

## The Five Minute VBA

By Mark Stefanchuk, Cadgurus.com

Developing a Visual Basic for Applications (VBA) tool is quite easy. That's mostly a true statement. On the surface the drag and drop interface development is outstanding.

Creating the interface for simple and even complex user input and visual queues is intuitive for the developer. The code that drives the application is the complicated part.

Each application that implements VBA, Word, Excel, AutoCAD, and now MicroStation all require special functions to manipulate the application's interface and data. It is definitely a powerful and extremely flexible development environment, but to create sophisticated applications requires the software management skills of a professional programmer.

Fortunately, for the CAD manager who has invested time to learn MicroStation BASIC Macros, or Visual Basic Script (VBS) learning to manipulate graphics and data using VBA is a simple extension of what they already know. And fortunately for the rest of us most of the code we need to write is as simple as setting line weight, color, style and level.

### Getting Help

Before getting started, you need to know where to look for examples and help. For VBA help there are two places.

The first is accessed under the help pull down menu from the VBA editor (*Utilities>Macro>Visual Basic Editor*). Use the VBA help to find information about VBA specific methods, objects, functions, and controls.

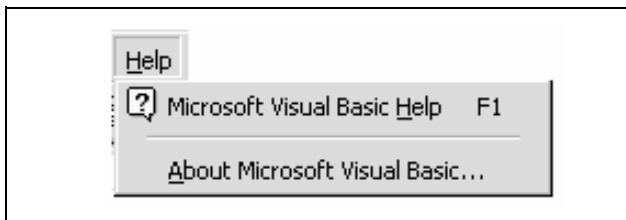


Figure 1 Opening VBA Help

The second is the MicroStation VBA help which is currently located in the MicroStation root directory. The file is called MicroStation VBA.chm. Use this help file to research MicroStation VBA methods, objects, functions, and controls.

The help files often are not the best place to start trying to solve a coding problem. If you need an example, or the correct syntax for a command, then by far the best method to use is a web search.

I prefer [www.google.com](http://www.google.com) and have recently started to use [www.google.com/ie](http://www.google.com/ie) for faster, trimmed down searches.

When you use a search engine include key words like VB, VB6, or VBS. An example of a search I made today to find the options for `CreateTextFile`: "VB CreateTextFile". The sixth result was a Beginner article describing how to use the file system object.

### Creating a MicroStation VBA

In this article you will learn how to create a simple MicroStation VBA (MVBA) program. The MVBA sets the active attributes for color, weight, style and level when a button is pressed on a User Form.

#### Tips

### Placing Text Along An Element in V8

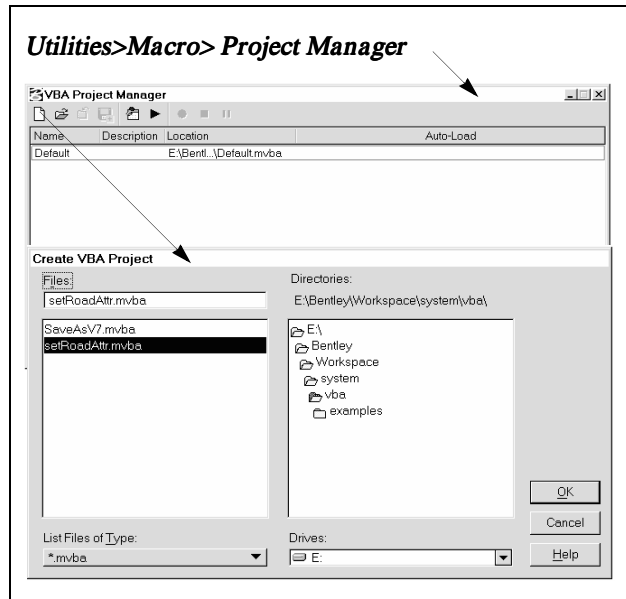
When I place text along elements, MicroStation J puts each letter, number, etc as an individual element. So when you edit text, you are picking individual elements.

-----  
The text along an element is working the way it is suppose to work in MS J. It places the text as individual characters and the text is a graphic group. The same command in V8 keeps the text as a text string.

CAD

So how do I create a useful MicroStation VBA and do it in five minutes?

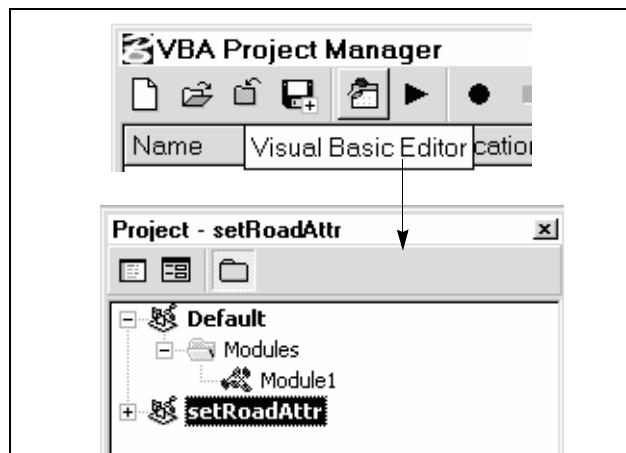
- The first step is to create a new MVBA project. In MicroStation, select *Utilities>Macro>Project Manager* to open the VBA Project Manager dialog box.
- Click the New Project icon to open the Create VBA Project dialog box.
- Type in the name *setRoadAttr.mvba*. The default directory is *...|Program Files|Bentley|Program|Workspace|System|VBA|*.



**Figure 2** Create a New Project

From the row of icons along the top of the Project Manager, select the Visual Basic Editor icon (the fifth button from the left). This is the command button which opens up the VBA editor.

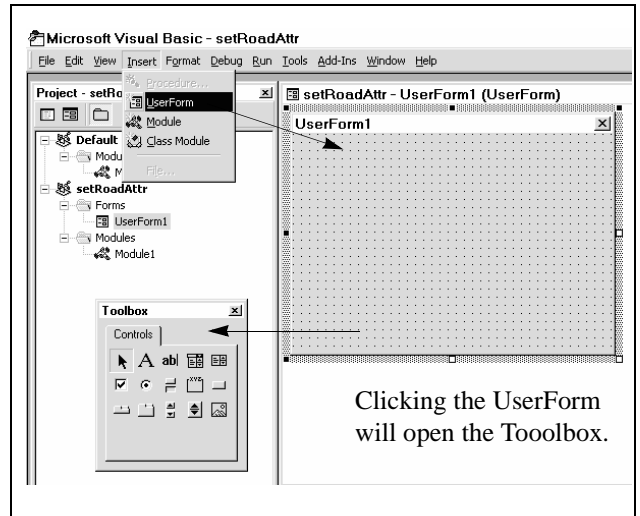
The editor is configured to show you a project file tree on the left side of the main window. There you will see a file tree that resembles A Windows Explore Form. It will show one file unexpanded. The file name is the name of the project you just created.



**Figure 3** Opening the VBA Editor

- At the top of the VBA editor window is a pull down menu called *Insert*. Select it and then the sub menu called *User Form*. When you do this a new form is added to the project.

Congratulations. You have just created your first dialog box. In VBA these are called User Forms.



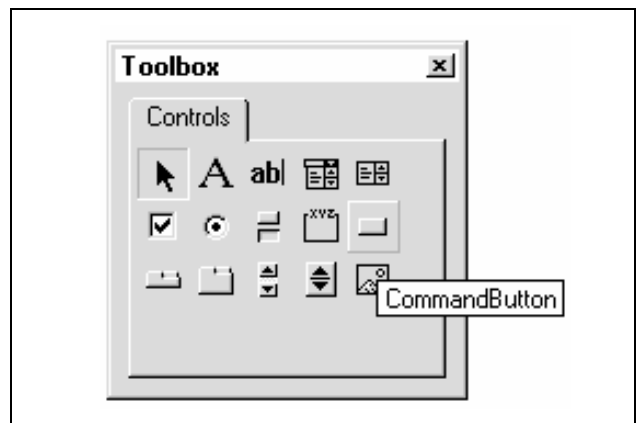
**Figure 4** VBA Editor with New User Form

Next, we will add a button control to the new form. When you inserted the form, several things happened to the interface.

The form was added, a second form called *Toolbox* opened, and a new docked window with a bunch of text in it was displayed. This last item describes the attributes of a VBA object and is referred to as the *Properties* Window.

Controls, like buttons, text boxes etc., and user forms are VBA objects. Other things can be objects too, but for our purposes this is a good enough definition.

- Click the *CommandButton* control button on the toolbox.



**Figure 5** VBA Controls Toolbox

- Draw a button on the user form. This is similar to drawing a box in MicroStation. Identify the first point and then the opposite corner. See Figure 6.

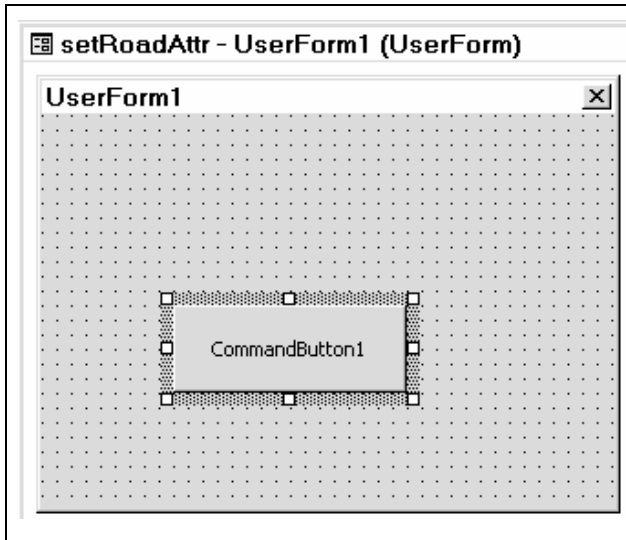


Figure 6 VBA Drawing a CommandButton

The CommandButton has the caption “Command Button1”.

In the Properties Window, the name of the property is displayed on the left hand side of the Window and the Property value is displayed to the right.

- Scroll down the Properties Window until you find the Property called “Caption”.
- Double click on the caption value, Command Button1. This will highlight the text. Type a new name. Use the name, “Set Road Attributes”.

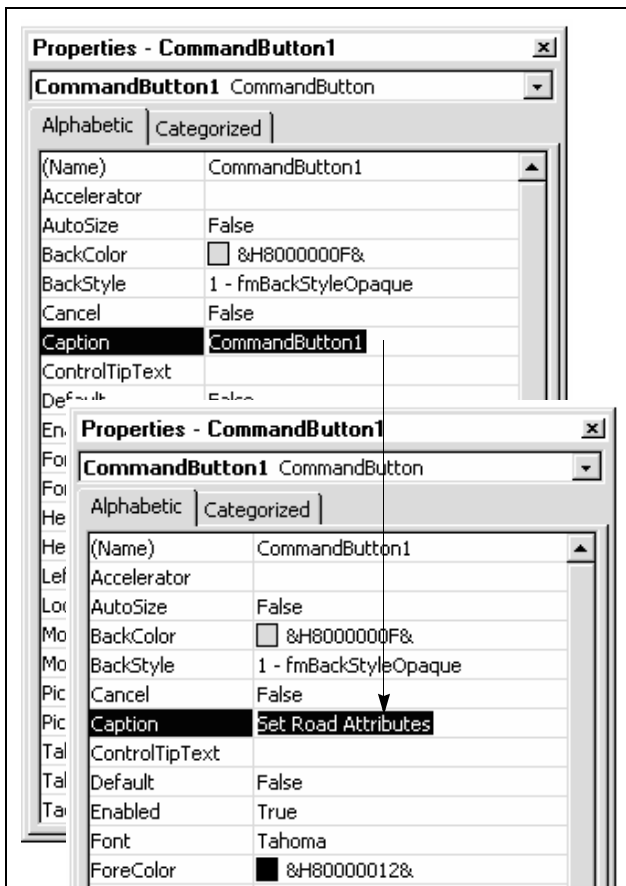


Figure 7 VBA Rename the button

As you type the new name the new text is displayed on the button. You can give the form a new name too. It’s very easy, so I will let you figure it out on your own.

- Also, scroll down to the property ShowModal and change the value to False.

This value will allow you to do other things, like draw a line, while the form is still displayed.

We need to do one more thing to complete the user form.

- Double click on the button. A new Window opens. It looks similar to notepad, but it will have text in it.

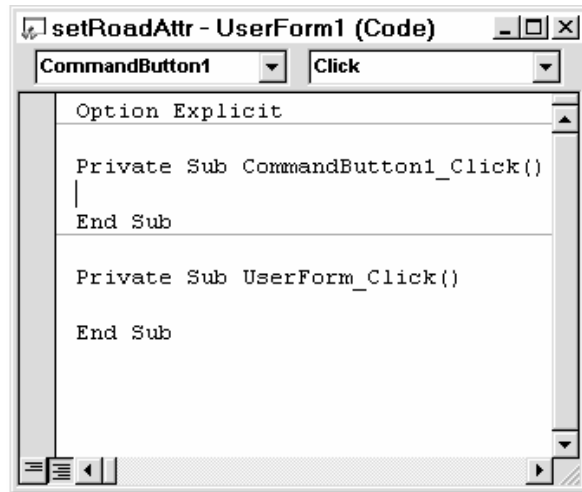


Figure 8 VBA Code

This is VBA code. It is created for you automatically when you double clicked the button. The first line, Option Explicit, tells VBA that all variables must be declared.

If you want to save the number of times something happens you might use a variable called count. The variable must be defined. The new code would look something like this.

```
Option Explicit
Private Sub CommandButton1_Click()
Dim count as Integer
End Sub
```

The use of Option Explicit is an option. If you don’t use Option Explicit, and you don’t declare the variable, VBA will assume the variable type is “Variant”. This can lead to problems when the code becomes more complicated, but for now just remember to declare your variables and use Option Explicit.

To turn **On** this option, go to the VBA pull down menu *Tools>Options* and on the Editor tab, check the box next to Require Variable Declaration.

The rest of the code displayed is just a container for the actions we want the button to perform. Add the following code between Private Sub CommandButton1\_Click() and End Sub.

It should look like this when you are done.

```
Option Explicit
Private Sub CommandButton1_Click()
ActiveSettings.Color = 3
ActiveSettings.LineWeight = 4
ActiveSettings.LineStyle = 0
ActiveSettings.Level =
    ActiveDesignFile.Levels("ROAD")
End Sub
```

Make sure that assignments remain on the same line. The following for instance,

```
ActiveSettings.Level =
    ActiveDesignFile.Levels("ROAD")
```

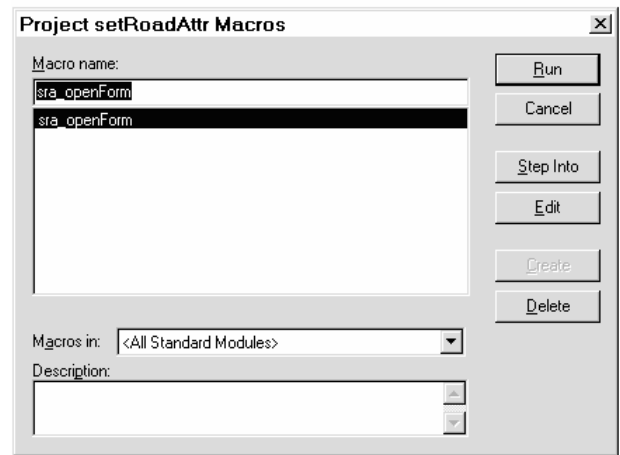
Must be one a single line.

There is one more step before we test the new program. We have to tell MicroStation how to start and load the form.

- In the Project window, under setRoadAttr is a folder called Modules. Expand it and double click on the file called Module1.
- A new editor window will open. Type in the following lines of code.

```
Option Explicit
Sub sra_openForm()
Load UserForm1
UserForm1.Show
End Sub
```

- Save your program, *File>Save*. Close the VBA editor, *File>Close + Return* to MicroStation.
- In the VBA Project Manager dialog box, highlight and double click on the project name setRoadAttr. A new form opens displaying a list of macros.
- Double click on sra\_openForm.
- Click your button. The attributes change to the values you set in your code.



**Figure 9** VBA Project Macros

## Summary

With a little practice you will wonder why it took you even five minutes to create this MVBA. Source code and download for this example are available in the **cadgurus.com** download archives. Look for the title, "Five Minute VBA".

## About The Author

*Mark Stefanchuk is a partner with Ramsey Systems, Inc., the developers of cadgurus.com. Mark can be contacted by email on mark@cadgurus.com.*

*Please email Mark with any feedback or suggestions for future articles.*

**CAD**

### Tips

## Pasting Text

How can I paste text from the clipboard to the key-in window without it trying to do a place text element?

-----  
Select SHIFT+INSERT

**CAD**

