



## Level 6 Spreadsheet 6N4089

# Section 6 – Macros and Auditing

### Contents

1.	Macros	2
	What is a macro? .....	2
2.	Record a Macro	2
	Turn on the Developer Tab .....	2
	Set the Security Level.....	3
	Record a Macro.....	3
	Run a Macro .....	5
	Assign a Macro to an Object or Graphic .....	6
	Delete a Macro .....	6
	Edit a macro .....	6
	Document a macro .....	8
	Saving a Macro Enabled Workbook.....	8
3.	Auditing	9
	Trace Precedents .....	9
	Trace Dependents.....	10
	Common Error Codes in Excel.....	10

Trace Errors.....	11
Evaluate Formula .....	12
Show Formulas.....	12
4. References	13
Websites: .....	13



*RMN*

## 1. MACROS

### WHAT IS A MACRO?

An Excel macro is a set of instructions that can be triggered by a keyboard shortcut, toolbar button or an icon in a spreadsheet. Macro means code (set of instructions). Macros are used to eliminate the need to repeat the steps of common tasks over and over, tasks such as; adding or removing rows and columns, protecting or unprotecting worksheets, selecting a range of cells, or adding the current date to a spreadsheet.

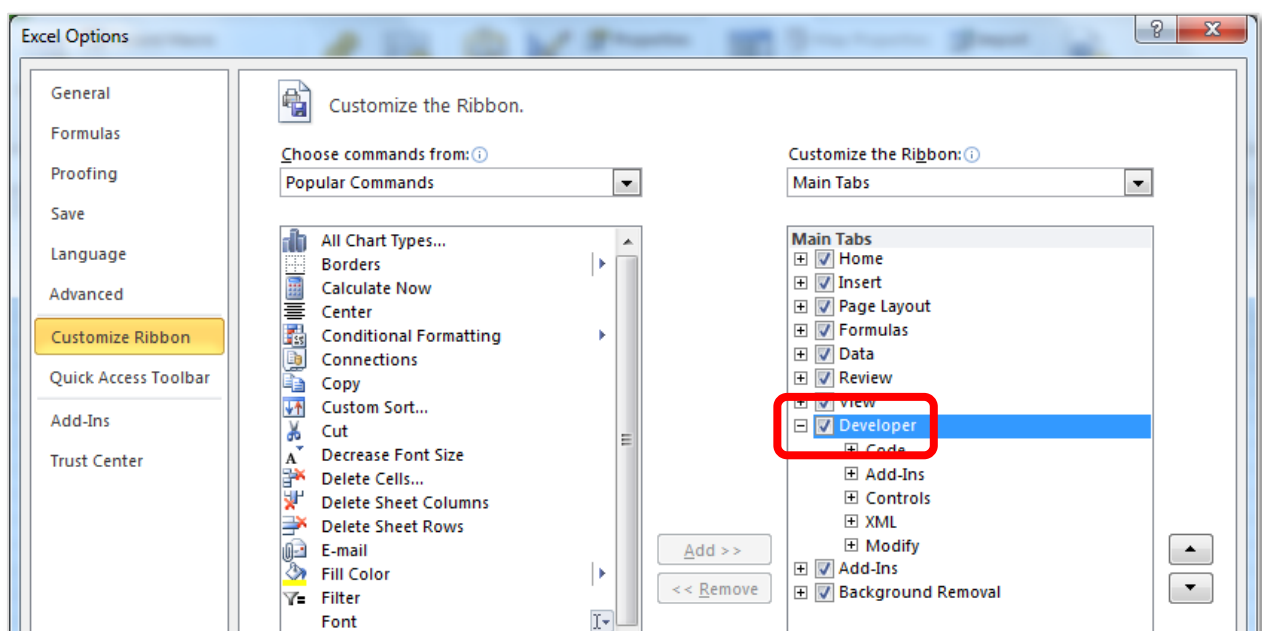
When a Macro is recorded a set of **VBA** (Visual Basic Application) code is automatically written in Microsoft Visual Basic. In the section Edit a Macro some of this code is reviewed. When saving a workbook in Excel 2007 or later is important to remember to always save the workbook as **Save As Type: Excel Macro-Enabled Workbook**.

## 2. RECORD A MACRO

To record macros use the macro recorder. To access this first turn on the **Developer** tab to access the **Code** group and set the **Security Level**.

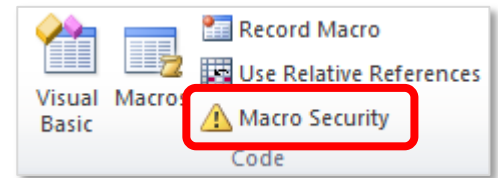
### TURN ON THE DEVELOPER TAB

To turn on the **Developer** tab go to **Excel Options**. In the **Customize Ribbon** options, turn on the **Main Tab Developer** by ticking the check box and then click **Ok**.



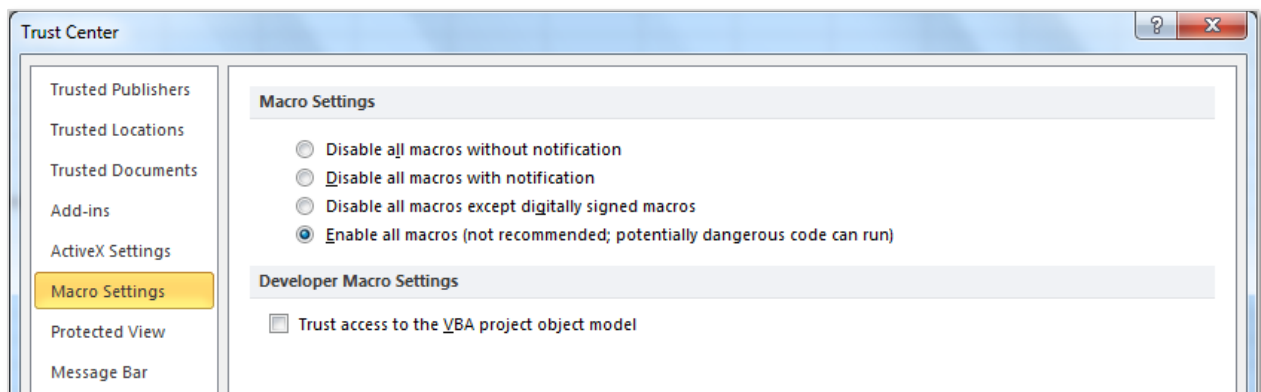
## SET THE SECURITY LEVEL

On the **Developer** tab, in the **Code** group, click on the command button **Macro Security**.



This will open the **Trust Centre** under the option **Macro Settings**, click the radio button **Enable all macros (not recommended, potentially dangerous code can run)** and then click **Ok**.

This will allow the macro code to run when the workbook containing the macro is opened. If this option is not turned on the macro code cannot run.

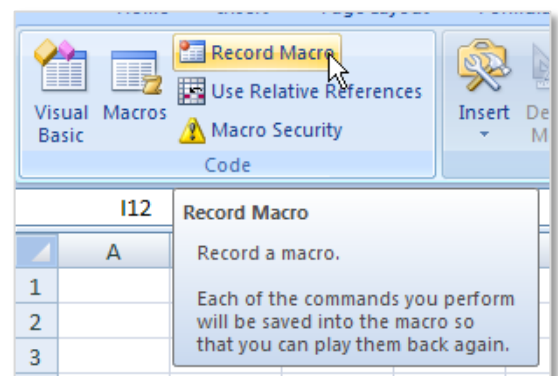


**Note:** Excel documents with macros code can potentially contain viruses, only open Excel document from a trusted source.

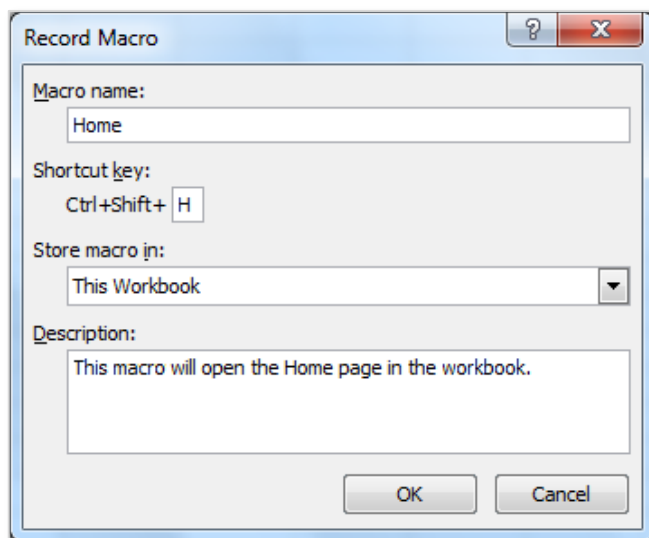
## RECORD A MACRO

To **Record a Macro** go to the **Developer** tab to the **Code** group and click on the command button **Record Macro**. This will open the **Record Macro** dialog box.

1. In the **Macro name:** text box enter a name for the macro (spaces are not allowed in the name of a macro).
2. If required set a shortcut key for the macro to run. In this example the shortcut keys chosen are **Ctrl+Shift+H**.
3. Decide which workbook the macro will be saved in by changing the **Store macro in:** option.



4. Include a description of the macro in the description box, give specific details to explain what the macro will do.
5. Click **Ok** to start recording - **REMEMBER** that anything selection or click completed from this point until the recording is stopped will become part of this macro.
6. Perform the actions the macro will do.
7. On the **Developer** tab, in the **Code** group, click on **Stop Recording** (a **Stop Recording** button is also available on the bottom left side of the status bar).



NOTES: Write in your own words how to record a macro...

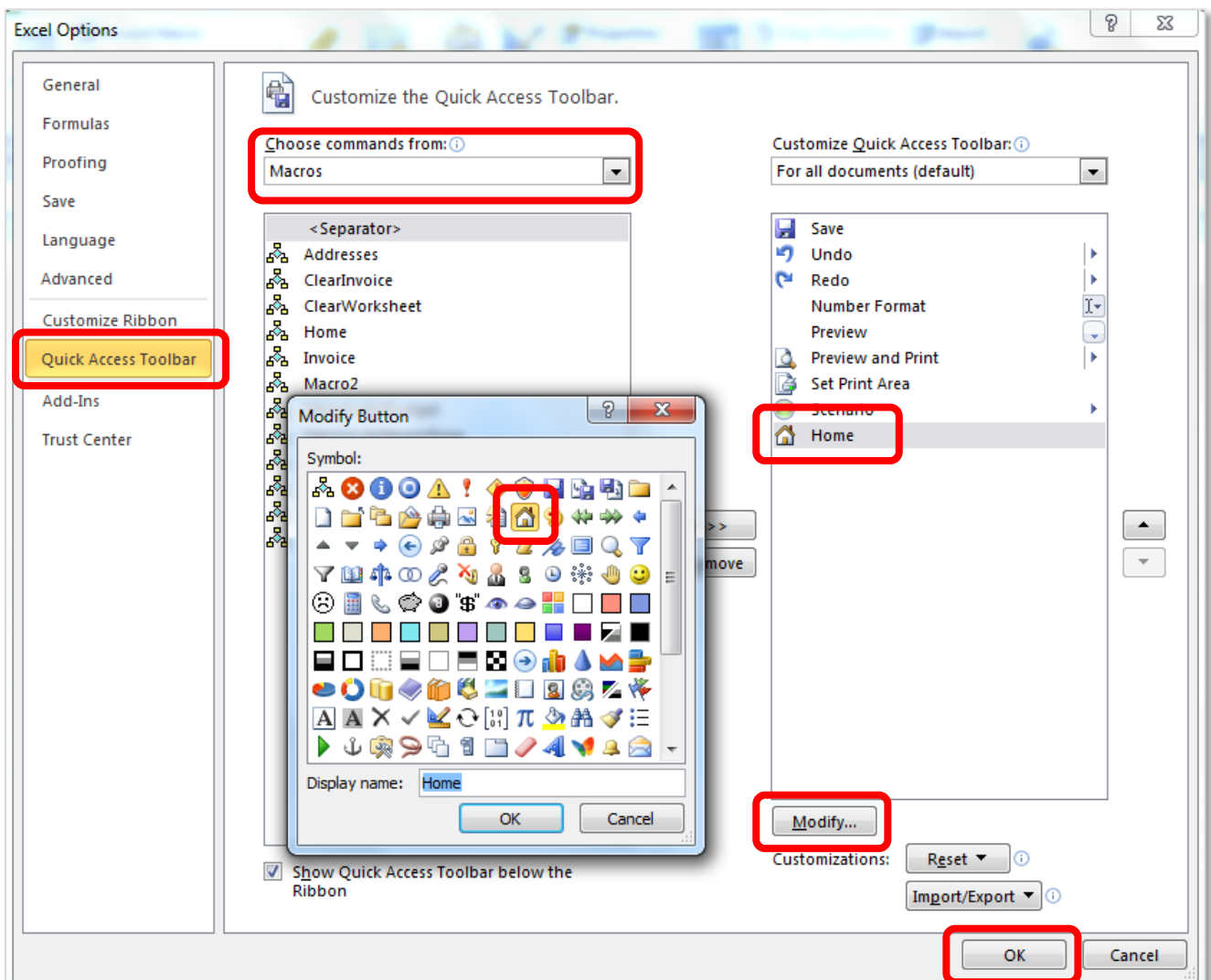
## RUN A MACRO

To **Run a Macro** there are a variety of methods to use:

**KEYBOARD SHORTCUT** - in the previous example the macro was assigned the keyboard shortcut of **Ctrl+Shift+H** pressing these keys will run this macro.

**MACRO DIALOGUE BOX** – go to the **Developer** tab to the **Code** group and click on the **Macros** command button, from the **Macro** dialogue box select the macro required and click on the **Run** button (keyboard shortcut for the macro dialogue box is **Alt+F8**).

**QUICK ACCESS TOOLBAR** – individual macros can be added to the **Quick Access Toolbar** (or the Ribbon to a New Group) by going to **Excel Options** to the **Quick Access Toolbar** and selecting **Choose commands from: Macros**. Select the required macros and **Add** it to the **Quick Access Toolbar**. The macro can be formatted to have a image by clicking on the **Modify** command button to access the **Modify Button** dialogue box, choose an image and click **Ok** and **Ok**. The macro will be added to the quick access toolbar.

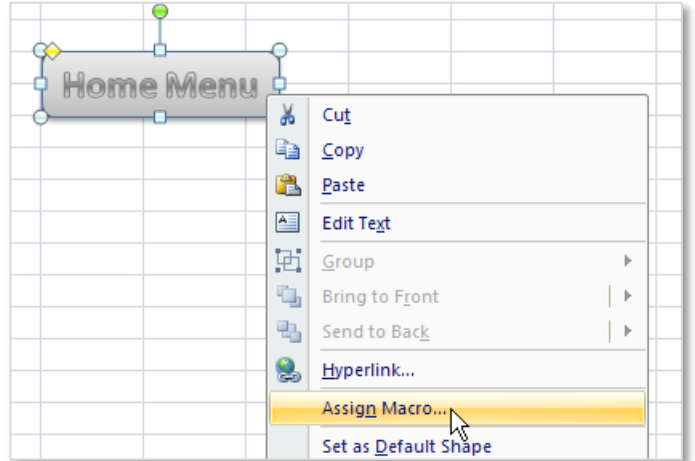


---

## ASSIGN A MACRO TO AN OBJECT OR GRAPHIC

An alternative method of running a macro to those outlined above it to assign a macro to an object or graphic. To this first create a button from a shape or image, in the following example a rounded edged rectangle was used.

1. Add an object to the worksheet – for example this may be a button to take link to a different work sheet. Create the button using **Shapes** and formatting.
2. Right click on the button or image created and click on the **Assign Macro** on the shortcut menu, this will open the **Assign Macro** dialogue box.
3. In the **Assign Macro** dialogue box, click the macro required for the button.
4. This will link the button and the macro.
5. To test the macro has been assigned click off the button onto the worksheet and click on the button, the macro will run automatically.

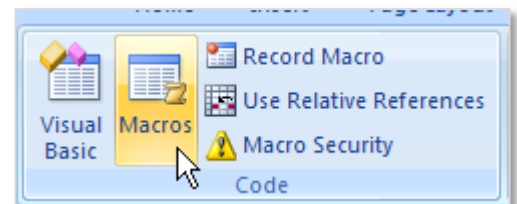


---

## DELETE A MACRO

Open the workbook that the macro is written in. On the **Developer** tab, in the **Code** group, click **Macros**.

In the **Macro** dialogue box click on the name of the macro to be deleted and click the **Delete** command button.



---

## EDIT A MACRO

To **Edit a Macro** and view the Visual Basic code that has been automatically created go to the **Developer** tab in the **Code** group click **Macros**. In the **Macro** dialogue box click the macro that to be edited. Then click on the **Edit** command button and this will open the **Microsoft Visual Basic for Application Editor** showing the **Visual Basic** code of that macro. Make the required changes to the code; **save** the changes and to exit from this select the **exit button** in the top right corner.

A macro is opened with the code **Sub** and closed with **End Sub**.

The *green code* shows the name of the macro.

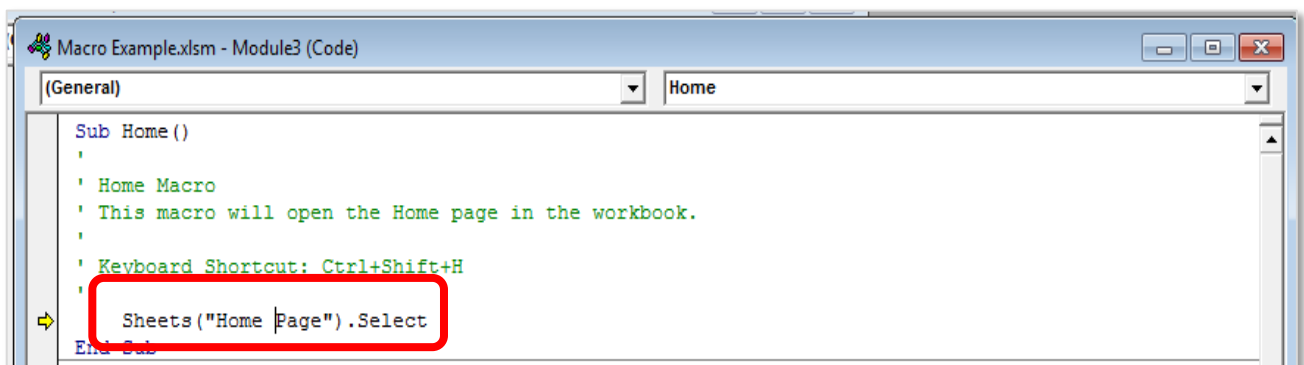
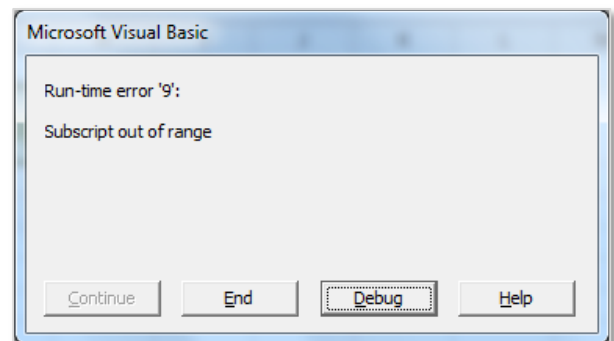
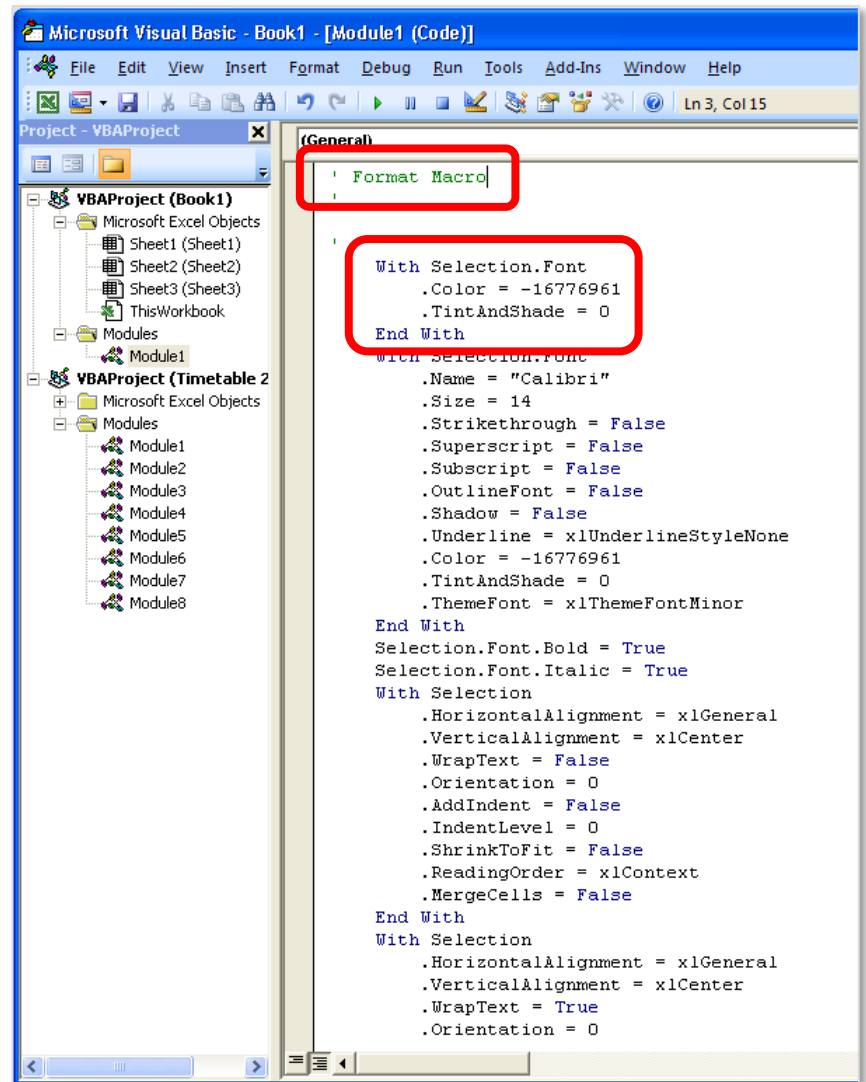
Each of the *blue With* sections of code specify actions which will be completed. These are ended with an **End With**.

To delete one particular action delete the **With** section of code as far as the **End With** option.

Every macro in one workbook is saved in a list of **Subs** under each other.

## DEBUG MACRO

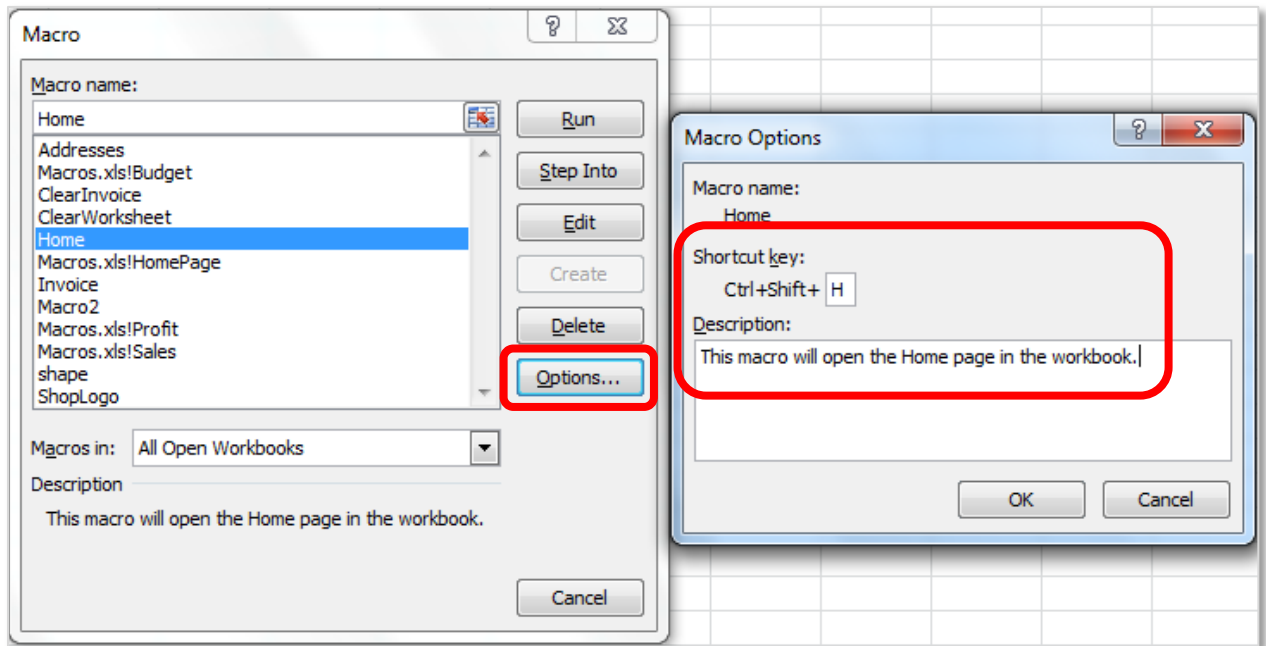
A common reason for editing a macro is when the name of the worksheet has been changed. If the worksheet has been renamed after the macro has been recorded the macro will not run. A **Microsoft Visual Basic** error message will display. To fix this problem click on **Debug** to edit the macro code change the **Sheets** to the new name (in this example it is changed to *Home Page*) **save** the changes and **exit** the editor.





## DOCUMENT A MACRO

The contents of a macro should be included in the **Description:** when the macro is being created. If this is not done it is possible to add the description in after the macro has been created. To document the contents of a macro go the **Developer** tab in the **Code** group click on **Macros**, select the required macro and click on the command button **Options**, this will open the **Macro Options** dialogue box. From here the **Shortcut key:** and **Description:** of the macro can be assigned.

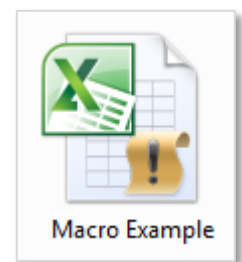


## SAVING A MACRO ENABLED WORKBOOK

When saving a workbook with any macro in it is important to remember to save it as a macro enabled workbook file type. Failure to do this will result in all of the macros that have been recorded being deleted from the workbook.

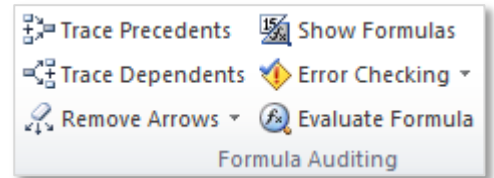
To save a workbook as **Macro Enabled** go to **File** open the **Save As** dialogue box. In the **Save as type:** text box change the file type to **Excel Macro-Enabled Workbook** (file type .xlsm).

Notice then an Excel document with the file type of **.xlsm** has a different logo than the normal **.xlsx** file. There is a yellow flag with an exclamation mark on it to warn the user that this file has macros in it.



### 3. AUDITING

Errors and design flaws are very common in spreadsheets and it is important to be aware of steps that can be taken to design more reliable and robust models. Excel includes some tools which can be used to investigate spreadsheets and clarify the processes and calculations being performed. **Formula auditing** in Excel allows the displaying of relationships between formulas and cells. To access these tools go to the **Formula** tab to the **Formula Auditing** group.



#### TRACE PRECEDENTS

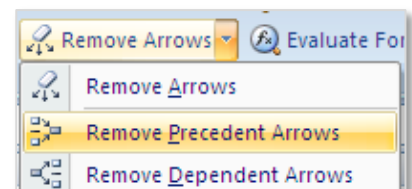
The **Trace Precedents** tool displays arrows which indicate every cell used to calculate the value of the selected cell. This can be very useful in correcting an error in a cell. This will only work if the current cell contains a formula, from that cell arrows are shown drawn leading back to the source cells.

To access **Trace Precedents** go to the **Formula** tab and go to the **Formula Auditing** group.

	A	B	C	D	E	F	G	
	Code	Furniture Item	Quantity Bought	Cost Price per Item	Total Cost Price per Item	Selling Price per Item	Total Selling Price per Item	Pro Each
1								
2	HG7654	Tables	60	€95.00	€5,700.00	€120.00	€7,200.00	
3	KL9876	Chairs	85	€88.00	€7,480.00	€130.00	€11,050.00	
4	LP9876	Cabinets	95	€56.00	#VALUE!	€80.00	€7,600.00	
5	YT9876	Dressers	52	€79.00	€4,108.00	€90.00	€4,680.00	
6	JT7654	Hall Tables	73	€51.00	€3,723.00	€77.00	€5,621.00	
7	KJ7654	Telephone Tables	55	€63.00	€3,465.00	€80.00	€4,400.00	€17.00

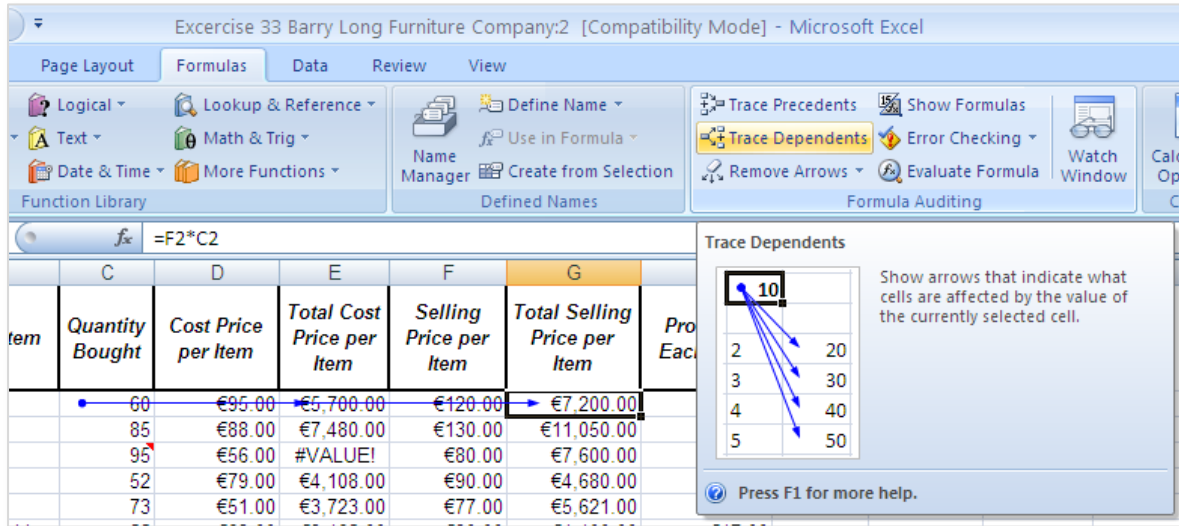
In this example the formula has an incorrect cell reference, the user is attempting to multiply a cell with text by a cell with a number in it, and this formula will not work. The trace lines can be used to follow the cells which the formula has been created from.

To turn off the arrows click on the **Remove Arrows** button.



## TRACE DEPENDENTS

**Trace Dependents** works in opposite to tracing precedents, in tracing dependents the arrows show the cells which depend selected cell (showing the cells which contain formula affected by the selected cell). This will only work if the selected cell is used in a formula.

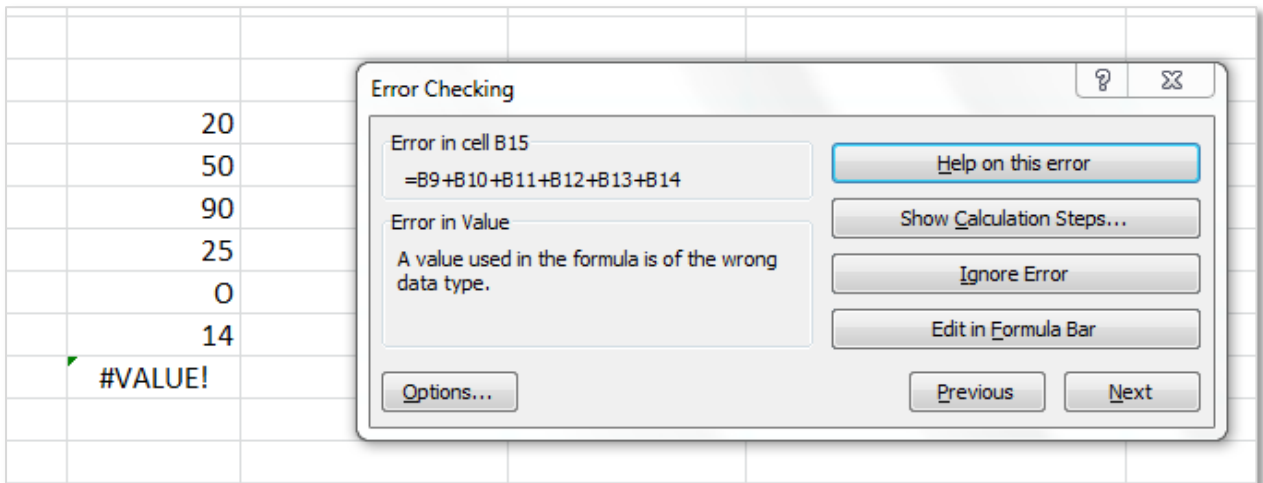


## COMMON ERROR CODES IN EXCEL

Error Code	Explanation
#####	The column isn't wide enough to display the value.
#DIV/0!	When a formula tries to divide a number by 0 or by an empty cell.
#N/A	A value is not available to a function or formula, for example missing data in a LOOKUP.
#NAME?	Excel does not recognise text in the formula, for example the name of the formula used is spelt incorrectly.
#NULL!	This error means that an intersection of two areas has been specified but the areas do not intersect.
#NUM!	This error shows that a formula or function contains invalid numeric values, for example a text data type might have been given for a function that requires a numeric argument.
#REF!	The formula relates to a cell that is not valid, for example adding a cell to its own cell reference creating a circular reference.
#VALUE!	A formula has the wrong type of argument, for example a word and a number have been added together.

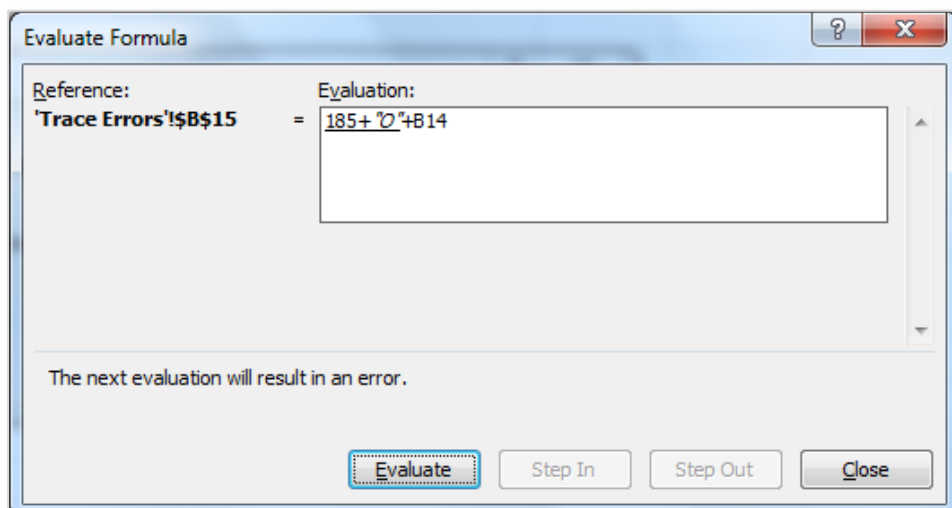
## TRACE ERRORS

Like a spelling checker that checks for errors in text, **Error Checking** allows the checking of common errors that occur in formula. These rules do not guarantee that the workbook is error free but they find common mistakes. To access the Error Checking dialogue box go to the **Formula** tab to the **Formula Auditing** group and select the **Error Checking** command button. If there are any errors in the worksheet they will be automatically detected.



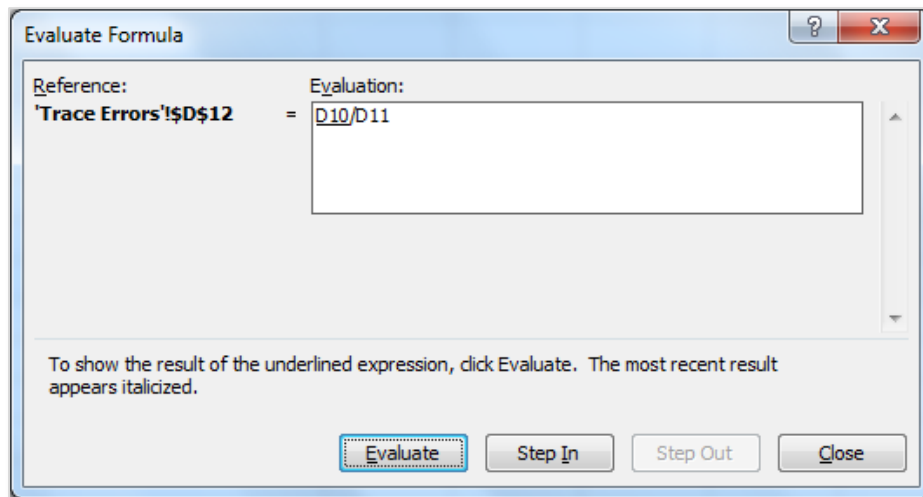
To access **Microsoft Help** with advice on how to correct the error select the **Help on this error** command button.

To see where in the calculation steps, the error has occurred select the **Show Calculation Steps...** command button, this will open the formula exactly at the point in the calculation where the problem is happening. In this example the cell B14 holds a letter O instead of the number 0. The **Evaluate Formula** dialogue box has opened exactly at this point to emphasize where the error is happening.

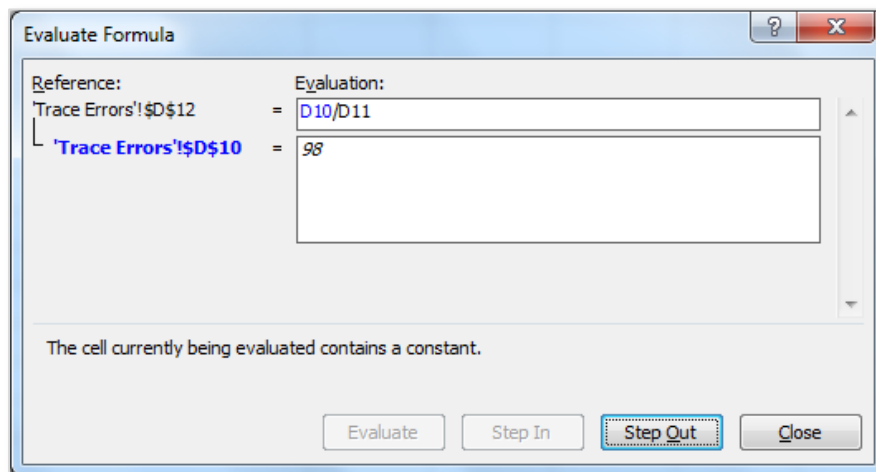


## EVALUATE FORMULA

The **Evaluate Formula** auditing tool can be used to follow a formula through step by step in order to understand how the formula is working or to see where a calculation error has been created. To use this tool first click on the cell that contains the formula to be evaluated, then go the **Formula** tab to the **Formula Auditing** group and select the **Evaluate Formula** command button. This will open the **Evaluate Formula** dialogue box.



To investigate the formula in detail use the **Step In** and **Step Out** buttons and the formula can be followed one step at a time as the computer would calculate it. The calculation steps of the formula are shown one at a time allowing the user to review each stage of the calculation individually.



## SHOW FORMULAS

The last command button available in the Formula Auditing group is **Show Formulas** this button turns any formula in the worksheet on to display the actual formula and not the resulting data. This button makes it easy to find any cells containing formula and allows the user to quickly read the formula. To turn this option off click on the **Show Formulas** button again.

## 4. REFERENCES

### WEBSITES:

Auditing – for more information visit [Easy-Excel.com](http://Easy-Excel.com).

Formula Errors – for more information visit [Easy-Excel.com](http://Easy-Excel.com).

Macros – for more information visit [Spreadsheets.About.Com](http://Spreadsheets.About.Com).



### NOTES: