

Why would you want to use VBA?

Visual Basic for Applications (VBA) is a powerful component of the Microsoft Office Suite

By Jackie Grubb (jackie@plumsuite.com)

Many Microsoft Office users are unaware that, underlying the entire suite, there is a powerful component available for use. Called Visual Basic for Applications (VBA), it is a full-fledged programming language. It is a part of Visual Basic, a programming language used by professional programmers, but unlike Visual Basic, which needs to be purchased separately, VBA is included in the Office Suite package.

VBA can be used for as small a thing as *automation* of a repetitive task, or it can be used to develop a full computer application. It is available in all of the Office Suite components, and can even be used from one of the components to automate an action in another component. For instance, from within Word, you could open up an Excel spreadsheet change the data in some of the cells, close Excel and return back to Word. Microsoft calls this, appropriately enough, *automation*.

How to get started

You could use a macro to get started in Word, Excel, and PowerPoint. Before I carry this too far, let me point out that there are differences between the various versions of Microsoft Office. The instructions I am giving are for Office XP (also known as Office 2002) and so if you are using an earlier version, the process, while similar, may not be exactly the same.

Since I am using Word in this get-started example, I also need to point out to you that your mouse will not act as it normally does in Word. During macro recording, you will need to use keystrokes instead of the mouse. Most of the keystrokes are pretty straightforward, but one that you might well need is the keystroke that starts defining text. It is the F8 key. Press it twice to select the word the cursor is in. A third press will select the sentence, a fourth the paragraph and the fifth the entire document. Use the Escape key to release the selection.

Another keystroke sequence that is useful when recording a macro is that which will get you up into the Menu Bar. Note that each of the menu items has a letter underlined. Using keystrokes, if you hold down the [Alt] key and press the underlined letter of the menu item you wish to use, it will drop down. So, if you wanted to format a word with a different font, you would press [Alt] then 'o' then 'f'. Note that the dropdown items on the menu also have a letter underlined.

You could also make your choice by using your cursor arrow keys and pressing [Enter]. While in the dialog box, you can move from choice to choice by using the [Tab] key. And, if you want to change a check box, tab to it and then press the space bar, which acts as an on-off toggle in that situation.

Now let's give it a try. In Word, open a new document and pull down the *Tools* menu. Then choose *Record New Macro* from the *Macro* submenu. Use the default choices in the dialog box for this first try, but note the name suggested – Macro1, for instance. Then go through a series of actions in your document. Type in some text, select a word or a sentence and change the font or put it in italics. While you are doing this, note that there is a small tool bar on the screen containing *Stop* and *Pause* buttons. After you have done a few things, click on the *Stop* button – or pull down the *Tools* menu and choose *Stop Recording* from the *Macro* submenu.

Next, try running the macro you just created. Do this by again pulling down the *Tools* menu and then choosing *Macros* from the *Macro* submenu. The macro you just recorded should show on the list. Highlight it and click on *Run*.

Now it is time to go look at the code that you have just created. Again, pull down the *Tools* menu and choose *Macros* from the *Macro* submenu. Highlight it, but this time click on *Edit*. You have now entered the wonderful world of VBA code.

If you have done any computer coding in the past you probably understand the importance of syntax. If the syntax is not exactly right, the code will not work. So you will need to be very careful about how you go about editing your code. One of the first things I suggest you try is turning a 'False' into 'True' or vice versa. So, if you set a particular word to be italic, you might want to make it be bold instead.

Excel and PowerPoint also have similar macro recording capabilities to Word. In Outlook you can create a macro from scratch – but there is no recording capability.

Access, however, is somewhat different, and is also a place where a lot of VBA is used by professional programmers. Access also has macros – but they are pretty much separate from VBA. VBA programmers tend to look down their noses at macros – mostly because they have no means of handling errors. If you are putting together an application for others to use, this can become a problem.

A macro in Access does not look like code at all. It is a spreadsheet-like chart that contains a list of actions to be taken in Access. Anything that can be done as a macro in Access can also be done in VBA – and indeed, Access provides a conversion utility that takes a macro and turns it into VBA code.

So, you could get started doing VBA in Access by first creating a macro – I suggest you see Access help on how to do that. Then highlight the macro, pull down the *Tools* menu and choose *Convert Macros to Visual Basic* from the *Macro* submenu. You will then find the converted code in the *Modules* area of Access.

Another way you can get going with VBA in Access is to use the wizards that are associated with the Toolbox that is used to develop a form or report. Assuming that you have an Access file that contains at least one table, you might create a new AutoForm based on your table. Then with your new form open, pull down the *View* menu and choose *Design View*. If the Toolbox does not show somewhere on the screen, pull down the *View* menu and choose *Toolbox*.

The top row of the Toolbox contains two buttons. The one that looks like a wand is an on-off button for the Toolbox wizard. With the wizard selected, hover over the Toolbox with your mouse, and find the *Command Button*. Click on it, and then move your cursor to a spot on the form where you would like the command button to be. The *Command Button Wizard* will appear, giving you numerous choices of what to do. After you have made your choices you can look at the code by right clicking on the button and choosing *Build Event*. In this case, the code for the button resides behind the form rather than in a module.

Should you go beyond the basics?

In this article, I have just barely touched the surface of the sphere of VBA. It can take a while to get up to speed. People who think analytically will find it easier than those who do not. Starting out from scratch it might take you six months of daily use to really get up to speed as a developer. On the other hand, if you just want to use it as automation for your own tasks you might be able to do a fair amount with just a little work.

Certainly, if you have a demanding job that would benefit from the type of automation that VBA provides, you might want to have a VBA expert do it for you.