Macros 1 -Recorded Macros

Tips, Tricks and Traps

By Neale Blackwood

A4 Accounting

a4@iinet.net.au

A4 ACCOUNTING

EXCEL CONSULTING, TRAINING AND WEBINARS

1. Introduction

Why learn about macros?

Macros are a powerful tool that can dramatically increase your productivity when working in Excel. Macros can extend Excel's functionality and speed up a lot of mundane tasks. Many accountants spend their working life in Excel and the ability to use it better is an important skill.

Macros can enable Excel do almost anything you want it to.

If you have a task that

- Is repetitive
- Is prone to human error
- Is difficult
- Is time consuming
- Involves many keystrokes and/or many mouse actions
- Is impossible with built-in Excel commands
- needs to be run at certain times

then a macro may provide a solution that will save time and effort and provide more accurate results.

What is a Macro?

A macro is a series of Visual Basic for Applications (VBA) statements that performs a certain task. VBA is a subset of the Visual Basic programming language and it works with Microsoft Office programs. Macros are used to automate tasks and provide functionality that standard Excel doesn't have.

Excel skill level

I would suggest that the better you are with Excel the better and more useful your recorded macros will be. One problem with Excel is that most users are not aware of most of its features. Excel does a lot of very clever things straight out of the box.

If you don't know Excel's most useful features two things will happen.

- You won't even try to do something because you don't know Excel can do it
- You will build a macro to do something that Excel already does

When you move to a new version of Excel it is important to find out the new features because what you want to do may have been added to Excel's functionality.

Learning macros requires persistence

As you progress you will encounter things that don't work as you expect and you must be flexible enough to learn from your mistakes. Don't give up too easily. There are lots of resources you can learn from. See Appendix 1.

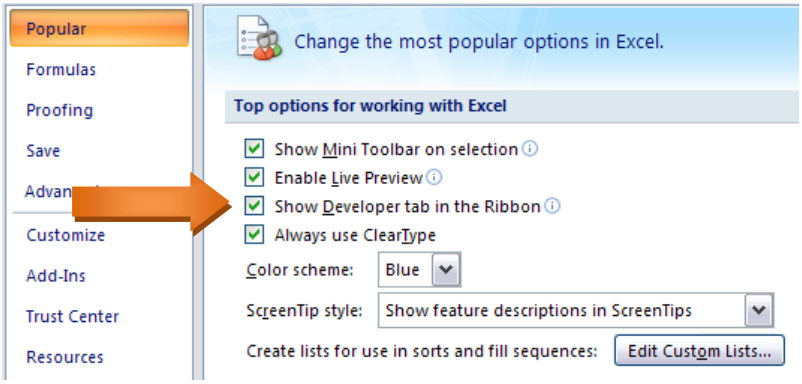
© 2016 Neale Blackwood

2. Where to Start?

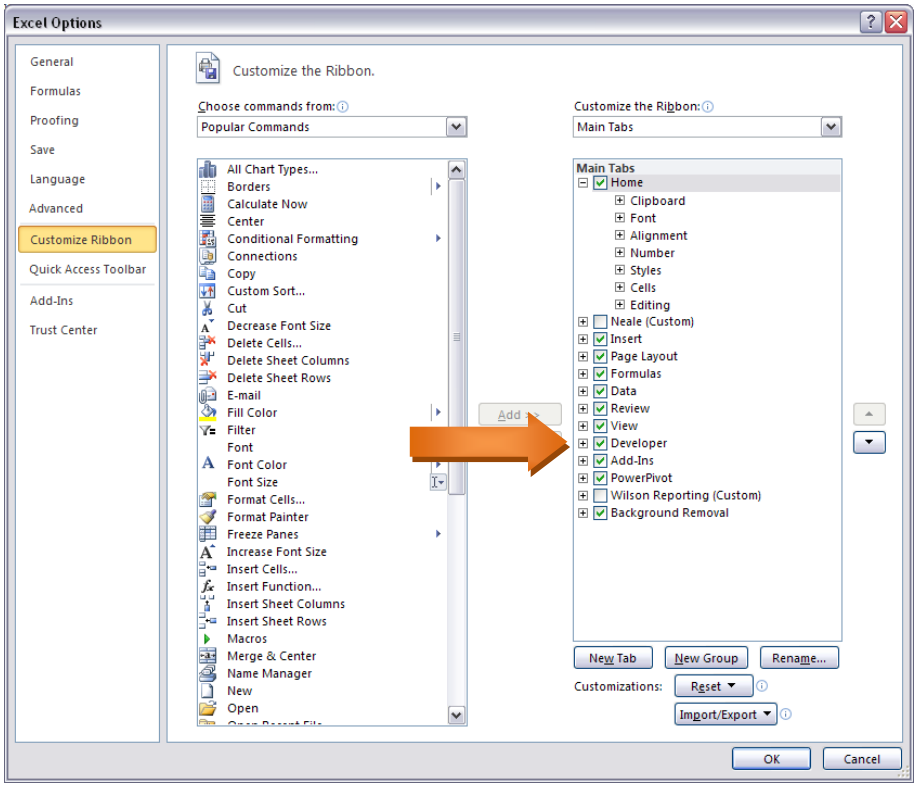
Adding the Developer Ribbon

The Developer ribbon is dedicated to VBA and doesn't display in the normal installation of Excel. You must add it yourself.

Office 2007 - Click the round Office button, click Excel Options and tick the Show Developer tab in the Ribbon option, click OK.

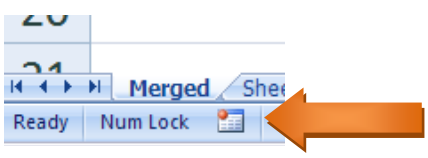


Office 2010 – right click the ribbon headings and click Customize Ribbon and tick the Developer option and click OK



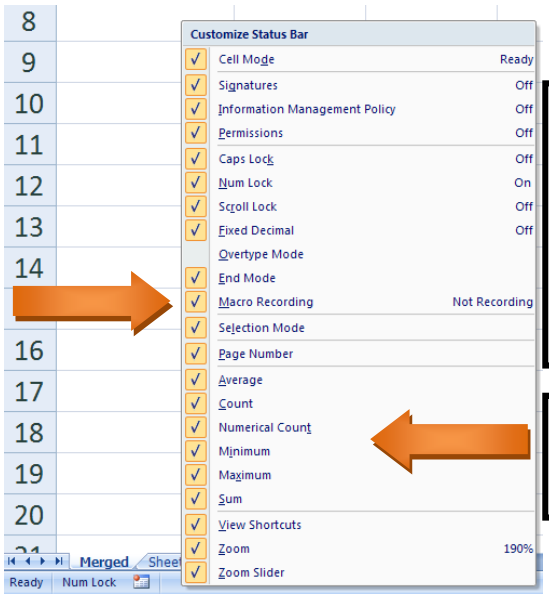
Record Macro button

Recording a macro is typically the first step people take in using macros. It is as easy as pressing the Record button, doing what you need to do and then pressing the Stop button.



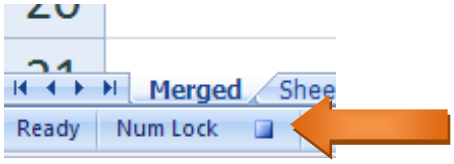
Record Macro button

Bottom left of screen



If the Record Macro button is not visible - right click the Status Bar and select Macro Recording from the menu.

This also has the Auto Calculate options



Stop Recording button

Bottom left of screen

The icon will change as soon as you start recording

Enabling Macros

Lastly ensure your Macro security is set as shown below which will ask you to enable macros when you open a file containing macros.

Steps

1 Click the Round Office Button (File Tab in Excel 2010)

2 Click Excel Option (Options in Excel 2010)

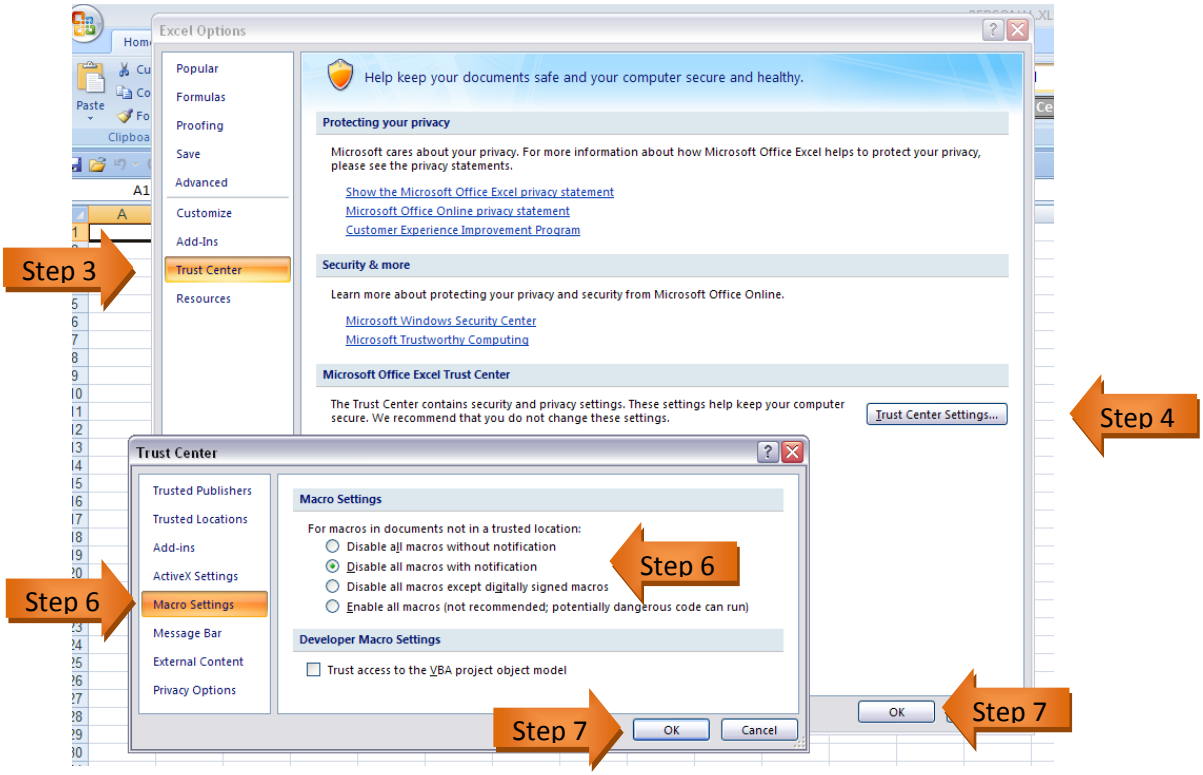
3 Click The Trust Center on the Left

4 Click the Trust Center Settings button

5 Click the Macro Settings on the left

6 Click the option to Disable all macros with notification

7 Click OK twice



Macro File types

In Excel 2007 and 2010 you must save files containing macros as Macro Enabled workbooks this uses the extension .xlsm rather than .xlsx.

WARNING:

If you save a file containing macros as a standard .xlsx file you will lose all the macros – Excel will warn you, so read saving dialogs carefully.

3. Recorded Macros

When you record a macro Excel watches what you do and creates VBA code that will replicate your actions. You can then run the recorded macro to repeat your actions. Recorded macros can be useful for repetitive tasks that don't vary. Processes like complicated formatting or copying and pasting between sheets.

Recorded macros have a number of limitations

- no interaction with the user e.g. can't ask the user a question
- they can crash reasonably easily, e.g. if a sheet name changes
- the ranges used in recorded macros are not updated if columns or rows are inserted or deleted
- the code works sequentially from start to finish and cannot make any decisions or handle errors

Whilst these limitations affect the usefulness of recorded macros, this doesn't mean that you don't need to record macros. Virtually all VBA programmers record macros regardless of their skill level.

Recording macros is the fastest way to create VBA code and it also quickly teaches you the VBA commands. It sure beats typing!

Future webinar sessions will address how to remove all those limitations using VBA code.

Recorded macros - the positives

- Easy to create and re-run
- Good for sequential tasks that don't require decisions
- If you use keyboard shortcuts to select ranges then recorded macros have some flexibility (see Exercise 3)
- Using range names to select ranges allows recorded macros to handle inserted rows and columns

There are a few things to consider when recording macros.

The first consideration is where you want to use the macro and the type of macros you are recording. Some macros are designed to work in only certain files and are usually saved in those files. Other macros are more general in nature and are typically saved in the Personal Macro Workbook.

Personal Macro Workbook (PMW)

When you record a macro you get the option to save it in the current file, a new file or the Personal Macro Workbook (PMW). The PMW is a special workbook that is saved in the XLSTART folder (see note below) and it opens each time Excel opens. The first time you record a macro and save it to the PMW Excel creates the PMW. The PMW is a hidden workbook. This means that when it is open it is

not visible. Macros in the PMW are always available for use. To unhide the PMW click the View ribbon and select Unhide. Select the PMW and click OK.

Be aware that the PMW is saved on your primary hard drive and as such may not be backed up by your standard system backups. It is a good idea to take a copy regularly and place it on a server that does get backed up. If Excel is reinstalled, or your system is a reinstalled, sometimes the PMW can be erased, which is another reason to keep a backup.

XLSTART Folder

The XLSTART folder (where the PMW is saved) resides on your primary hard drive. Any files or shortcuts to Excel files that are saved to the folder will open every time you open Excel.

Recording considerations

Selecting Ranges

Before recording the macro you should determine whether the macro is to work on any range that you select or on a specific range each time it is run. A macro is more flexible if it will work on any range and you may be able to reuse it for other projects.

Another consideration is if the range is going to change due to inserted rows or columns and whether you need to use range names to refer to the ranges.

A flexible macro or a fixed range macro is determined by when you select the range in relation to starting the recording. If you select the range before you click the Record Macro button then, Excel will work on the selected range which it calls Selection (this is an important keyword in VBA).

If you choose a range after you click Record then Excel will always choose that same range when it runs the macro. The range selected after the macro has started recording will be recorded as a fixed reference and it will not adjust if rows or columns are inserted or deleted.

Practice

You should practice your more complex recorded macros to iron out any issues before you start recording – macros record all that you do - even mistakes!

Let’s record a macro and see what happens.

Recording a useful macro

When deciding whether or not to create a macro to do a certain action you need to weigh up the pros and cons between whether you keep doing what you’re doing or whether recording a macro will save you time and effort. Often the time taken to do the task can be used to create the macro and then the task has been automated for future use.

80/20 rule – sometimes macros can’t do everything in a task, but they can do 80% of the work, thus reducing the time taken to complete the task.

One advantage with macros is that once they have been debugged (tested and fixed) they do usually work without error which means that you can virtually set them and forget them. So you can click a button and the actions will be done.

WARNING:

In general macros cannot be undone and running a macro clears the undo list. In some cases you may need to save the file before running a macro to enable you to undo what has been done. You can close and not save after running the macro to revert back to the pre-macro workbook.

Recorded Macro – Apply Center Across Selection format

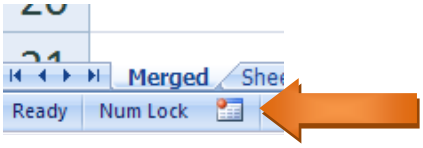
Merged Cells format can cause some macros to crash and they have other issues when you try to copy and paste or modify a range. A better alternative is to use Center Across Selection format which has no issues.

There is no easy shortcut for this format, so it is ideal for a macro solution to save a few clicks each time you use it.

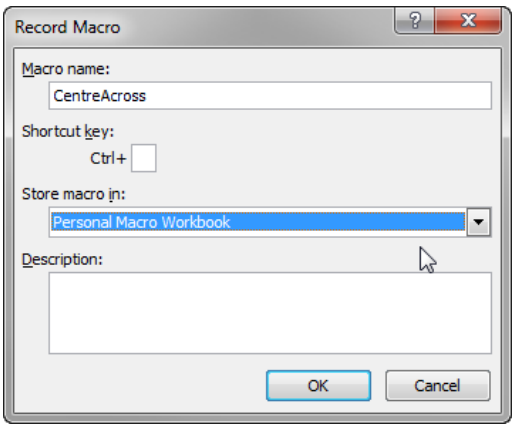
Exercise 1

1. On the Center Across sheet select the range B6:D6 – we are selecting the range first to make this macro flexible.

2. Click the Record Macro button at the bottom of the screen.



Enter the Macro name as CentreAcross as below – click the drop down and select Personal Macro Workbook and click OK. Excel will now be recording your actions.

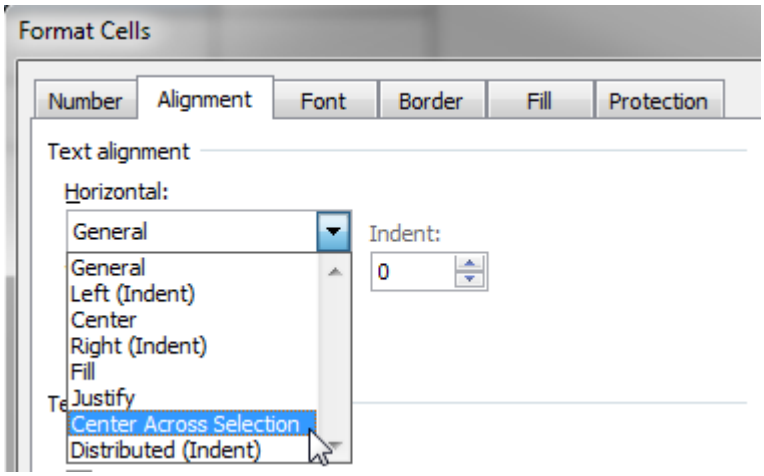


When you save your first macro to the Personal Macro Workbook Excel will create that workbook. You may need to save it when asked.

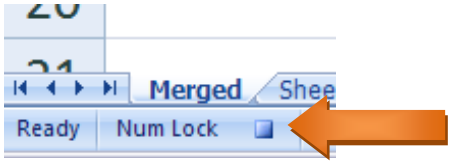
Note: You can't have spaces in Macro names. You can use the _ (underscore character) eg

Centre_Across

3. Press Ctrl + 1 to open the Format Cells dialog and click the Alignment tab
   Click the Horizontal drop down and select Center Across Selection see image below.

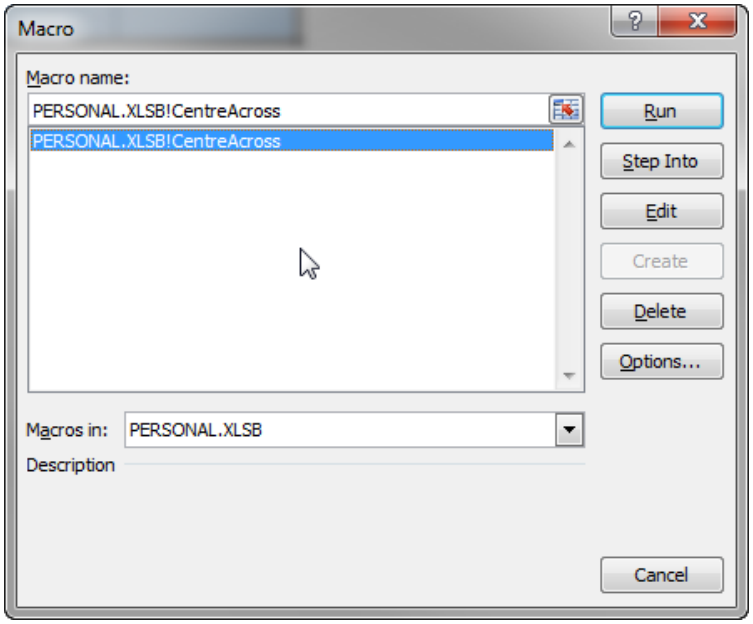


4. Click OK and then click the Stop Macro button at the bottom of the screen.



5. The macro has been created. Let's test it – always test your macros.

6. Select the range A8:E8 and press Alt + F8 – this opens the Macro dialog below.



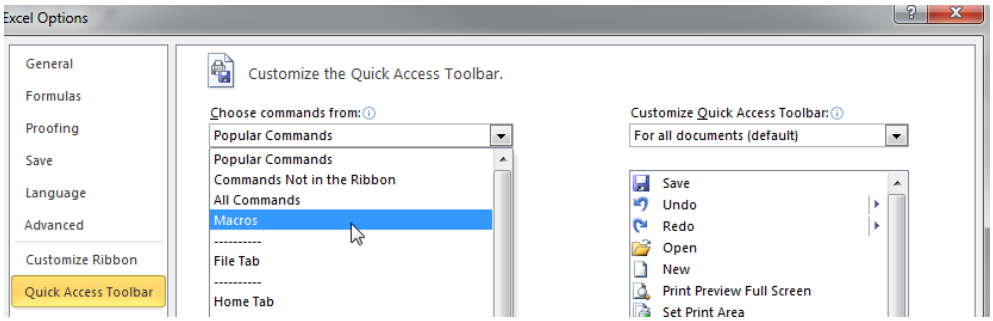
7. With the CentreAcross macro selected press Enter or click the Run button.

8. The text should have been formatted.

9. Next let's put a button on the Quick Access Toolbar (QAT) to run the macro whenever we want it.

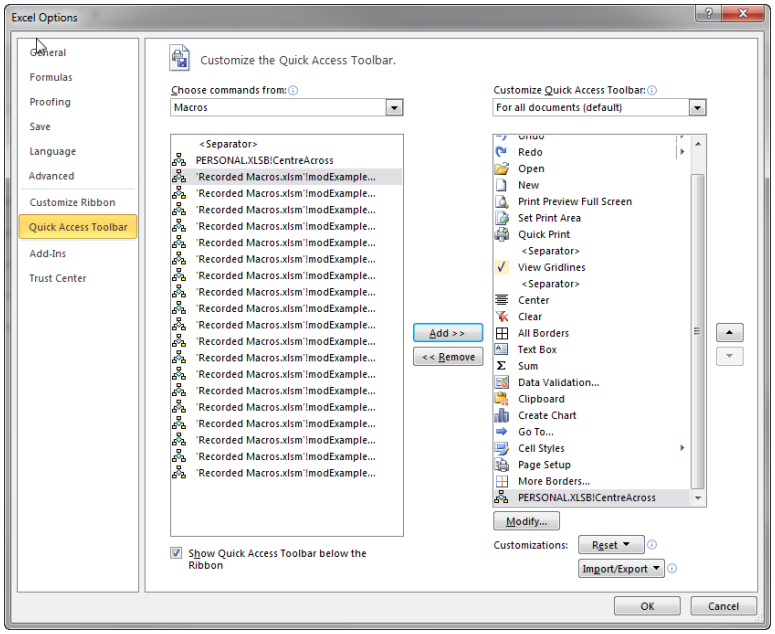
F4 key – If you have just run a macro pressing F4 will re-run it quickly and easily.

10. Right click the QAT it's either above the Ribbon or below it – it has the Save icon as the first icon. Choose *Customize the Quick Access Toolbar*

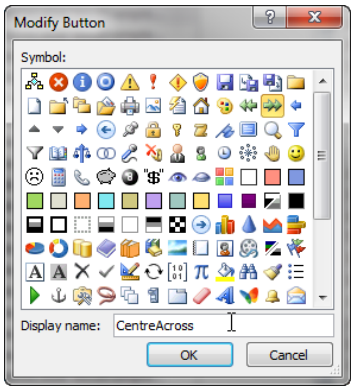


11. Click the *Choose commands from* drop down and select Macros.

12. Choose the CentreAcross macro and click the Add button



13. With the CenterAcross macro selected on the right-hand side click Modify and choose an icon and change the Display name to CenterAcross then click OK and OK again to finish.



The Display name is what is displayed when you point the mouse at the icon on the QAT

14. Select the range A10:E10 and test the new QAT icon.

15. Next we'll examine the macro that was created.

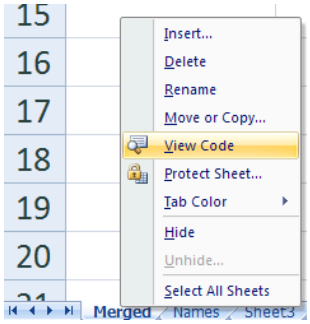
4. VBA Window

If you only ever want to record macros you don't need to review your code.

I recommend however that you review all your macros. You can learn a lot by reviewing the code created by the macro recorder.

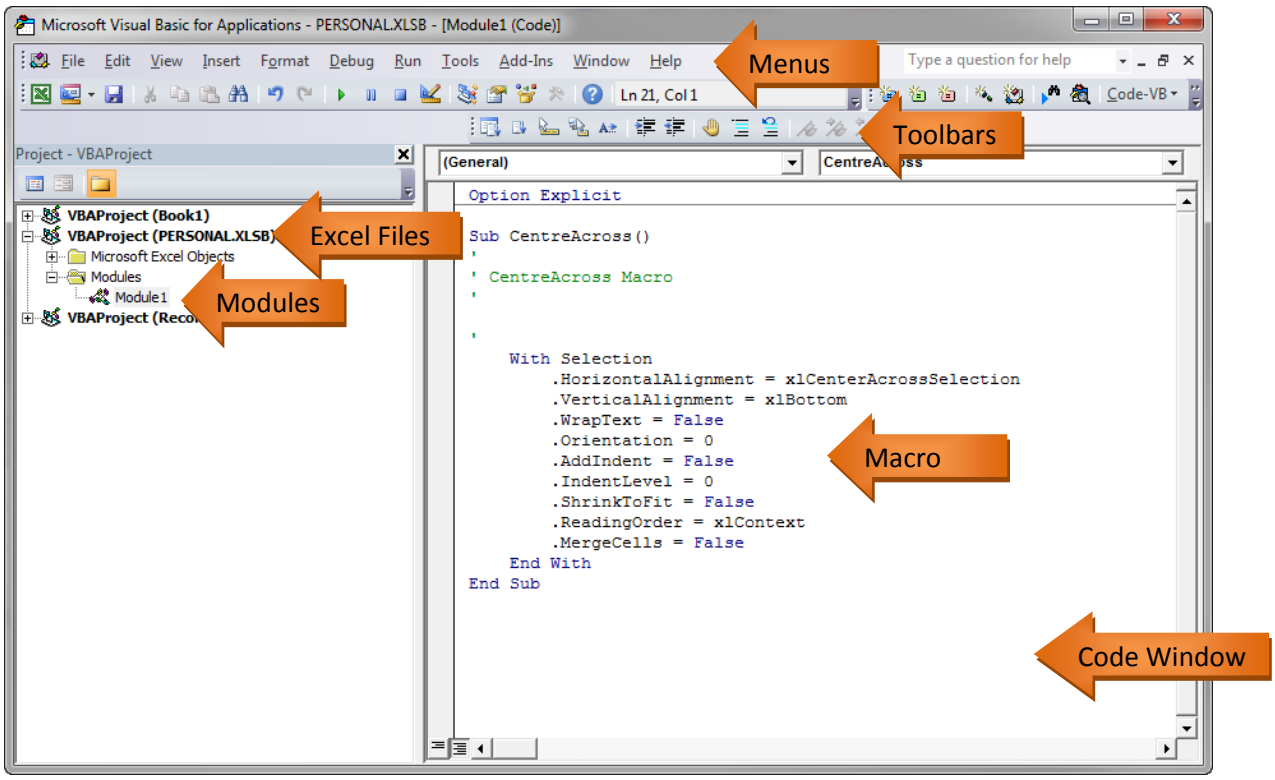
Opening the VBA window

- Right click any sheet tab and select View Code



- Alt+F11 will also open the window and acts as a toggle to switch between the Excel window and the VBA window.

Layout of the VBA window



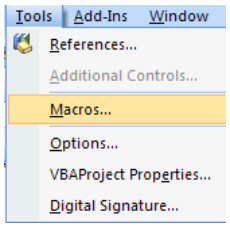
Modules

All recorded macros are created in Modules. Excel uses the generic Module1, Module2 names.

As a general rule all your macros should be saved in a Module. The exceptions to this are event macros which are entered in individual Sheets or in ThisWorkbook.

If you create a macro in a sheet and that sheet gets deleted then the macro is also deleted. So always save your macros in a Module.

To see your macro double click Module1 under the Personal.xlsb File.



You can also click the Tools menu and select Macros…

Change the File drop down to the Personal Macro workbook and select the macro you want and click Edit button

Edit Toolbar

Right click the Toolbar and make sure Edit is ticked – this has some useful icons when you are programming.

Colour coding of text in the VBA Code Window

Excel uses basic colour coding in the VBA code window to help you identify the types of commands that Excel uses.

Green – is used to identify comments in code.

Blue – keywords

Black – all other words

Red - Errors

5. Editing Recorded Macros

Recorded macros are usually inefficient

It is important to realise that whilst Excel creates VBA code quickly and easily, the code it creates is usually inefficient. As you use VBA more you will learn to recognise the parts of the code that can be removed or improved.

One of the advantages of VBA compared to other programming languages is its readability. You can read the code (with no training) and still get a reasonable understanding of what the code does.

The macro captured every option in the Alignment tab – even though we only changed one option. Each line represents one option's setting.

```
Sub CentreAcross()
'
' CentreAcross Macro
'
'
    With Selection
        .HorizontalAlignment = xlCenterAcrossSelection
        .VerticalAlignment = xlBottom
        .WrapText = False
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = False
    End With
End Sub
```

The above macro can be modified to the following single line of code.

```
Sub CentreAcross2()

    Selection.HorizontalAlignment = xlCenterAcrossSelection

End Sub
```

This new macro will perform the one format change that we need.

The key word in both macros is Selection.

Selection means whatever you have selected before running the macro. If you had a Chart selected the above macro will crash. This macro is designed to have a range selected when you run it.

Recorded Macro – Changing Page Setup options

Changing the Page Setup can take a few clicks. Let's record a macro to standardise some Page Setup options.

Exercise 2

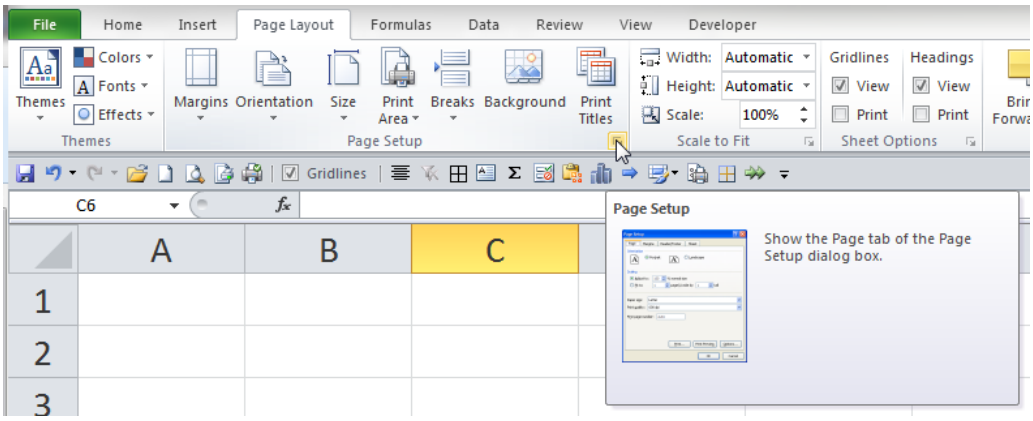
1 Open the Report sheet

2 Click the Record macro button

3 Name the Macro StandardPage and save it to the PMW click OK

4 Click the Page Layout ribbon

5 Click the Page Setup drop arrow



6 Choose Landscape

7 Click the Scale to fit drop arrow and select Fit to 1 and 1

8 Click the Sheet tab choose Blank and White

9 Click the Stop recording Macro button

The code created by Page Setup is always inefficient as it lists all the options possible whether they were changed or not. You can go through and delete all those options that you didn't change and convert over 50 lines to 9 lines as below.

```
Sub StandardPageShort()

    With ActiveSheet.PageSetup
        .Orientation = xlLandscape
        .BlackAndWhite = True
        .FitToPagesWide = 1
        .FitToPagesTall = 1
        .Zoom = False
    End With

End Sub
```

6. Worked Example

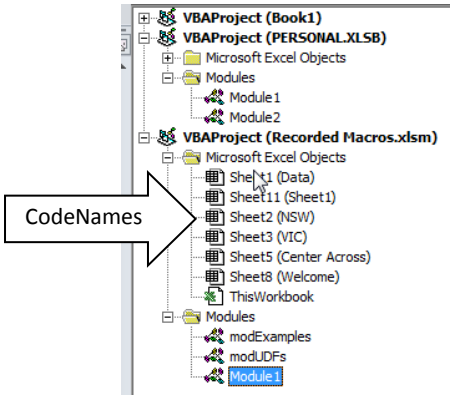
Let's create a flexible macro. We will use a keyboard shortcut to select the range that will add flexibility to the macro.

Challenge

We have a data table of names and addresses. We want to copy the NSW and VIC addresses to separate sheets, one for NSW and one for VIC. The table has been formatted using the Format As Table icon on the Home ribbon tab. This makes working with tables in VBA much easier.

Exercise 3

1. In the Data sheet select any cell within the data table.
2. Click the Record Macro button (bottom left of screen) and name the macro NSWVICData and save it in the current file then click OK.
3. Click the State Filter drop down and click Select all and then click NSW.
4. Press Ctrl + * this keyboard shortcut selects the current region and is very helpful when recording flexible range macros.
5. Press Ctrl + c to copy the range then select the NSW sheet and click cell A1 and press Ctrl + v to paste the data.
6. Select the Data sheet use the State Filter to select VIC and press Ctrl + * and Ctrl + c
7. Open the VIC sheet and select cell A1 and press Ctrl + v
8. Return to the Data sheet and click the Data Ribbon tab and click the clear Filter icon. As a general rule you should return the user to the sheet they started on unless the macro is supposed to take them somewhere. You should also clear the filter you applied.
9. Click the Stop Macro button
10. Go to the NSW and VIC sheet and delete the contents – Press Control + a and then press the Delete key on each sheet.



11. Test the macro by clicking in the Data sheet and select any cell in the table. Use Alt + F8 to find and Run the macro.

See the VBA code on the following page and note the changes - these handle sheet name changes.

The amendment uses the sheet CodeName – see image on left. – instead of the sheet tab name. All recorded macros use the sheet tab name and this can cause errors when the sheet tab name changes.

Data = Sheet1, NSW = Sheet2 and VIC = Sheet3

```
Sub NSWVICData()
'
' NSWVICData Macro
'
    Range("E10").Select
    ActiveSheet.ListObjects("Table1").Range.AutoFilter Field:=6, Criteria1:= _
        "NSW"
    Selection.CurrentRegion.Select
    Selection.Copy
    Sheets("NSW").Select
    Range("A1").Select
    ActiveSheet.Paste
    Sheets("Data").Select
    Range("E21").Select
    ActiveSheet.ListObjects("Table1").Range.AutoFilter Field:=6, Criteria1:= _
        "VIC"
    Range("E54").Select
    Selection.CurrentRegion.Select
    Application.CutCopyMode = False
    Selection.Copy
    Sheets("VIC").Select
    Range("A1").Select
    ActiveSheet.Paste
    Sheets("Data").Select
    Range("F10").Select
    Application.CutCopyMode = False
    ActiveSheet.ShowAllData

End Sub
```

Other useful keyboard shortcuts – assumes no blank rows

Ctrl + up arrow - selects the cell at the top of the current column

Ctrl + Shift + down arrow – selects a range down from the current cell to the bottom cell of the current column

```
Sub NSWVICData2()
'
' NSWVICData2 Macro
'
    Range("E10").Select
    ActiveSheet.ListObjects("Table1").Range.AutoFilter Field:=6, Criteria1:= _
        "NSW"
    Selection.CurrentRegion.Select
    Selection.Copy
    Sheet2.Select
    Range("A1").Select
    ActiveSheet.Paste
    Sheet1.Select
    Range("E21").Select
    ActiveSheet.ListObjects("Table1").Range.AutoFilter Field:=6, Criteria1:= _
        "VIC"
    Range("E54").Select
    Selection.CurrentRegion.Select
    Application.CutCopyMode = False
    Selection.Copy
    Sheet3.Select
    Range("A1").Select
    ActiveSheet.Paste
    Sheet1.Select
    Range("F10").Select
    Application.CutCopyMode = False
    ActiveSheet.ShowAllData

End Sub
```

Use the sheet code names Sheet2 and Sheet1

Use the sheet code names Sheet3 and Sheet1

Where to from here

This has been an introduction to recorded macros and contains some tips to creating better and more flexible recorded macros. Creating your own is the best way to learn and improve.

Future sessions will focus on editing recorded macros to make them even more flexible and useful, as well as creating macros from scratch using VBA code.

Bonus macros

See the modBonus module for bonus macros. The Appendix 2 – Bonus macros details what they do and how to use them.

Appendix 1- VBA Resources

Recommendations

Dual Screens

If you are using macros and the VBA window frequently you will find using dual screens more efficient. The VBA window can be on one screen and Excel on the other screen and you can see the effects of your macros immediately without having to change Windows. This speeds up working with macros. It is also beneficial to display other Windows in separate screen, eg Outlook or a Browser.

Reading List

The best Excel author is John Walkenbach, a couple of his VBA books are

VBA for Dummies

Power Programming with VBA (various Excel versions)

A great reference book is

Excel 2007 VBA by Green, Bullen, Bovey and Alexander

Best book buying sites – free postage

www.bookdepository.co.uk

www.fishpond.com.au

Recommended Web sites

www.a4accounting.com.au	My site
www.ozgrid.com	WA site
www.mrexcel.com	US site
www.chandoo.org	Indian Site

Webinar Series Topics

Macros 2 - From Recorded To Remarkable

Macros 3 - Interacting with the user

Macros 4 – Events - automate your macros

Macros 5 – User Defined Functions – make your own Functions

Macros 6 + 7 – Workshops 1 + 2

© 2016 Neale Blackwood

Appendix 2- Bonus Macros

These macros are not recorded macros. These macros demonstrate some of the techniques that are covered in future webinars. Some contain basic error handling.

CentreAcrossSelection – this macro turns on and turns off the CentreAcrossSelection format. The one macro does both. Simply select the range and run the macro.

ApplyAutoFit – this macro applies the column AutoFit format across all the sheets in the current workbook in one step.

ChangeSumToSubTotal – select a range before running this macro - it converts standard SUM functions to SUBTOTAL functions. SUBTOTAL is the best function to use for subtotaling, but it doesn't have an automated way to create it. This allows you to create a series of SUM functions using the AutoSum icon (easy and quick) and then convert all of them into a SUBTOTAL function in one step.

SaveAllFiles – this saves all the open workbooks in one step. It doesn't save files that have yet to be saved eg Book1

© 2016 Neale Blackwood