CHAPTER 1

RECORDED MACROS

INFOCUS

WPL_E845

Macros provide a way of automating operations in Microsoft Excel. One of the easiest ways to create a macro is to use the macro recorder.

The macro recorder records the actions that you perform through the keyboard and mouse. It creates a program from these steps that you can run any time you need to repeat the actions.

In this way, macros enable you to work more efficiently.

In this session you will:

- ✓ gain an understanding of macros in ***Excel***
- ✓ learn how to set macro security
- ✓ learn how to save a document as macro enabled
- ✓ learn how to record a simple macro
- ✓ learn how to run a recorded macro
- ✓ learn how to record relative cell references
- ✓ learn how to run a macro with relative references
- ✓ learn how to view a macro
- ✓ learn how to edit a macro
- ✓ learn how to assign a macro to the toolbar
- ✓ learn how to run a macro from the toolbar
- ✓ learn how to assign a macro to the ***Ribbon***
- ✓ learn how to assign a keyboard shortcut to a macro
- ✓ learn how to delete a macro
- ✓ learn how to copy a macro.

UNDERSTANDING EXCEL MACROS

Macros were added to Excel many years ago to provide a way to automate routine operations. In earlier versions, macros enabled you to record the keystrokes used to perform an operation.

These days, macros have evolved into a full programming language, allowing you to fully automate virtually every facet of workbook production.

What is a Macro?

A macro is simply a programmed set of instructions that tell Microsoft Excel (very specifically) what it should do. Macros are written or recorded in a procedure.

How Are Macros Created?

Excel offers two main ways of creating macros. Macros can be ***recorded*** using the built-in macro recorder, which records what you do and then converts this into a macro program. This is a great way of creating macros for performing routine, complex or boring repetitious tasks. Once recorded, these tasks can be performed quickly and accurately over and over again using the macro.

Macros can also be developed from scratch. In other words, you can ***type*** the programming steps yourself rather than recording them using the built-in macro recorder.

You can also use a combination of the two techniques to fine tune or change the functionality of a macro.

What Types of Macros Are There?

As a very broad generalisation, there are two types of macros – ***global*** and ***local***.

A ***global*** macro is available to all of the workbooks that you create. For example, you might have a macro that adds your company name and details to the footer of a workbook. Since you want all workbooks to have this, the macro to add the footer should be available to all workbooks and would therefore need to be global.

A ***local*** macro is one that is available only to one particular workbook. For example, you might have a monthly report workbook that needs to have information imported into it from other sources. You can set up a macro that will conduct the importation for you so that you don't have to remember or perform the steps each time.

Where Are Macros Located?

Macros are either attached to the current document or located in a ***Personal Macro Workbook*** which makes them available to all workbooks (i.e. global). When you create a macro, if you select ***Personal Macro Workbook*** as the location in which to store it, a hidden personal macro workbook called ***Personal.xlsb*** is created and the macro is stored within it. This then makes it available each time you open Excel.

How Do You Access Macro Code?

Macros can only be viewed using the ***Visual Basic Editor*** which is accessed via the tool of the same name on the ***Developer*** tab on the ribbon. You can also press Alt + F11.

What is VBA?

VBA (***Visual Basic for Applications***) is the programming language used to create macros. Earlier versions of Excel used more primitive versions of this language. VBA is a common programming language found in virtually all Microsoft Office applications. Once you have learned it for one product, you can easily adapt what you've learned to the other products.

Do I Need to be a Programmer to Create Macros?

Absolutely not! While macros may appear cryptic and difficult to understand at first, tools such as the macro recorder make creating macros easy and effortless.

SETTING MACRO SECURITY

Recording or writing macros allows you to hack into Excel and manipulate the application. One of the consequences of this is that macros become a potential source of viruses. To reduce the risk of viruses, Microsoft has a ***Trust Centre*** that allows you to enable or disable macros based on whether or not they are stored in a trusted location or have a digital signature.

Try This Yourself:

Open File

Before starting this exercise you MUST ensure that a new, blank workbook is open...

1 If the ***Developer*** tab is not visible, click on the ***File*** tab and select **Options**, click on ***Customise Ribbon*** and click on ***Developer*** in ***Customise the Ribbon*** until it is ticked

2 Click on **[OK]**, then click on the ***Developer*** tab to display the ***Code*** group

3 Click on ***Macro Security*** ⚠ to display the ***Trust Centre*** dialog box

By default all macros are currently disabled unless they are in a trusted location...

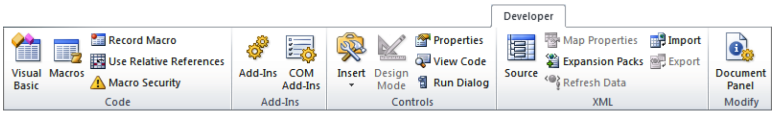
4 Click on ***Trusted Locations*** to see the list of trusted folders

5 Click on **[Add new location]** to display the ***Microsoft Office Trusted Location*** dialog box

6 Click on **[Browse]**, locate the course files folder and click on **[OK]** then click on **[OK]** again

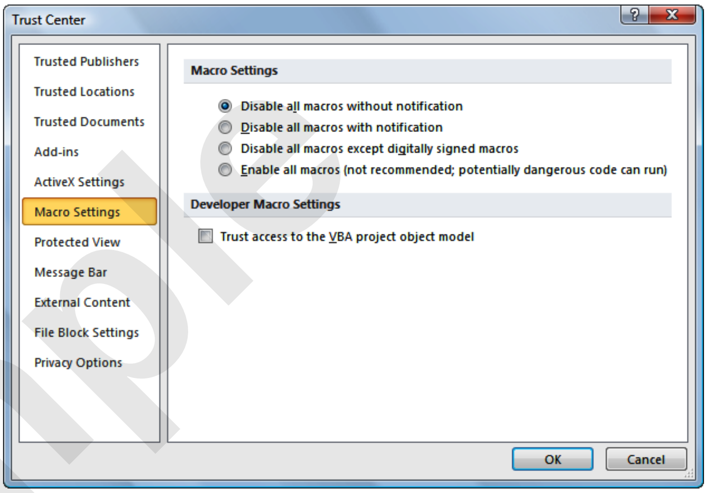
The course files folder will be added to the list of Trusted Locations...

7 Click on **[OK]** to close the dialog box



2

3



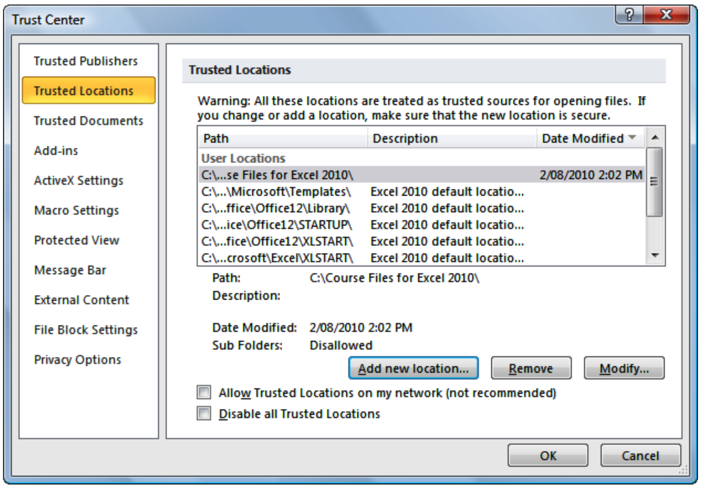
Macro Settings

● 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

6



Trusted Locations

Warning: All these locations are treated as trusted sources for opening files. If you change or add a location, make sure that the new location is secure.

Path	Description	Date Modified
User Locations		
C:\...se Files for Excel 2010\		2/08/2010 2:02 PM
C:\...\Microsoft\Templates\	Excel 2010 default locatio...	
C:\...ffice\Office12\Library\	Excel 2010 default locatio...	
C:\...ice\Office12\STARTUP\	Excel 2010 default locatio...	
C:\...fice\Office12\XLSTART\	Excel 2010 default locatio...	
C:\...crosoft\Excel\XLSTART\	Excel 2010 default locatio...	

Path: C:\Course Files for Excel 2010\
Description:

Date Modified: 2/08/2010 2:02 PM
Sub Folders: Disallowed

□ Allow Trusted Locations on my network (not recommended)
□ Disable all Trusted Locations

For Your Reference…

To ***set*** the ***level of macro security***:

1. Click on the ***Developer*** tab
2. Click on ***Macro Security*** ⚠
3. Click on the required level of security
4. Click on **[OK]**

Handy to Know…

- A ***digital signature*** is an encrypted electronic stamp used to authenticate a macro or document. This signature confirms that the macro or document originated from the signer and has not been altered.

SAVING A DOCUMENT AS MACRO ENABLED

Microsoft Excel 2010 has several different file formats that controls whether or not macros can be saved with the file. The default workbook format of ***.xlsx*** does not allow macros to be saved with the workbook. To ensure that macro code can be saved, you must change the workbook type to ***.xlsm*** which is known as an ***Excel Macro-Enabled Workbook***.

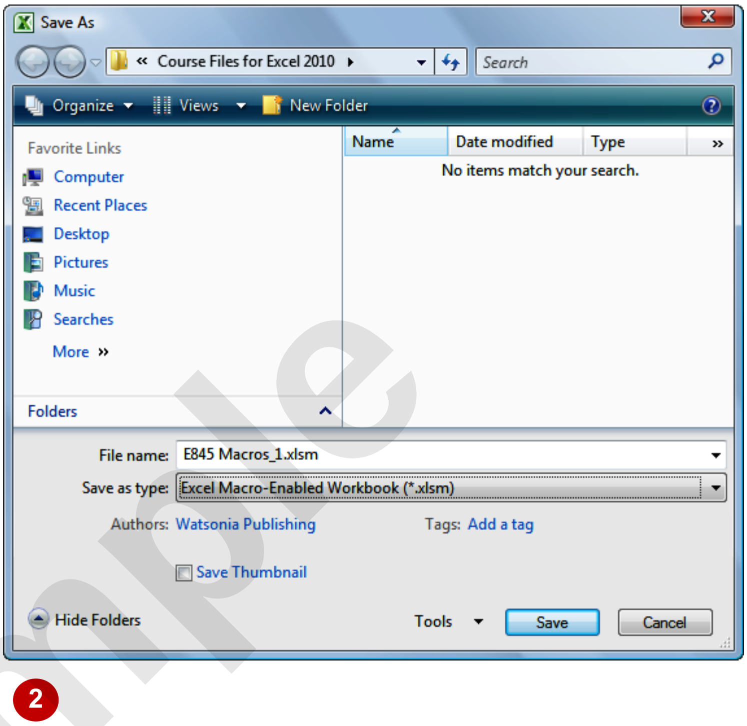
Try This Yourself:

Open File

Before starting this exercise you MUST open the file E845 Macros_1.xlsx...

1. Click on the ***File*** tab and select **Save As** to display the ***Save As*** dialog box
2. Click on the drop arrow ▾ for ***Save as type*** and click on ***Excel Macro-Enabled Workbook (*.xlsm)***
3. Click on **[Save]**

The filename, shown in the title bar, will reflect the file type



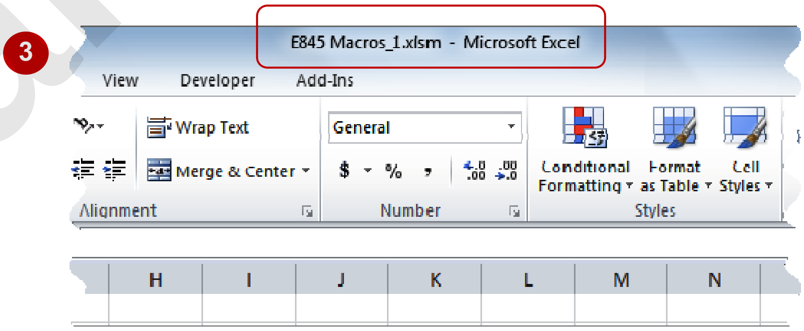
No items match your search.

File name: E845 Macros_1.xlsm

Save as type: Excel Macro-Enabled Workbook (*.xlsm)

Authors: Watsonia Publishing

Tags: Add a tag



For Your Reference…

To ***save*** a ***workbook*** as ***macro-enabled***:

1. Click on the ***File*** tab and select **Save As**
2. Click on the drop arrow ▾ for ***Save as type*** and click on ***Excel Macro-Enabled Workbook (*.xlsm)***
3. Click on **[OK]**

Handy to Know…

- Excel 2010 files saved as either ***.xlsx or xltx*** cannot be used to store macros, while those saved as either ***.xlsm*** or ***.xltm*** can be used to store macros. You can create a macro in a workbook that is not macro-enabled, but you won't be able to save it.

RECORDING A SIMPLE MACRO

Simple ***macros can be recorded*** to perform steps that you need to repeat often. For example, you might need to enter the names of your company's departments in each workbook you create. When you create a macro, you need to assign it a unique name and a location to store it. When you record the steps required, the recorder takes care of writing the macro commands.

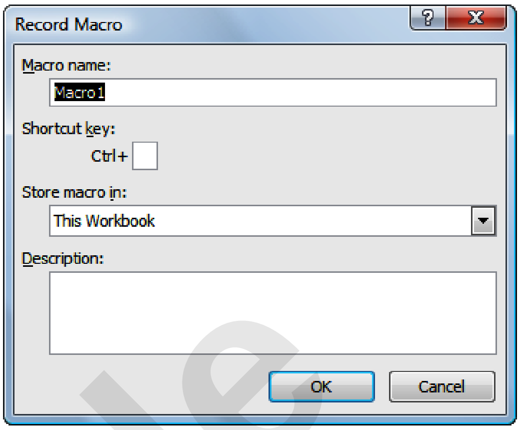
Try This Yourself:

Same File *Continue using the previous file with this exercise, or open the file E845 Macros_2.xlsm...*

1. Click on the ***Developer*** tab of the ***Ribbon***, then click on ***Record Macro*** Alt to display the ***Record Macro*** dialog box
2. Type **DepartmentNames** (no spaces) in ***Macro name***
3. Ensure that ***Store macro in*** is set to ***This Workbook***, then click in ***Description*** and type **Enter the names of the company departments**
4. Click on **[OK]** to begin recording

   The Stop Recording tool appears in the Code group on the Ribbon and also in the Status bar, indicating that recording is in progress...
5. Click on cell ***A5*** then enter the text as shown – press Enter after each entry, including the last one
6. Click on ***Stop Recording*** in the ***Code*** group on the ***Ribbon***

1



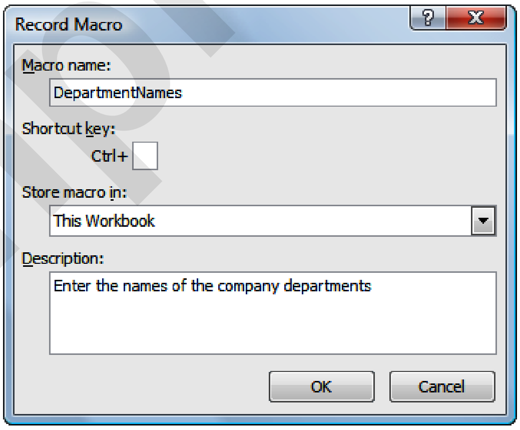
Macro name: Macro1

Shortcut key: Ctrl+

Store macro in: This Workbook

Description:

3



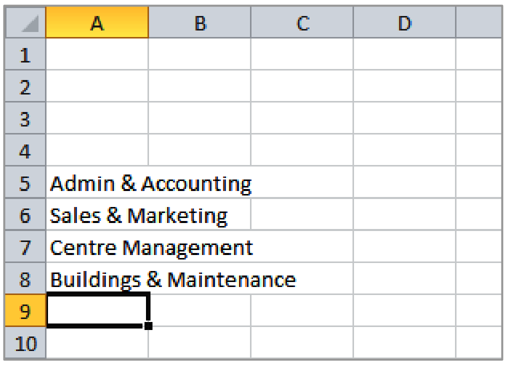
Macro name: DepartmentNames

Shortcut key: Ctrl+

Store macro in: This Workbook

Description: Enter the names of the company departments

5



	A	B	C	D
1				
2				
3				
4				
5	Admin & Accounting			
6	Sales & Marketing			
7	Centre Management			
8	Buildings & Maintenance			
9				
10				

For Your Reference…

To ***record*** a ***macro***:

1. Click on ***Record Macro*** on the ***Developer*** tab
2. Type a ***Macro name*** and select a location
3. Click on **[OK]** and perform the steps
4. Click on ***Stop Recording***

Handy to Know…

- If you want to be able to access a macro from more than one workbook, store it in the ***Personal Macro Workbook***.
- The shortcut key combination option in the ***Record Macro*** dialog box enables you to run a macro without having to access the ***Ribbon***.

RUNNING A RECORDED MACRO

Once a macro has been recorded, it can be played or run as often as you need it. All you need to know is which macros are available and what they do. The description is very important at this stage, especially if you haven't used your macros for a while. The description of a macro can be viewed in the ***Macro*** dialog box, displayed when you want to ***run a recorded macro***.

Try This Yourself:

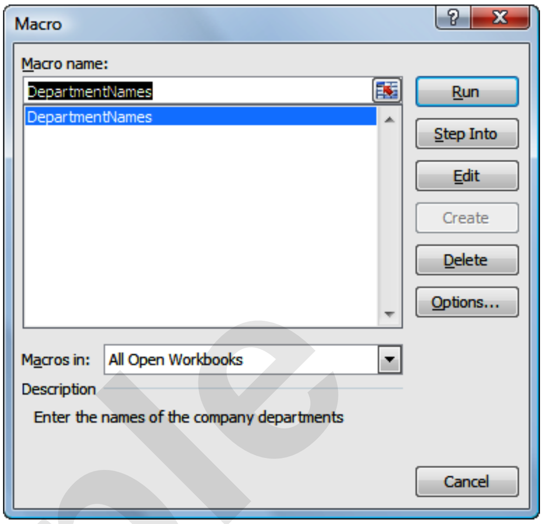
Same File

Continue using the previous file with this exercise, or open the file E845 Macros_3.xlsm...

1. Select the range ***A5:A8***
2. Press Del to remove the department name labels
3. Click on ***Macros*** on the ***Developer*** tab to display the ***Macros*** dialog box
4. Click on ***DepartmentNames***, if it is not already selected, then click on **[Run]**

   The macro will run and the department names will be inserted in the correct cells...
5. Add the rest of the data as shown, then resize the columns to fit the data

3



Macro name: DepartmentNames

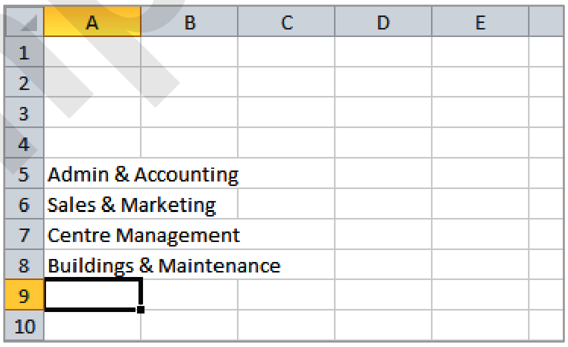
DepartmentNames

Macros in: All Open Workbooks

Description

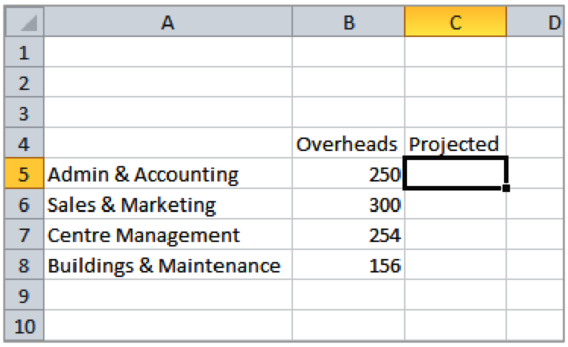
Enter the names of the company departments

4



	A	B	C	D	E
1					
2					
3					
4					
5	Admin & Accounting				
6	Sales & Marketing				
7	Centre Management				
8	Buildings & Maintenance				
9					
10					

5



	A	B	C	D
1				
2				
3				
4		Overheads	Projected	
5	Admin & Accounting	250		
6	Sales & Marketing	300		
7	Centre Management	254		
8	Buildings & Maintenance	156		
9				
10				

For Your Reference…

To ***run*** a ***recorded macro***:

1. Click on ***Macros*** on the ***Developer*** tab
2. Click on the ***Macro name***
3. Click on **[Run]**

Handy to Know…

- The ***Macros*** dialog box can also be displayed using the keyboard shortcut Alt + F8.
- You can also press the keyboard shortcut assigned to the macro, to run it.

Relative Cell References

Unlike ***absolute cell references*** that identify specific cells such as *A5*, ***relative cell references*** are an offset from the current active cell. For example ***RC[-1]*** refers to the cell one column to the left of the currently active cell. You can force Excel to record ***relative cell references*** so that your macros can be used in any cell in a workbook.

Try This Yourself:

Same File

Continue using the previous file with this exercise, or open the file E845 Macros_4.xlsm...

1. Click on cell ***C5***

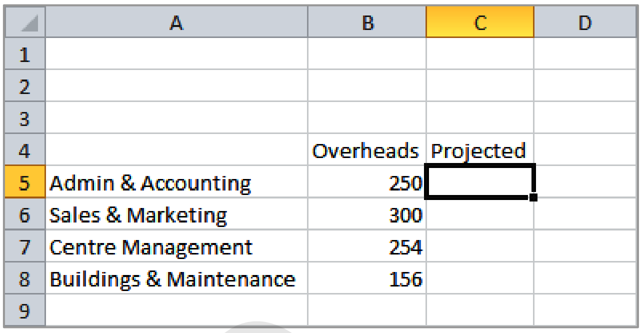
   This is where we want the macro to place the new value...

2. Click on ***Use Relative References***, on the ***Developer*** tab of the ***Ribbon***, to activate the option
3. Click on ***Record Macro*** on the ***Developer*** tab, complete the macro information as shown then click on **[OK]** to start recording
4. Type **=TRUNC(B5*1.15)** and press Enter

   The calculated value will appear in C5...

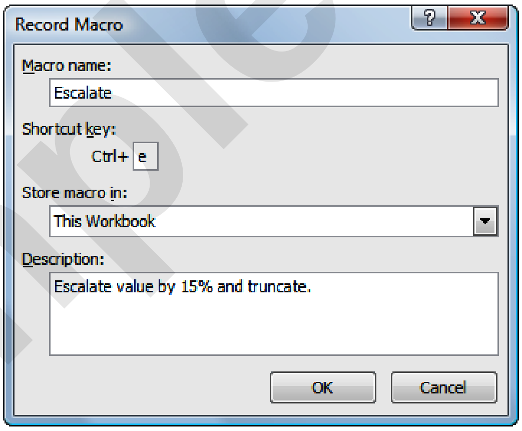
5. Click on ***Stop Recording*** in the ***Code*** group on the ***Developer*** tab
6. Click on ***Use Relative References*** to turn off the option

1



	A	B	C	D
1				
2				
3				
4		Overheads	Projected	
5	Admin & Accounting	250		
6	Sales & Marketing	300		
7	Centre Management	254		
8	Buildings & Maintenance	156		
9				

3



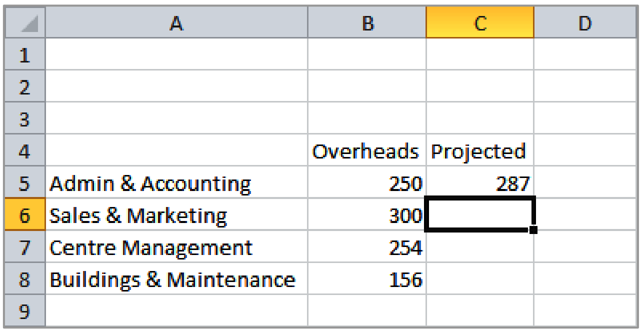
Macro name: Escalate

Shortcut key: Ctrl+ e

Store macro in: This Workbook

Description: Escalate value by 15% and truncate.

4



	A	B	C	D
1				
2				
3				
4		Overheads	Projected	
5	Admin & Accounting	250	287	
6	Sales & Marketing	300		
7	Centre Management	254		
8	Buildings & Maintenance	156		
9				

The macro enters a formula into the current cell. The formula calculates 15% more than the value one cell to the left and truncates that value to an integer. This final value is entered into the current cell.

For Your Reference…

To ***record*** a ***macro with relative references***:

1. On the ***Developer*** tab, click on ***Use Relative References*** until it is selected
2. Click on ***Record Macro***
3. Fill in the macro details and click on **[OK]**

For Your Reference (cont'd)…

4. Perform the steps
5. Click on ***Stop Recording*** on the ***Developer*** tab

Running A Macro With Relative References

When you record a macro with ***absolute cell references***, it records the exact cell references. No matter which cell is active, when you run the macro, the actions will be performed on the cells that you used when recording the steps. With ***relative cell references*** the macro will run relative to the currently active cell, so you need to take a bit more care with positioning.

Try This Yourself:

Same File

Continue using the previous file with this exercise, or open the file E845 Macros_5.xlsm...

1. Ensure that ***C6*** is active

   This is where we want the macro to perform the recorded steps...

2. Click on ***Macros*** on the ***Developer*** tab of the ***Ribbon***, to display the ***Macro*** dialog box

3. Click on ***Escalate***, then click on **[Run]**

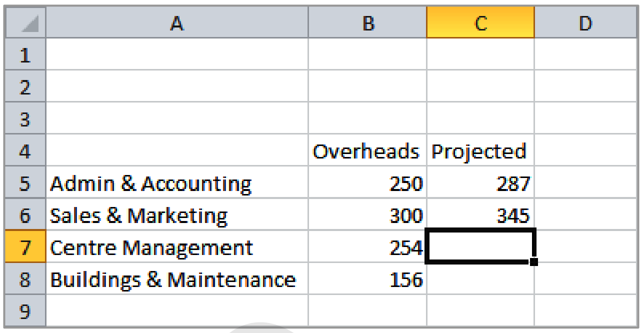
   The formula is entered into the cell and the value resulting from the calculation is displayed. You can also run the macro using the shortcut key...

4. Ensure that ***C7*** is selected, then press Ctrl + E to run the macro again

   The value is entered...

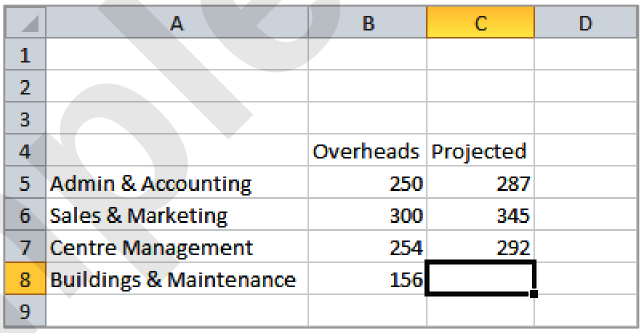
5. Ensure that ***C8*** is selected then press Ctrl + E to complete the column

3



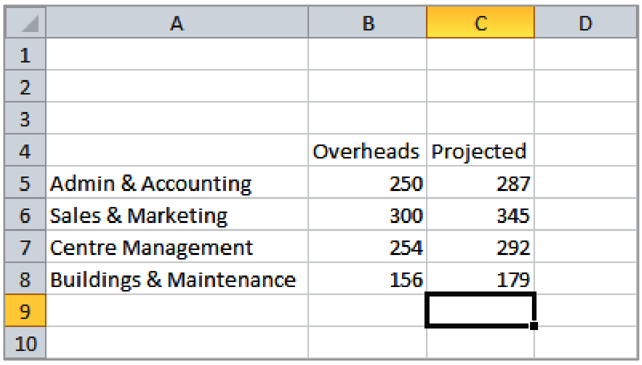
	A	B	C	D
1				
2				
3				
4		Overheads	Projected	
5	Admin & Accounting	250	287	
6	Sales & Marketing	300	345	
7	Centre Management	254		
8	Buildings & Maintenance	156		
9				

4



	A	B	C	D
1				
2				
3				
4		Overheads	Projected	
5	Admin & Accounting	250	287	
6	Sales & Marketing	300	345	
7	Centre Management	254	292	
8	Buildings & Maintenance	156		
9				

5



	A	B	C	D
1				
2				
3				
4		Overheads	Projected	
5	Admin & Accounting	250	287	
6	Sales & Marketing	300	345	
7	Centre Management	254	292	
8	Buildings & Maintenance	156	179	
9				
10				

For Your Reference…

To ***run*** a ***macro with relative cell referencing***:

1. Click on the cell where you want the macro to perform
2. Click on ***Macros*** on the ***Developer*** tab
3. Select the macro and click on **[Run] OR**
1. Press the ***shortcut key*** combination

Handy to Know…

- By holding down Shift when you assign a shortcut key combination in the ***Record Macro*** dialog box, you have access to at least another 26 possible combinations. Just be careful that you are not overriding built-in shortcuts already in place in Excel.

VIEWING A MACRO

When you record a macro, you actually create a series of commands in a programming language called ***Visual Basic for Applications (VBA)***. Each time you run the macro, the code is executed. VBA can be ***viewed and edited*** using the ***Visual Basic Editor***. The advantage of using the editor is that you can easily change, copy or delete macro code.

Try This Yourself:

Same File

Continue using the previous file with this exercise, or open the file E845 Macros_6.xlsm...

1 On the ***Developer*** tab of the ***Ribbon***, click on ***Macros*** to display the ***Macro*** dialog box

2 Click on ***DepartmentNames***, then click on **[Edit]**

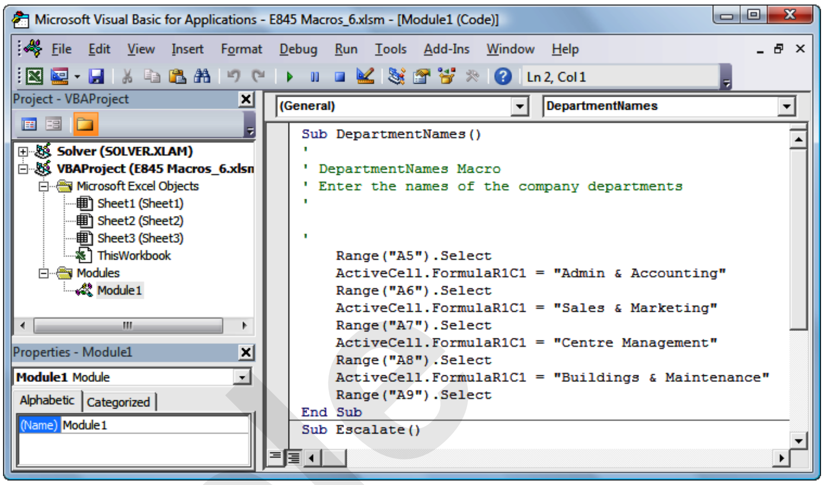
The Visual Basic Editor will be displayed. The code for the selected macro is shown in the Module window...

3 Spend a few moments reading through each line of code – see if you can work out what each line does

4 Scroll down the Module window to display the ***Escalate*** code

Excel has placed both programs into the same Module...

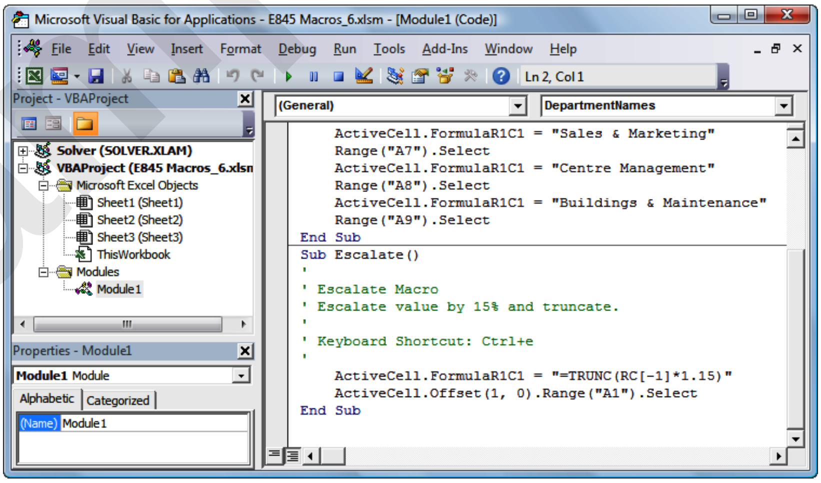
5 Select **File** > **Close and Return to Microsoft Excel** to close the Visual Basic Editor window



```
Sub DepartmentNames()
'
' DepartmentNames Macro
' Enter the names of the company departments
'

'
    Range("A5").Select
    ActiveCell.FormulaR1C1 = "Admin & Accounting"
    Range("A6").Select
    ActiveCell.FormulaR1C1 = "Sales & Marketing"
    Range("A7").Select
    ActiveCell.FormulaR1C1 = "Centre Management"
    Range("A8").Select
    ActiveCell.FormulaR1C1 = "Buildings & Maintenance"
    Range("A9").Select
End Sub
Sub Escalate()
```

2



```
    ActiveCell.FormulaR1C1 = "Sales & Marketing"
    Range("A7").Select
    ActiveCell.FormulaR1C1 = "Centre Management"
    Range("A8").Select
    ActiveCell.FormulaR1C1 = "Buildings & Maintenance"
    Range("A9").Select
End Sub
Sub Escalate()
'
' Escalate Macro
' Escalate value by 15% and truncate.
'
' Keyboard Shortcut: Ctrl+e
'
    ActiveCell.FormulaR1C1 = "=TRUNC(RC[-1]*1.15)"
    ActiveCell.Offset(1, 0).Range("A1").Select
End Sub
```

4

For Your Reference…

To ***view*** a ***macro***:

1. Click on ***Macros*** on the ***Developer*** tab
2. Select a macro
3. Click on **[Edit]**

Handy to Know…

- When viewing macros in the Visual Basic Editor window, you will notice that some of the code is in green. These are known as ***comments*** and are used to explain the what, why and who of code. This information is ignored when the macro is run, but can be invaluable in understanding the code.

EDITING A MACRO

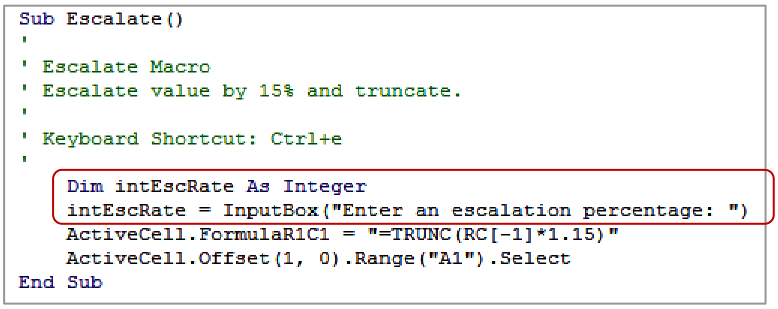
You might find that you want to make a minor change to your macro or insert additional code to improve the macro's functionality. Macro instructions can be ***edited*** in the ***VBA Editor*** window. You need to have a reasonable grasp of the programming language before you can confidently make changes, but you will find that the ***Help*** system in the editor window is very useful.

Try This Yourself:

Same File — *Continue using the previous file with this exercise, or open the file E845 Macros_7.xlsm...*

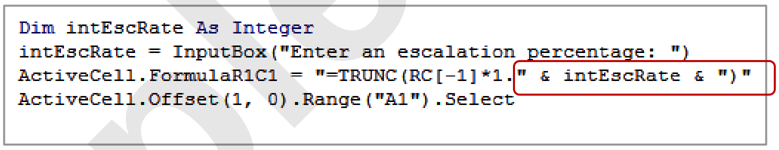
1. On the ***Developer*** tab of the ***Ribbon***, click on ***Macros***, to display the ***Macro*** dialog box
2. Click on ***Escalate*** then click on **[Edit]** to display the code in the ***Visual Basic Editor*** window
3. Type the additional lines as shown, using Tab to indent lines and Enter for new lines
4. Modify the formula line as shown
5. Select **File** > **Save E845 Macros_7.xlsm**
6. Select **File** > **Close and Return to Microsoft Excel**
7. Click on cell ***D5*** and press Ctrl + E to see a dialog box
8. Type **50** and click on **[OK]**

The edited macro enables you to now specify an escalation % at the time it is run. Here, the macro increases the value 287 by 50% and truncates the result



```
Sub Escalate()
'
' Escalate Macro
' Escalate value by 15% and truncate.
'
' Keyboard Shortcut: Ctrl+e
'
    Dim intEscRate As Integer
    intEscRate = InputBox("Enter an escalation percentage: ")
    ActiveCell.FormulaR1C1 = "=TRUNC(RC[-1]*1.15)"
    ActiveCell.Offset(1, 0).Range("A1").Select
End Sub
```

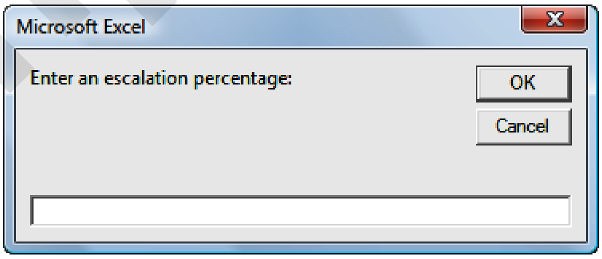
3



```
Dim intEscRate As Integer
intEscRate = InputBox("Enter an escalation percentage: ")
ActiveCell.FormulaR1C1 = "=TRUNC(RC[-1]*1." & intEscRate & ")"
ActiveCell.Offset(1, 0).Range("A1").Select
```

4

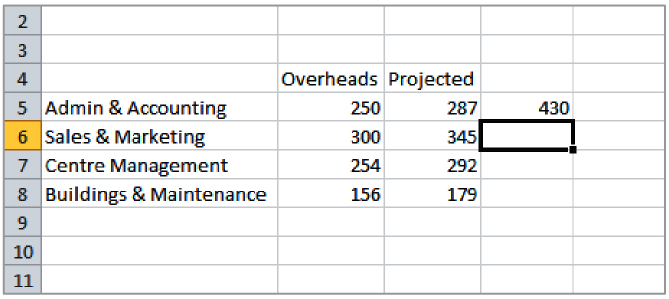
7



Microsoft Excel

Enter an escalation percentage:

8



2					
3					
4		Overheads	Projected		
5	Admin & Accounting	250	287	430	
6	Sales & Marketing	300	345		
7	Centre Management	254	292		
8	Buildings & Maintenance	156	179		
9					
10					
11					

For Your Reference…

To ***edit*** a ***macro***:

1. Click on ***Macros***, select a macro and click on **[Edit]**
2. Make the changes as required
3. Select **File** > **Save…**
4. Select **File** > **Close…**

Handy to Know…

- The ***Dim*** statement in macro code declares (creates) a ***variable*** (temporary holder) by the name given. ***InputBox*** displays a dialog box and the assigned question. The user's response is then placed in the variable.
- You can also press Alt + F11 to display the Visual Basic Editor window.

ASSIGNING A MACRO TO THE TOOLBAR

Running a macro from the ***Macros*** dialog box is not necessarily the most practical way to do it. However, you can create a custom button to place on the ***Quick Access Toolbar*** (QAT) and attach the macro to this button. As the QAT is always visible, the macro is easily accessible. You can also change the name and icon associated with the button.

Try This Yourself:

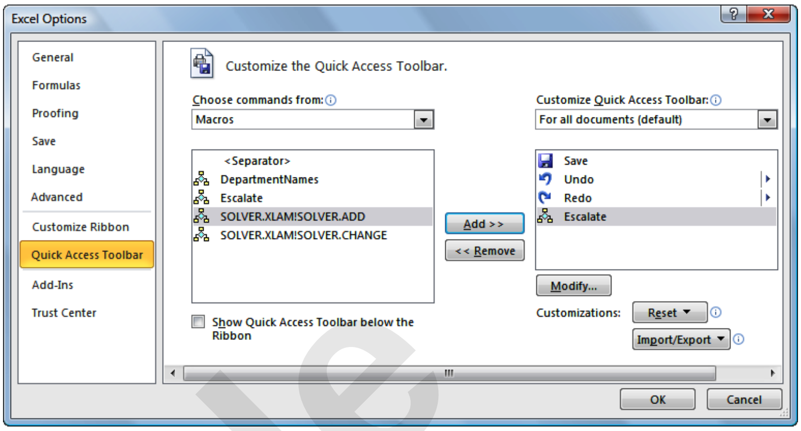
Same File

Continue using the previous file with this exercise, or open the file E845 Macros_8.xlsm...

1. Click on the drop arrow ▾ of the ***Quick Access Toolbar*** and select **More Commands** to display the ***Excel Options*** dialog box
2. Click on the drop arrow ▾ for ***Choose commands from*** and select **Macros** to list the available macros
3. Click on ***Escalate*** then click on **[Add]** to add the macro to the QAT list on the right
4. Click on **[Modify]** to display the ***Modify Button*** dialog box
5. Click on an icon of your choice, then type **Escalate Value** in ***Display name***
6. Click on **[OK]** then click on **[OK]** again

The new button will display in the Quick Access Toolbar...

7. Hover the mouse pointer over the button in the ***Quick Access Toolbar*** to display the name



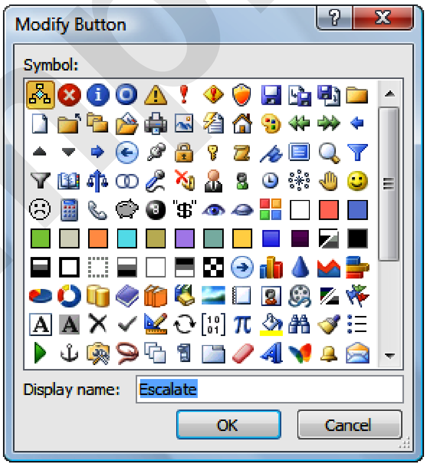
3

Excel Options

Customize the Quick Access Toolbar.

Choose commands from: Macros

Customize Quick Access Toolbar: For all documents (default)

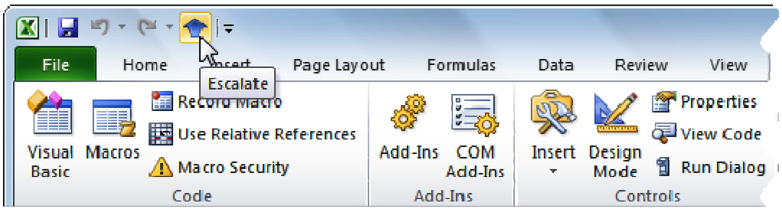


4

Modify Button

Symbol:

Display name: Escalate



7

For Your Reference…

To ***assign*** a ***macro*** to a ***toolbar button***:

1. Click on the drop arrow ▾ for the ***Quick Access Toolbar*** and select **More Commands**
2. Click on the drop arrow ▾ for ***Choose commands from*** and select **Macros**

For Your Reference (cont’d)…

3. Click on the macro and click on **[Add]**
4. Click on **[Modify]** to change the name and/or icon
5. Click on **[OK]**
6. Click on **[OK]** again

Running a Macro From the Toolbar

If you have created a custom button on the Quick Access Toolbar for one of your macros, you can then quickly ***run the macro*** by clicking on the button. The only consideration you need to make is whether or not the position of the cell pointer is important – this is relevant when relative cell references are used in the macro. In this example, the active cell determines which value is escalated.

Try This Yourself:

Same File

Continue using the previous file with this exercise, or open the file E845 Macros_9.xlsm...

1. Click on cell ***D6*** to position the cell pointer

   The calculation will be performed on the cell to the left of the cell pointer...

2. Click on ***Escalate Value*** in the ***Quick Access Toolbar***

   The dialog box will appear asking for the escalation value...

3. Type **50** and click on **[OK]**

   You can also press Enter *after typing the value. The figure will be calculated and placed in the cell...*

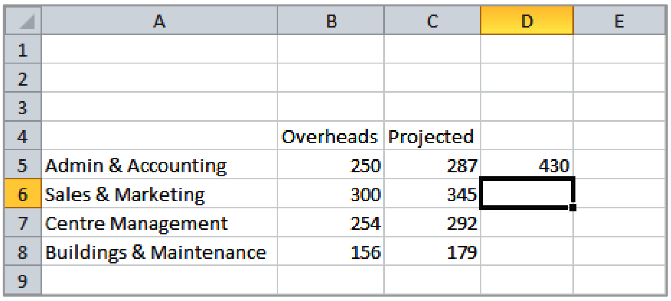
4. Repeat steps ***1*** to ***3*** for the following cells, using the values as shown:

   D7   ***30***

   D8   ***45***

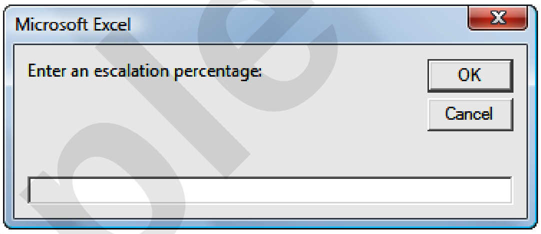
   As you can see, the custom button makes running and re-running the macro very easy

1



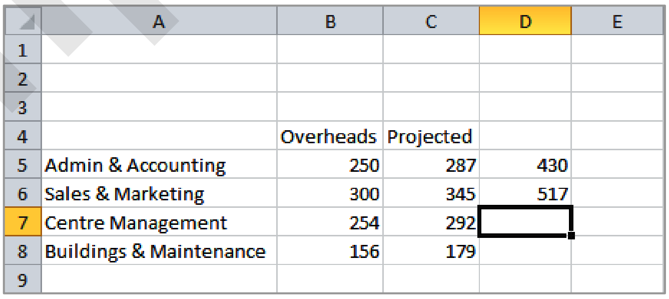
	A	B	C	D	E
1					
2					
3					
4		Overheads	Projected		
5	Admin & Accounting	250	287	430	
6	Sales & Marketing	300	345		
7	Centre Management	254	292		
8	Buildings & Maintenance	156	179		
9					

2



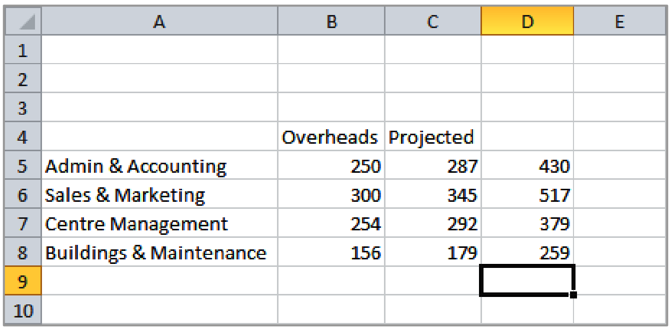
Enter an escalation percentage:

3



	A	B	C	D	E
1					
2					
3					
4		Overheads	Projected		
5	Admin & Accounting	250	287	430	
6	Sales & Marketing	300	345	517	
7	Centre Management	254	292		
8	Buildings & Maintenance	156	179		
9					

4



	A	B	C	D	E
1					
2					
3					
4		Overheads	Projected		
5	Admin & Accounting	250	287	430	
6	Sales & Marketing	300	345	517	
7	Centre Management	254	292	379	
8	Buildings & Maintenance	156	179	259	
9					
10					

For Your Reference…

To ***run*** a ***macro*** assigned to a ***toolbar button***:

1. Position the cell pointer
2. Click on the button in the ***Quick Access Toolbar***

Handy to Know…

- You can remove a button from the ***Quick Access Toolbar*** by right-clicking on the button and selecting **Remove from Quick Access Toolbar**.