 change immediately to regular type.


RunOnOpen - Microsoft Excel

B2 Center

Center	Volume
Northeast	1,450,392
Atlantic	998,751
Southeast	1,000,089
North Central	843,552
Midwest	795,022
Southwest	810,123
Mountain West	602,398
Northwest	890,544
Central	745,631

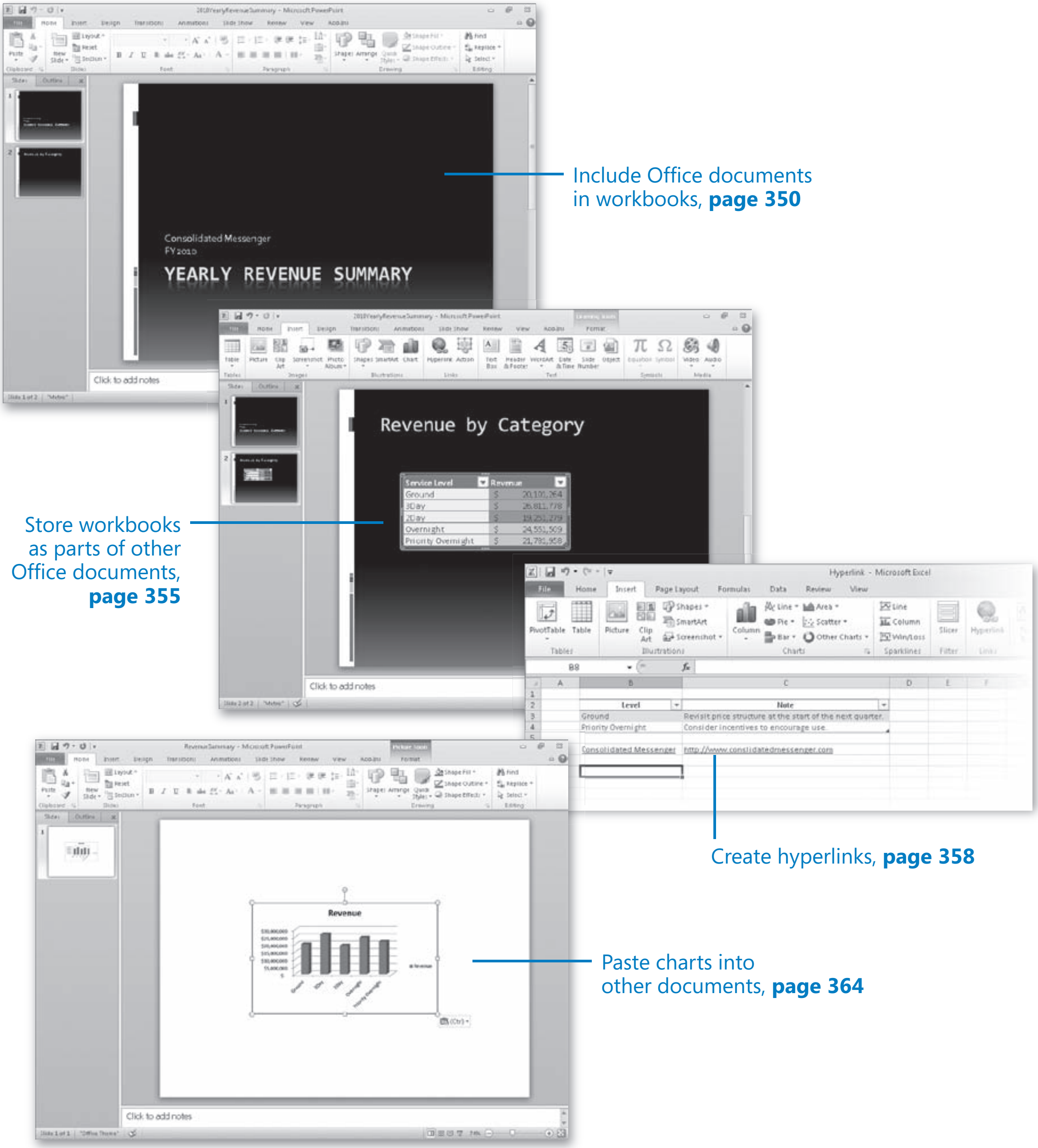
12. On the Quick Access Toolbar, click the **Save** button to save your work.

CLEAN UP Close the RunOnOpen workbook. If you're not continuing directly to the next chapter, exit Excel.

Key Points

- Macros are handy tools you can use to perform repetitive tasks quickly, such as inserting blocks of text.
- You don't have to be a programmer to use macros; you can record your actions and have Excel save them as a macro.
- Excel uses macro-enabled workbook types, which have the file extensions .xlsm (a macro-enabled workbook) and .xltm (a macro-enabled template workbook).
- If you're curious about what a macro looks like, you can display it in the Visual Basic Editor. If you know a little VBA, or if you just want to experiment, feel free to modify the macro code and see what happens.
- You can create Quick Access Toolbar buttons and shapes that, when clicked, run a macro.
- If you want a macro to run whenever you open a workbook, create a macro named Auto_Open.

Chapter at a Glance



Consolidated Messenger
FY2010

YEARLY REVENUE SUMMARY

Revenue by Category

Service Level	Revenue
Ground	$ 20,101,264
3Day	$ 26,811,778
2Day	$ 19,251,279
Overnight	$ 24,551,509
Priority Overnight	$ 21,791,958

Level	Note
Ground	Revisit price structure at the start of the next quarter.
Priority Overnight	Consider incentives to encourage use.

Consolidated Messenger http://www.consolidatedmessenger.com

Include Office documents in workbooks, **page 350**

Store workbooks as parts of other Office documents, **page 355**

Create hyperlinks, **page 358**

Paste charts into other documents, **page 364**