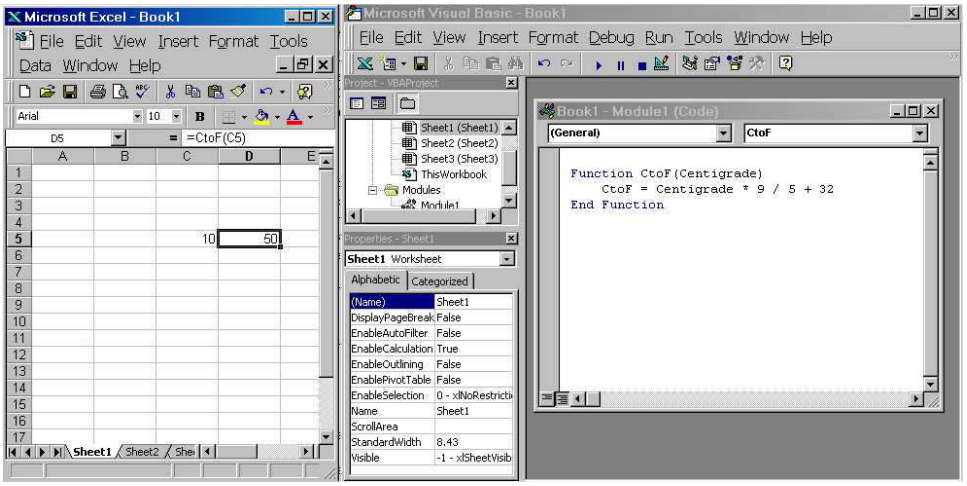
TECH EXCHANGE No. 9 – Microsoft Visual Basic for Applications (VBA) (Part 2)

In this part we will take a look at some uses of VBA in Microsoft Excel. First some history, Excel made its debut on the Macintosh in 1985. In 1987 it was released for the Windows operating system. There were spreadsheets prior to this, VisiCalc, Multiplan and of course 1-2-3. VisiCalc started the spreadsheet revolution but fell by the wayside early on. Multiplan was Microsoft's predecessor to Excel, which used the RnCn (row-column) method of cell addressing, which is still available as an option in Excel. But it was Lotus 1-2-2 released in 1982 that dominated the PC spreadsheet market. This was in the early days of the Windows OS (Operating System), when another GUI (Graphical User Interface) OS was making a run for the hearts and hard drives of our computers - IBM's OS/2. The early versions of windows needed much more computer resources to operate then the DOS (Disk Operating System) and the new OS/2. So Lotus thinking Windows was going to be a flash in the pan, made a business decision not to produce a Window's version of 1-2-3. This was going to prove their biggest mistake, by the time Windows was well established and they had reversed their decision, users had deserted Lotus 1-2-3 for Microsoft Excel. Legend has it that Lotus set-up macros in 1-2-3 as a testing and debugging aid and only realised the potential of macros at the last moment and included them in the release version. The original Excel macro language required you to write your macros in a special spreadsheet called a macro sheet and saved with a .xlm file extension. This kept them separate from the worksheets that were saved with a .xls file extension. These were often referred to as XLM macros or Excel 4 macros. Excel 5 released in 1993 introduced VBA (Visual Basic for Applications) as the common macro language and the rest is history.

```
Function CtoF(Centigrade)
    CtoF = Centigrade * 9 / 5 + 32
End Function
```

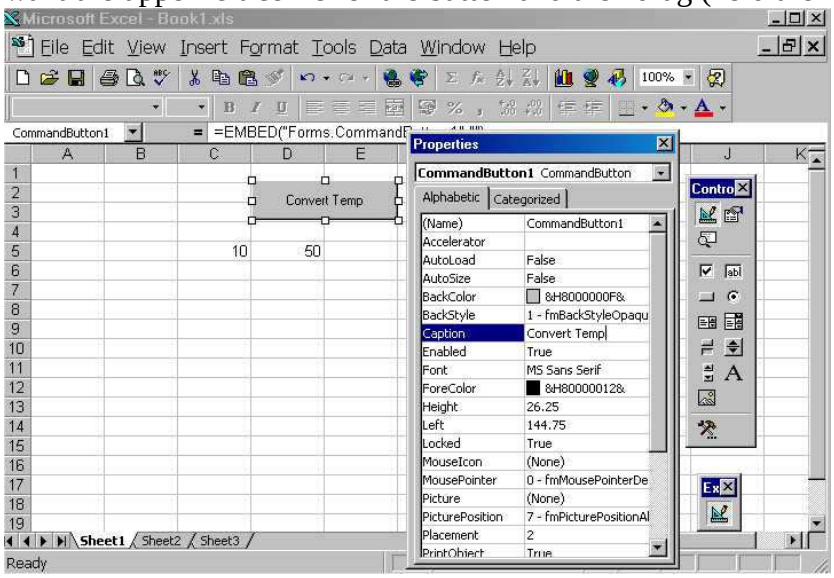



So to our first example, we are going to write a function that will take a temperature in degrees Centigrade and convert it to degrees Fahrenheit. In part one I described how to run the Visual Basic Editor (VBE) and a brief tour of its features. To recap use the keystroke **ALT-F11** from within Microsoft Excel to invoke the VBE. Next we want to insert a standard code module by right clicking in the Project Explorer window and choosing **Insert | Module**. Enter the code shown in the above box and close the VBE to return back to the spreadsheet. You will notice that that as you press enter after each line of code, the VBE will syntax check and colour code each line. You will notice that to return a value from the function, we must assign it to a variable that has the same name as the function. To test our function we can pick any two spreadsheet cells, say C5 and D5. Into cell C5 we enter the temperature in degrees Centigrade we wish to convert and into cell D5 we enter **=CtoF(C5)**. When we press return cell D5 shows us the temperature in degrees Fahrenheit as shown in the screenshot. If we change the value in cell C5 the temperature in degrees Fahrenheit is automatically recalculated. Congratulations you have written your first program.

```
Sub getCtoF()
   Centigrade = InputBox(Prompt:=”Enter degrees Centigrade”)
   If Centigrade=”” Then Exit Sub
   Range(“C5”).Value=Centigrade
End Sub
```

For our second example we will expand on the example above to introduce some new concepts to provide a starting block for your own experimentation. We are going to place a command button on the spreadsheet, that when pressed will display a prompt allowing the user to enter a temperature in degrees Centigrade. This value will be placed in the cell we are using for the conversion, i.e. C5, which will cause the converted temperature in degrees Fahrenheit to be displayed in cell D5. Open up the VBE

with **ALT-F11** and enter the code in the box above and then close the VBE to return to the spreadsheet. You will notice that the VBE places a dividing line between each function. Next we need to place the command button on the spreadsheet. Excel has two different sets of controls that can be embedded in spreadsheets. One set is on the Forms toolbar and the other is on the Controls toolbar. The Forms toolbar has been inherited from Excel 5 and 95. Excel 97 introduced the newer ActiveX controls that are selected from the Controls toolbar. Display the Controls toolbar, if not already visible with View | Toolbars | Controls. To create an ActiveX command button control, click on the sixth button on the Controls toolbar. Click on the spreadsheet where you want the upper-left corner of the button and then drag (hold the mouse button down) the mouse to draw the button to the required size. If you press the **Alt** button while dragging the button size will snap to the nearest grid cell. When you draw the button on the spreadsheet, design mode is automatically selected. In design mode you can select a control with a left-click and edit it. You must exit design mode if you want the control to respond to events, e.g. when we click on the button. There is a button on the Controls toolbar for entering and exiting design mode. Next we have to attach a piece of code to the button that is executed when we click the button. This is what is known as an event procedure. To do this make sure you are still in design mode and double click the command button. This will open the VBE and display the code behind the spreadsheet. The Sub and End Sub code statement lines have already been added for you. In-between these two lines we need to enter a line of code which will call the function we entered above. Type in **Call getCtoF** and then close the VBE. This piece of code calls (runs) our function we entered above when we click the button. It would be nice to have some more descriptive text on the button (its caption), so once again make sure you are in design mode and the button is selected. Click the properties toolbar button which displays the Properties window (see screenshot). Change the caption entry to say 'Convert Temp'. Exit design mode and click on the button. You should get a dialog box asking you to enter a temperature which if you click Ok will be displayed in cell C5, causing the converted temperature in degrees Fahrenheit to be displayed in cell D5. Congratulations you have written your second program and from little acorns great big oak trees grow.




In future articles we will continue our look at the use of VBA in the Microsoft Office components. In the next article we will look at one of the common uses of computers today - Email. That's it for this week, please send any questions or comments to TECHEXECHANGE@ENDHOUSESOFTWARE.COM.

By Gavin Baker.