

# Excel macros made easy



Jane Barrett, IT Training & Engagement Team  
Information System Services

Version 1.1



# Scope

## Learning outcomes

- Understand the concept of what a macro is and what it does.
- Demonstrate how to record a macro.
- Understand how macros can be assigned within workbooks.

# Table of Contents

<b>Scope</b> .....	<b>2</b>
<b>Table of Contents</b> .....	<b>2</b>
<b>Introduction</b> .....	<b>3</b>
<b>Macros or Visual Basic for Applications (VBA)?</b> .....	<b>3</b>
What are macros and VBA? .....	3
Writing your first macro .....	3
Write a simple macro .....	3
Practice writing some macros .....	5
<b>Choosing where to store macros</b> .....	<b>5</b>
The Workbook .....	5
Personal Macro Workbook .....	5
New Workbook .....	5
<b>Choosing macro shortcut keys</b> .....	<b>6</b>
<b>Recording a macro with absolute references</b> .....	<b>6</b>
Write a macro with an absolute reference .....	6
<b>Understanding macro security</b> .....	<b>7</b>
Three ways to trust a macro enabled workbook .....	7
<b>Saving macros</b> .....	<b>8</b>
Saving your macros .....	8
<b>Implementing macro security</b> .....	<b>8</b>
Setting the Trust Center's macro settings .....	9
Set a trusted folder for your macros .....	9
<b>Recording macros with relative references</b> .....	<b>10</b>
Record a macro with relative references .....	11
<b>Using shapes to run macros</b> .....	<b>12</b>
Assign a macro to a button.....	12
Add a shape to your worksheet.....	13
Connecting the macro to the shape .....	14
Connecting a hyperlink to a shape .....	14
<b>VBA (the code behind the macros)</b> .....	<b>15</b>
Record a macro and watch the VBA appear .....	16
<b>Contacts</b> .....	<b>17</b>
For queries or help .....	17
<b>Useful resources</b> .....	<b>17</b>
Training videos .....	17
Booking other courses.....	17
<b>Your feedback</b> .....	<b>17</b>
Help us improve your training.....	17

# Introduction

Have you ever spent too much time on a repetitive task in an Excel worksheet trying to reorganise or format data?

If this sounds familiar, then you need to know about **macros**. This course will show you how they can be used to turn repetitive tasks into single clicks. You will learn how to record, edit and assign your macros to workbooks, or worksheets, and start to explore what VBA, (Visual Basic for Applications) has to offer.

## Macros or Visual Basic for Applications (VBA)?

### What are macros and VBA?

Generally there is a lot of confusion when the words **macro** and **VBA** are mentioned, often in the same sentence. A macro simply records mouse clicks as you go about a task and then converts the actions into a VBA programming code. The advantage of this is that you can record a repetitive task once using a macro and then play all the steps back again using a single mouse click.

To use macros you just need to understand the task you wish to record. If you want to look at the code and alter it, you can, **but you don't have to**, the macro does all the hard work for you. Macros can give you a good insight into how VBA works, so if you would like to learn to write it at a later date, it will give you a good starting point.

VBA is not just used in Excel, but is the building block for all the main office applications.

### Writing your first macro

The macro recorder is able to record and playback every mouse click or keystroke you make. The process works in three distinct phases.

1. Assign a name and shortcut key for your new macro, then start the macro recorder to record.
2. Perform the series of mouse clicks and keystrokes you wish to record (if it's a long sequence it is a good idea to plan this phase before you start to avoid errors).
3. Stop the recorder - don't forget to do this otherwise you will end up with a very long macro!

Once you have completed these steps you can then play your macro using the shortcut key or review the VBA code written by the macro.

## Task 1

### Write a simple macro

1. Open a new Excel workbook.
2. Select **View** → **Macros** → **Record macro**.

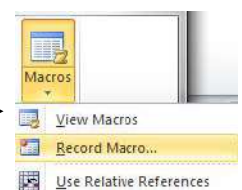


Figure 1

- The macro dialogue box appears, type **MyName** into the macro name box. (macro names must begin with a letter, no spaces are allowed).
- Type a lower case **n** into the Ctrl+ box.
- Select **This Workbook** in the store macro drop down list (the other options will be explained later).
- Write a short description of what your macro does in the Description box.
- Click the **OK** button then **STOP** and do nothing else!

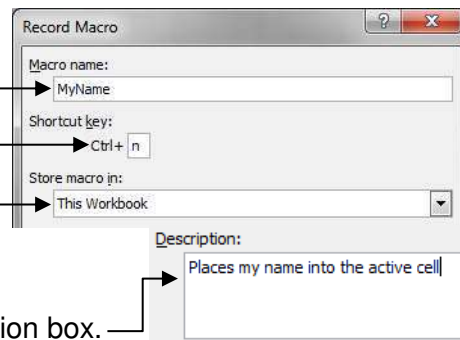


Figure 2

You are now recording your macro. There's no time element to a macro recording so you can be relaxed. However, the macro will now record every key press and mouse click, so it is important that you do not press any keys or click anywhere with the mouse.

### Record a macro that will put your name into the active cell

- Type your **name** into the currently active cell, remember that every click and key stroke is recorded so be careful not to move to any other cell. If you did your name would always appear in the cell you moved to when the macro was played.
- Press **Ctrl + Enter**, this saves the value to the cell without moving to the next line. This keystroke combination is really useful when recording macros.

### Stop the macro recording

You can do this in two ways:

- click, View → Macros → Stop Recording or
- click the **Stop button** on the bottom left of the status bar

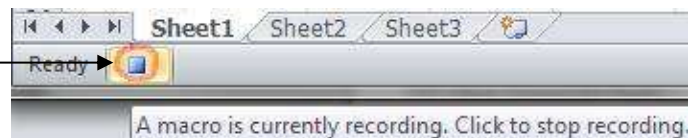


Figure 3

It is quite common for people to forget they are recording a macro and remember to stop it.

### Test the macro

- Click in any blank cell.
- Click **View → Macros → View Macros**. The macro dialogue appears listing only the **MyName** macro, as you have only recorded one macro so far.
- Click the **Run** button. Your name should appear in the currently active cell.
- Click in another blank cell and press **Ctrl + n**, once again your name appears in the active cell.

## End task

## Task 2

### Practice writing some macros

Using the same work book, create macros to:

1. format text to font **Arial Black**, size **16** and **centered**
2. format a number to show **5** decimal places
3. write your name in cell **C3**
4. change the sheet colour to **yellow** and all text to **red**
5. save the workbook as **Macros 1** - as the workbook contains macros you must select the **Save as type** to **Macro enabled workbook** otherwise you will lose your macros.

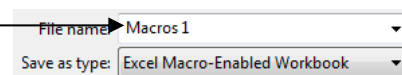


Figure 4

## Task End

## Choosing where to store macros

When you record a macro, Excel offers you three possible places to store it.

Each one determines which workbooks your macro will be available to and who can access it. Therefore it is really important that you are aware of the differences in the choices on offer.

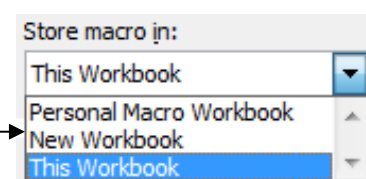


Figure 5

## The Workbook

If you select **This Workbook** as an option to store your macro then the macro you create will be available to other workbooks as well, but only if the workbook containing the macro is open.

## Personal Macro Workbook

The **Personal Macro Workbook** is a special workbook called **Personal.xlsb**. This is stored in a folder called XLSTART buried deep in the file system where nobody is likely to find it and change it. Whenever you open Excel, the Personal Macro Workbook opens in the background, but it is hidden so you never see it. As it is always open every workbook has access to its macros at all times.

The main purpose of the Personal Macro Workbook is to act as a container to store macros that you want to be available to every workbook. However, if you store macros in the Personal Macro Workbook they will only work on your machine. If you email a piece of work that depends on one of these macros, it will not work on the recipient's machine. In this type of scenario you will want to store the macro in **This Workbook**.

## New Workbook

The **New Workbook** option is rarely used as it opens a new Excel file and the macro is only available in this new file, hence you would always have to make sure that this file was open if you wanted to access the macro.

## Choosing macro shortcut keys

Excel provides many built in shortcut keys such as **Ctrl + c** to copy or **Ctrl + v** to paste. When you choose a shortcut key combination to run your macro, if you select a combination that matches a built in shortcut then your macro would override the built in shortcut; eg **Ctrl + c** would run your macro instead of copying the selection.

There are several strategies to avoid causing clashes with the built in shortcut keys:

1. use capital letters for shortcut keys - this will result in a shortcut key of **Ctrl + Shift + c** which means that you will still be able to copy using the **Ctrl + c** keystroke
2. run macros from the ribbon, as you did for your first macro,  
**View → Macros → View Macros**
3. provide the user with buttons to run macros, something we will cover later, together with...
4. adding a button to the quick access toolbar

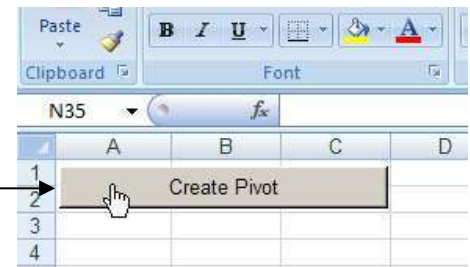


Figure 6

## Recording a macro with absolute references

Imagine this is a form you need to fill in every week. Every time you make an expenses claim, you have to add your employee number and other details.

This isn't very efficient so you decide to create a macro to do the job for you and complete the form with the required information.

This is an example of using absolute cell references as the information must be put in cells B3, B5, B7, B9 and B11.

A screenshot of an Excel spreadsheet showing a form titled 'Expenses Claim'. The form is located in columns A and B, rows 1 through 12. The fields are: Name (B3), Department (B5), Employee Number (B7), Bank Account No (B9), and Sort Code (B11). The cells B3, B5, B7, B9, and B11 are highlighted in yellow. An arrow points from the text 'information must be put in cells B3, B5, B7, B9 and B11' to these cells.

Figure 7

## Task 3

### Write a macro with an absolute reference

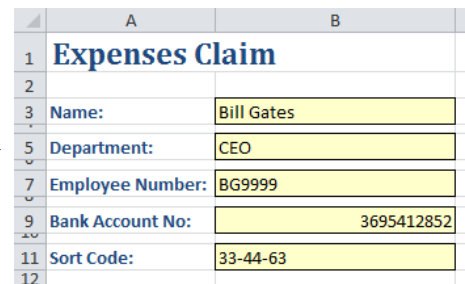
Open the Excel workbook, **Expenses Claim – 1** from your m drive.

1. Begin recording a macro **View → Macros → Record macro**.
2. Name the macro **FillInExpenseForm** into the macro name box, always use mixed case or underscore for the macro names rather than abbreviating as this can cause confusion.
3. Assign a shortcut key of **Ctrl + e** for the macro.
4. The macro only relates to this form and will only be used in this workbook, so choose the **This Workbook** option for storing the macro.
5. Add a **description** to the macro so that other users will understand what it is used for and click **OK**.
6. Record the macro, start by making B3 the active cell.

This is a potential pitfall. If B3 is already the active cell when you begin recording the macro and you simply type your name the recorder will place your name into whatever cell is the active cell when the macro is run. This may not be B3.

For this reason you should click into a different cell and then back again into cell B3 to explicitly record your intention to move the cursor to B3.

7. Fill in each box with fictitious details and remember to press the **Ctrl + Enter** key after entering the last cell or you'll stay in edit mode and won't be able to stop the macro recorder.
8. Click **View** → **Macros** → **Stop Recording**.
9. Now test your macro by **running** it from the **ribbon** or using the **shortcut key**.



	A	B
1	<b>Expenses Claim</b>	
2		
3	Name:	Bill Gates
4		
5	Department:	CEO
6		
7	Employee Number:	BG9999
8		
9	Bank Account No:	3695412852
10		
11	Sort Code:	33-44-63
12		

Figure 8

## Understanding macro security

Macros are written using VBA which has access to most Windows system calls and executes automatically when documents are opened. This makes it relatively easy to write a macro which could be hidden within a Microsoft Office application, such as Word or Excel, which executes a virus when the document or workbook is opened. This became one of the most common types of computer viruses and as a result, applications running macros or VBA, now have improved features to make sure you are aware if macros are present and allows you to choose to run them or not.

So whilst macros are a wonderful Excel feature they can also be very dangerous to your computer in many ways. Macro security provides several methods to identify which macros you can trust and which you can't. Excel 2007 took a huge leap forwards in the battle against macro viruses. There used to be only one Excel format, the .xls file, now there are two:

1. **Excel Workbook** with a **.xlsx** file extension. This can always be trusted because it is incapable of storing a macro.
2. **Excel Macro Enabled Workbook** with a **.xlsm** file extension. You must ensure that you can trust the origin of this type of workbook as it is capable of carrying destructive macro viruses.

## Three ways to trust a macro enabled workbook

Excel provides three mechanisms for recognising a trusted workbook.

1. Place the workbook in a **trusted folder**. Excel allows you to designate a folder as a trusted location. You are then able to open macro enabled workbooks that are saved to this location without warnings or restrictions.

This is probably the easiest and best way to manage macro security.

2. **Digitally sign the macro** to confirm the identity of the author. Excel will also warn the recipient if it has been changed since the author signed it.

Excel can be configured to run digitally signed macros from trusted publishes without warnings or restrictions.

3. **Warn the user before running a macro**. Excel warns you that a workbook contains macros and asks whether you wish to run them.

# Saving macros

## Task 4

### Saving your macros

1. **Expenses Claim – 1** should still be open, try saving it as a regular Excel Workbook and you will get the message below, which warns you that you cannot save it as a macro free workbook.

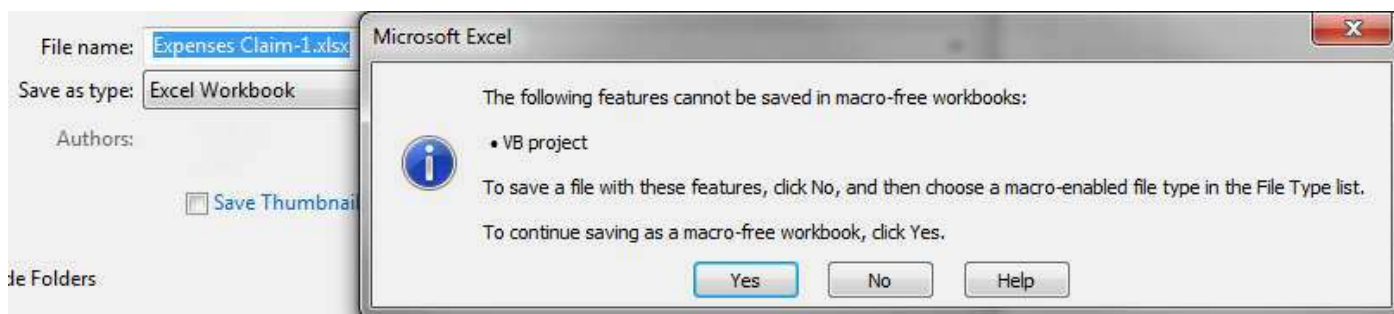


Figure 9

2. Click the **No** button to close the dialog.
3. This time change the *Save as type* to **Excel Macro Enabled Workbook** (.xlsm).
4. Change the file name to **Expenses Claim – 2** and click **Save**.
5. **Close** the file.

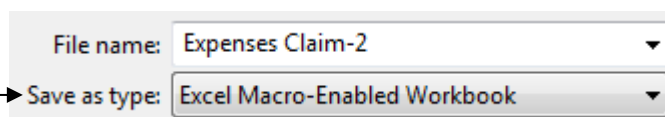


Figure 10

## Task end

# Implementing macro security

When you open a macro enabled workbook, one of three things will happen.

1. The workbook opens without warnings and macros are enabled.
2. Excel displays a security warning, (this is the default setting).

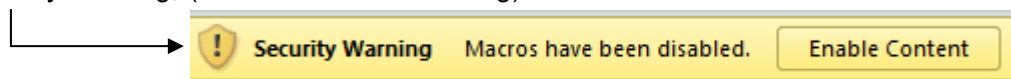


Figure 11

3. The workbook opens without warning, but the macros are disabled.

It doesn't matter which is the case on your computer at the moment, the following task will teach you how to set up the macro security options to any of the above.



---

## Task 5

### Setting the Trust Center's macro settings

The Trust Center allows you to set the options of how you wish do deal with macros.

1. Click **Office Button** → **Excel Options** → **Trust Center**.
2. Click the **Trust Center settings** button.
3. Click **Macro Settings** in the left hand selection bar, this will display the macro settings.

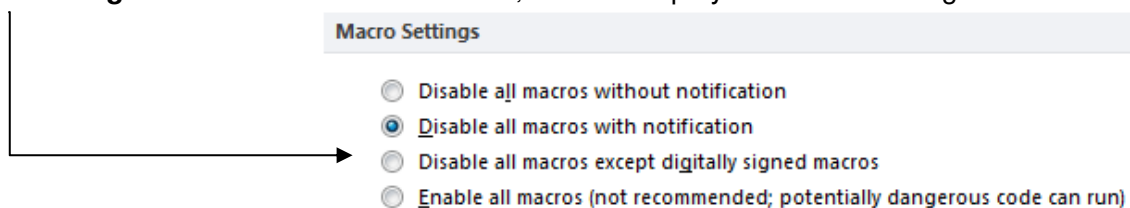


Figure 12

4. This is where you can choose how macros are opened, the best option is probably the one already selected, **Disable all macros with notification**.
5. Click **OK** and **OK** again.
6. Now open the **Expenses Claim – 2** workbook and the warning should be displayed.
7. Click on the **Enable Content** and a dialogue box will be displayed asking if you wish to make this a Trusted Document. Click **Yes** and now you will be able to run your macro within the workbook.



Figure 13

## Task end

---

## Task 6

### Set a trusted folder for your macros

This is probably the best method of all when dealing with macro security. If you designate a folder for your trusted macro-enabled workbooks, Excel will know that they are trusted and not bother you with any warnings.

For extra security, the trusted folder should always be a non-shared folder on your local hard drive ie (c:/), not a network drive.

1. **Create a folder** for your trusted macro enabled workbooks, eg **c:/Trusted Files**.
2. Click **Office Button** → **Excel Options** → **Trust Center**.
3. Click the **Trust Center settings** button.
4. Click the **Trusted Locations** on the left-hand selection bar.

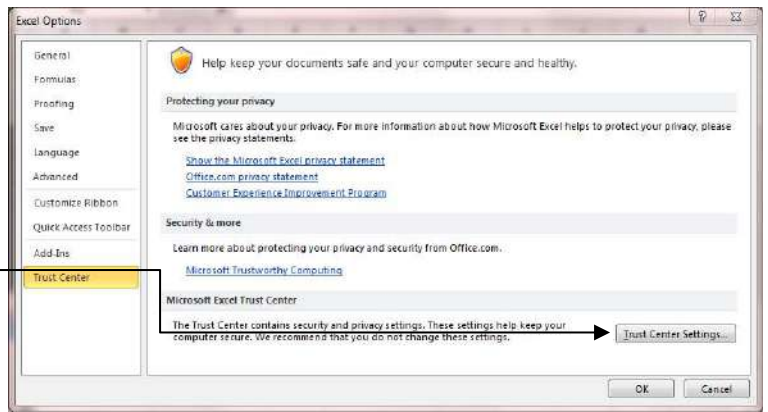


Figure 14

5. Click **Add New Location** button.
6. Click the **Browse...** button
7. **Navigate to the folder** that your trusted files will be placed in.
8. Click the **OK** button
9. Click **OK** and **OK** again to close the dialogues.

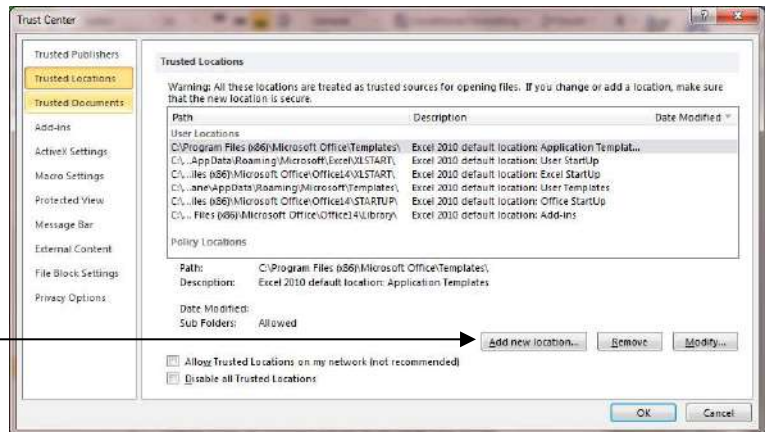


Figure 15

**Test to see if the Trusted Folder works.** Place the **Expenses Claim – 2** workbook into the folder, now it should open without warnings and the macro should still run.

**Task end**

## Recording macros with relative references

The last example of an expense claim form was a useful example of a macro involving absolute cell references, ie the cell references were always the same locations.

However sometimes we need to be able to write a macro that uses relative cell references to perform the same task at different places on the worksheet or workbook.

For example the workbook on the right contains 88 companies, but it is in the wrong format for a mail merge. We need the format to be as below.

	A	B	C	D	E	F
1	<b>Company List</b>					
2						
3	<b>Contact Name</b>	<b>Company Name</b>	<b>Address</b>	<b>City</b>	<b>Post Code</b>	<b>Country</b>
4	Alejandra Camino	Romero y tomillo	Gran Vía, 1	Madrid	28001	Spain
5	Alexander Feuer	Morgenstern Gesundkost	Heerstr. 22	Leipzig	4179	Germany
6	Ana Trujillo	Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	México D.F.	5021	Mexico

	A
1	<b>Company List</b>
2	
3	<b>Contact Name</b>
4	Alejandra Camino
5	Romero y tomillo
6	Gran Vía, 1
7	Madrid
8	28001
9	Spain
10	
11	
12	
13	Alexander Feuer
14	Morgenstern Gesundkost
15	Heerstr. 22
16	Leipzig
17	4179
18	Germany

Figure 16

This would take a long time to manually convert the records but a macro can do it in less than 5 seconds using relative references.

Relative reference macros record an offset from the active cell. Imagine that the active cell is A4 (Alejandra Camino). The relative macro will record the actions:

- move one cell down
- copy
- move one cell up and one to the right
- paste

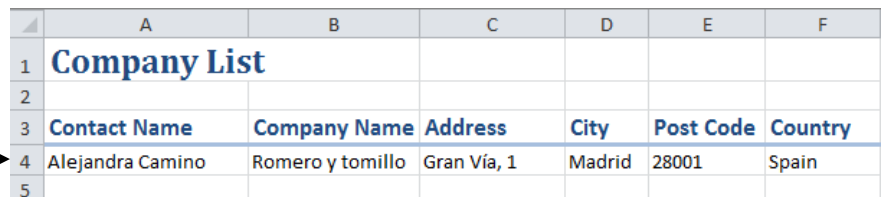
...and so on until all of Alejandra's details are moved to row 4. The macro will then delete the blank rows and be ready to start again with the next person, Alexander Feuer.

## Task 7

### Record a macro with relative references

Open the workbook, **Company List**. First the macro recorder has to be told to use relative references. To do this follow the steps below.

1. Click **View** → **Macros** → **Use Relative References**.
2. Make cell **A4** the **active cell**. When recording a relative reference macro, it is vital that you select the active cell **before** beginning the recording.
3. Click **View** → **Macros** → **Record macro**.
4. Call your macro **MoveAddress** with a shortcut of **m** (lower case).
5. Store the macro in **This Workbook** and click **OK** to begin recording the macro.
6. Click in cell **A5** and **Cut**.
7. Click in cell **B4** and **Paste**.
8. **Continue** until the complete address is pasted into row 4.



	A	B	C	D	E	F
1	<b>Company List</b>					
2						
3	<b>Contact Name</b>	<b>Company Name</b>	<b>Address</b>	<b>City</b>	<b>Post Code</b>	<b>Country</b>
4	Alejandra Camino	Romero y tomillo	Gran Vía, 1	Madrid	28001	Spain
5						

Figure 17

9. Select rows 5:12.
10. **Right click** inside the selected cells and **delete**.
11. Click in **A5** to make it the **active cell**. By placing the cursor in A5 before stopping recording it will be in the right place to convert the next address when you run it many times.
12. Click **View** → **Macros** → **Stop Recording**.

Now you can convert the other 87 addresses by running the macro multiple times. Make **A5** the active cell then press **Ctrl + m** and let the keyboard automatically repeat. The next 87 addresses should be converted very quickly and soon looks like this:

	A	B	C	D	E	F
1	<b>Company List</b>					
2						
3	<b>Contact Name</b>	<b>Company Name</b>	<b>Address</b>	<b>City</b>	<b>Post Code</b>	<b>Country</b>
4	Alejandra Camino	Romero y tomillo	Gran Vía, 1	Madrid	28001	Spain
5	Alexander Feuer	Morgenstern Gesundkost	Heerstr. 22	Leipzig	4179	Germany
6	Ana Trujillo	Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	México D.F.	5021	Mexico
7	Anabela Domingues	Tradiç?o Hipermercados	Av. Inês de Castro, 414	S?o Paulo	05634-030	Brazil
8	André Fonseca	Gourmet Lanchonetes	Av. Brasil, 442	Campinas	04876-786	Brazil
9	Ann Devon	Eastern Connection	35 King George	London	WX3 6FW	UK
10	Annette Roulet	La maison d'Asie	1 rue Alsace-Lorraine	Toulouse	31000	France
11	Antonio Moreno	Antonio Moreno Taquería	Mataderos 2312	México D.F.	5023	Mexico
12	Aria Cruz	Familia Arquibaldo	Rua Orós, 92	S?o Paulo	05442-030	Brazil
13	Art Braunschweiger	Split Rail Beer & Ale	P.O. Box 555	Lander	82520	USA
14	Bernardo Batista	Que Delícia	Rua da Panificadora, 12	Rio de Janeiro	02389-673	Brazil
15	Carine Schmitt	France restauration	54, rue Royale	Nantes	44000	France

Figure 18

Save your work as a macro-enabled workbook called **Company List – 2**.

**Task end**

## Using shapes to run macros

Running macros using the command keys you set up, or from the View menu are fine when it is your workbook. However sometimes you may like to automate a workbook for a user and would like to make it as simple as possible to do things such as print, perform a calculation or any other task you have assigned to a macro.

The easiest way to do this is by creating a button and assigning a macro to that button and then all your user has to do is click on that button to make the macro work.

## Task 8

### Assign a macro to a button

1. Open the Excel workbook **Mortgage Calculator**.

Excel can be easily modified to make interactive sheets that don't even look like a spreadsheet. Using this sheet we are going to record two macros to save and print as well as add a hyperlink to the buttons.

2. Click **View** → **Macros** → **Record macro**.
3. Type **PrintWorkbook** in the macro name box.
4. **Don't** assign a short cut and save the macro in **This Workbook**.
5. **Write** a short description of what the macro does in the **description box**.
6. Click the **OK** button.
7. Click the **Quick Print** button which will print out on the default printer.
8. Click **View** → **Macros** → **Stop Recording**.

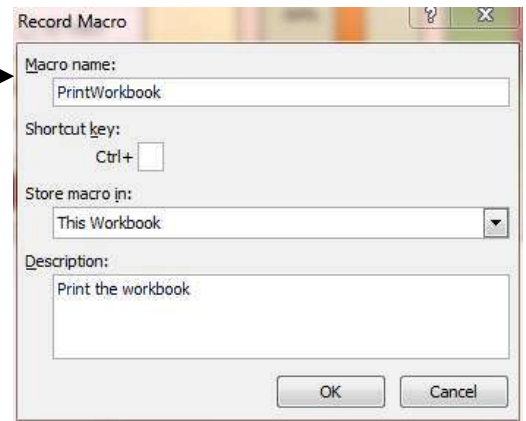


Figure 19

Now you have your first macro of the workbook recorded, save your work as a macro enabled workbook called **Mortgage Calculator 2**. This means if you mess up the second macro it is easy to go back to the previous version without having to start again from the beginning.

1. Now record a macro called **SaveWorkbook**, completing the macro dialogue box as before except for saving the workbook.
2. Click **OK** to start recording.
3. Click **File** → **Save**.
4. Click **View** → **Macros** → **Stop Recording**.

Now you have a macro that will save the workbook at a touch of a button, we just now need to add the buttons now!

## End task

## Task 9

### Add a shape to your worksheet

Although we are going to add shapes to the worksheet the same technique could be used with pictures or clipart.

1. **Resize** the **chart** to make space for a row of buttons below it.
2. Click **Insert** → **Illustrations** → **Shapes** → **Rectangles** → **Rounded Rectangle**.
3. **Click and drag** on the worksheet to draw the button.

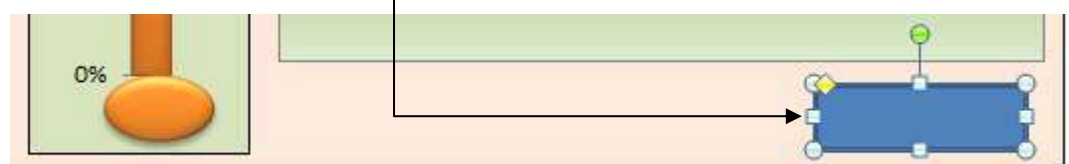


Figure 20

4. Click **Drawing Tools** → **Format** → **Shape Styles** → **Shape Style Gallery** → **Blue - Accent 1**.
5. **Copy** and **Paste** the shape **twice** and space them out underneath the chart.
6. Right click on the far right button and select **Edit Text**.

- Type **Save Quote** to add a caption to the shape.
- Add **Print Quote** and **Go to Website** to the other two buttons.

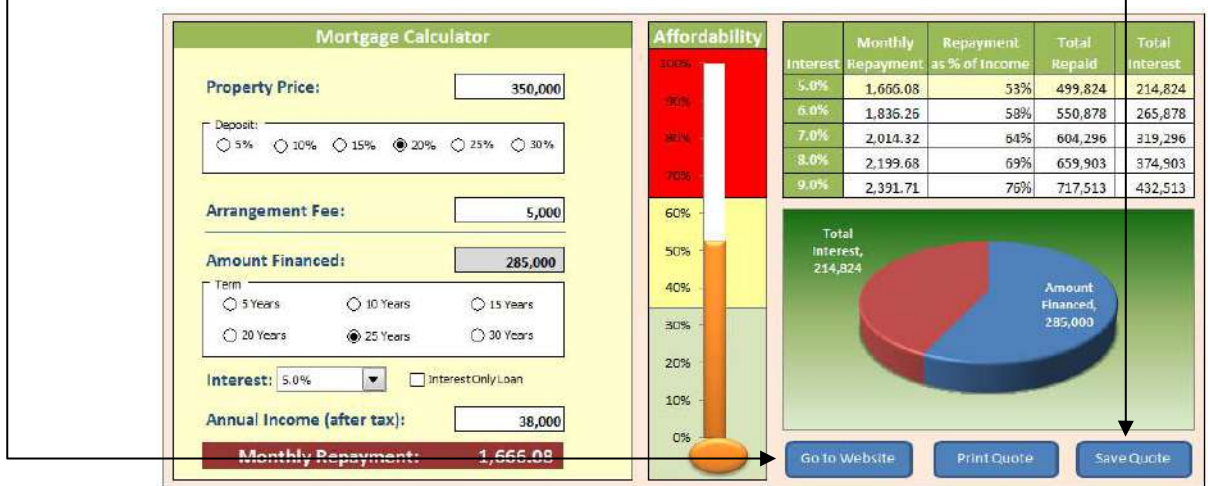


Figure 21

### Connecting the macro to the shape

- Right click the **Save Quote** shape.
- Click **Assign Macro** on the shortcut menu.
- Select the **SaveWorkbook** macro.
- Click **OK**.
- Now follow the same procedure to assign the **PrintWorkbook** macro to the **Print Quote** shape.

### Connecting a hyperlink to a shape

- Right click the **Go to Website** shape.
- Click **Hyperlink** on the shortcut menu.
- Click **Existing File or Web Page** on the left hand selection bar.
- Type <http://www.essex.ac.uk> into the address box.
- Click **OK**.

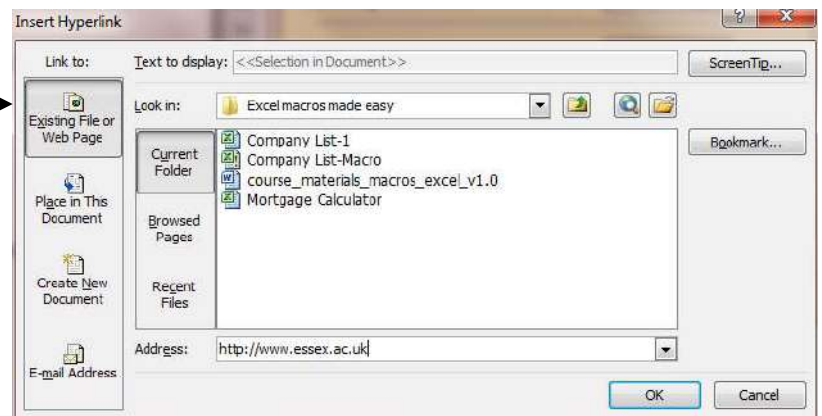


Figure 22

Now you can protect the worksheet and save it. Then reopen the workbook and test the buttons, (remember to enable the macros when you reopen in).

**Task end**

# VBA (the code behind the macros)

When you record a macro the sequence of mouse clicks are automatically converted into VBA code. If you open the Visual Basic Editor you can actually watch the code being written as you click the mouse.

To add the developer tab to the ribbon click **File** → **Options** → **Customize Ribbon** → **Choose commands from Macro** → **Developer** → **OK**.

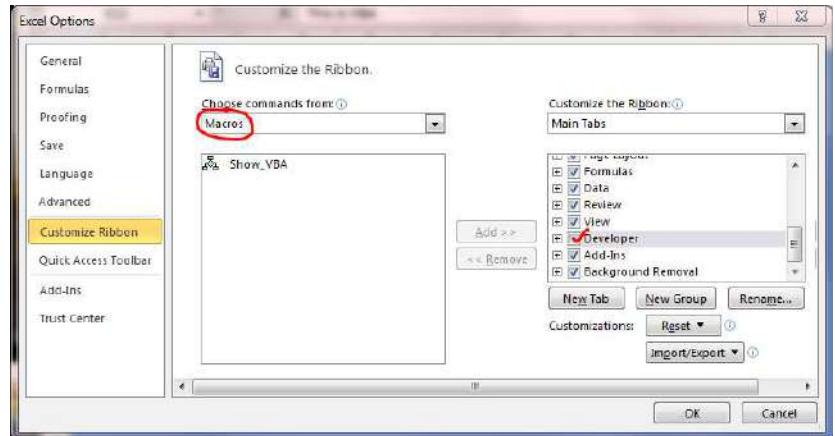


Figure 23

Now the Developer tab will be visible, click the **Visual Basic Icon** which brings up the VBA window.



Figure 24

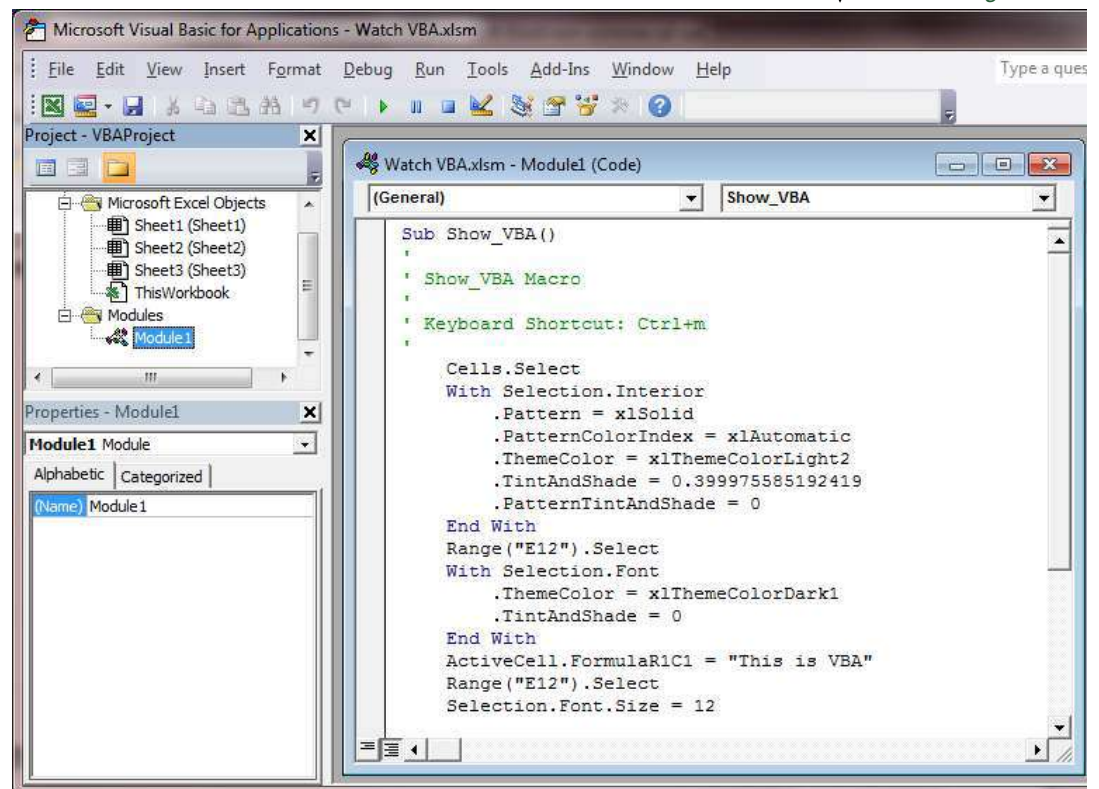


Figure 25

## Task 10

### Record a macro and watch the VBA appear

1. Arrange the Excel window and the VBA window side by side.

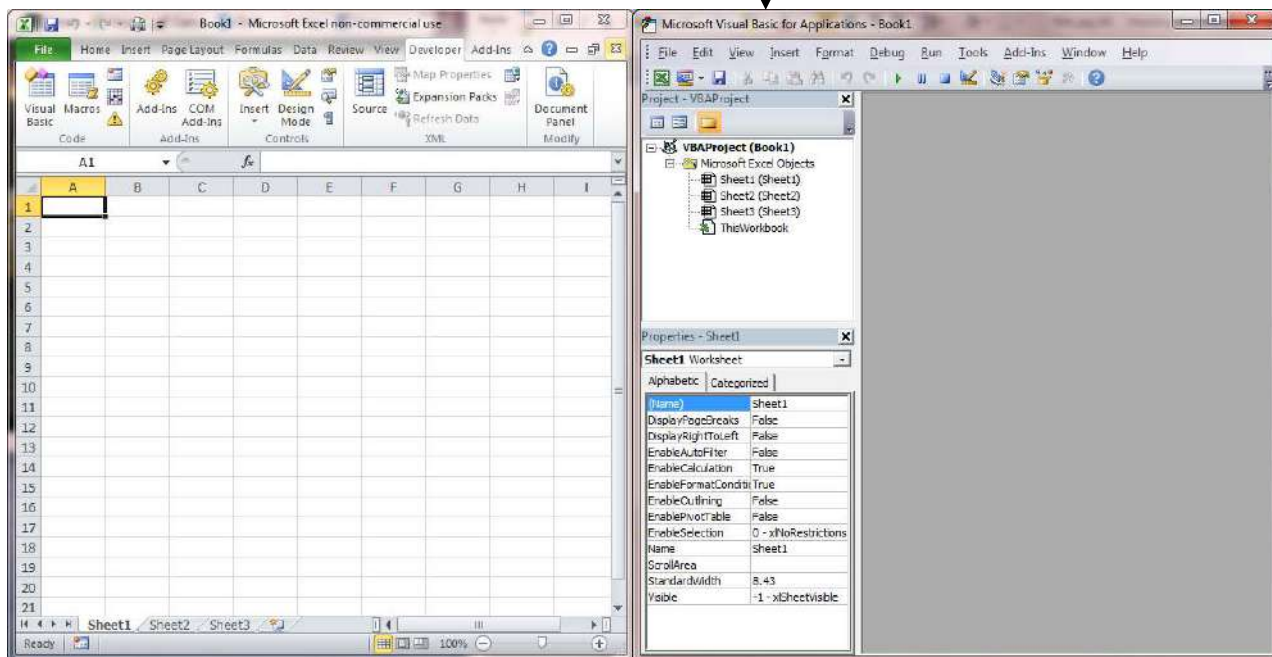


Figure 26

2. **Start recording the macro**, call it 'Test', as soon as you press **OK**, go to the VBA window and click on **Modules** → **Module 1**.  
You will now see your macro.

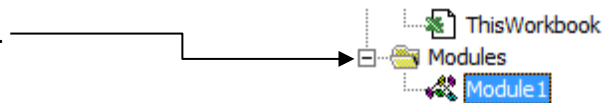


Figure 27

3. **In Excel**, select the whole worksheet, change the background, type some text and format it. When you have finished, **stop the macro**. You should now be able to look at the code and match it to what actions you performed with the mouse.

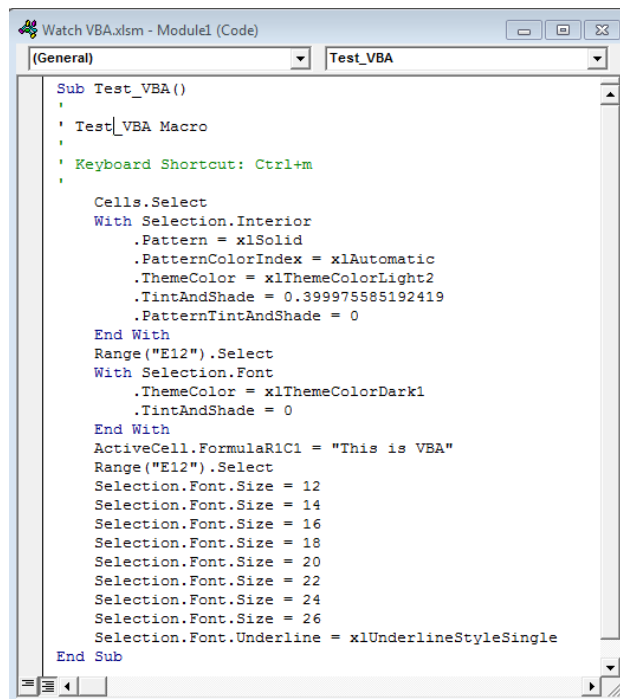


Figure 28

## Task end



## Contacts

### For queries or help

---

- email: [ittraining@essex.ac.uk](mailto:ittraining@essex.ac.uk)

## Useful resources

### Training videos

---

- [www.youtube.com/uniessexit](http://www.youtube.com/uniessexit)

### Booking other courses

---

- [www.essex.ac.uk/it/training/courses/](http://www.essex.ac.uk/it/training/courses/)

## Your feedback

### Help us improve your training

---

Your feedback is important to us.

You will receive an email asking you to complete an online feedback survey.

Please take a few minutes to tell us what you think.