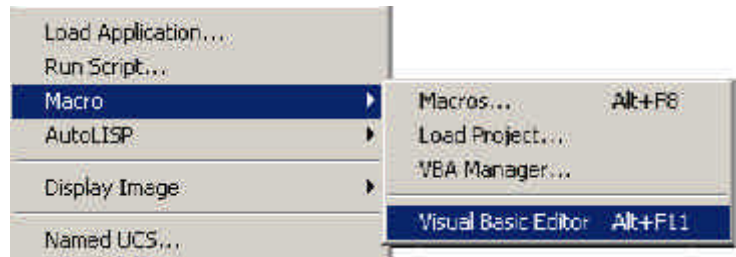


VBA Foundations, part 3

A Tutorial in VBA for Beginners—The Third in a Twelve Part Series

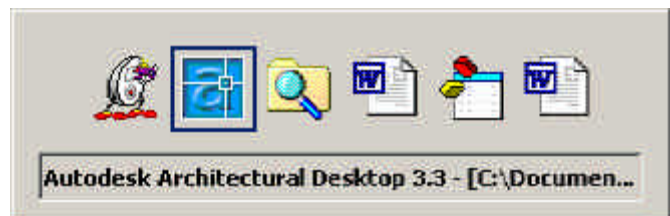
Richard L. Binning / rbinning@mediaone.net

Lets get started with some code by working with the Visual Basic Editor (VBE). Like most of your tasks in AutoCAD, there are many ways to access the VBE. You could hold the “alt” key down and press the “F11” key. You could also select “Tools” from the AutoCAD menu bar, select “Macros” and click on “Visual Basic Editor” as shown in the graphic on the right. All you “salty” veterans can type “vbaide” at the command line if you just can’t give up the command line editor. Any of the above actions will launch the VBE environment. Remember the screen shot of the Visual Basic Editor shown in the last issue? When this set of windows is launched for the first time during an AutoCAD session, it may or may not have projects loaded. Most likely you will see the default Global project displayed in the “Project Explorer” window. This project will be created whenever there are no projects currently loaded. You may notice that this file is named such as “Global#” with the pound sign replaced by a number. This number represents the number of macros previously loaded during the current session of AutoCAD. Whenever a new project is displayed the default class, “ThisDrawing”, will be created. Don’t worry if you don’t know what a class is or does, just remember that you have been utilizing classes since the first bit of code you entered in VBA. We’ll cover classes in-depth later after you have some code under your belt. When we discuss classes you’ll be surprised and relieved to discover how much you have been utilizing “classes” since the beginning without even fully understanding them and they won’t seem as daunting as they might now.



I mentioned in the previous paragraph that a new project is often created by default when you open the editor. If you already had a project loaded when you opened the editor you might have noticed that unlike most windows programs a quick glance at the options available under the “File” menu fail to uncover either of the following two typical Windows commands: “Open” or “New”. You will be unable to create, unload, or open a new project from within the editor itself. You can save, however, and I recommend that you make saving your projects frequently your second nature.

a new project can be created through the VBA Manager dialog. To access this dialog you must return “the focus” to AutoCAD’s window by clicking on AutoCAD’s window or closing the VBE. Alternatively, you can select the AutoCAD “a” icon just below the File Menu (notice the trademark “a” that begins this paragraph), which will minimize the editor and return focus to AutoCAD’s window. You could also hold the Alt key down and press the Tab key until you see the AutoCAD icon selected as shown in the graphic on the right. (Note this method is a standard windows interface method and enables you to switch between any of the currently open applications.)



Before we continue, allow me to explain this concept of focus I keep referring to. Not only is it an integral part of working in any version of Microsoft Windows, but understanding focus is a truly vital aspect of working with the Visual Basic Editor. The key to knowing when a window has focus is the color of that window’s Main Title Bar. When a window has focus it will turn a brighter or darker color. Notice that all

(Continued on page 4)

VBA Foundations, part 3

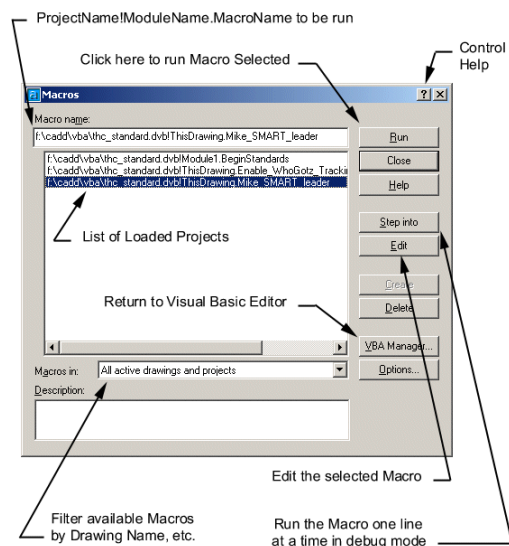
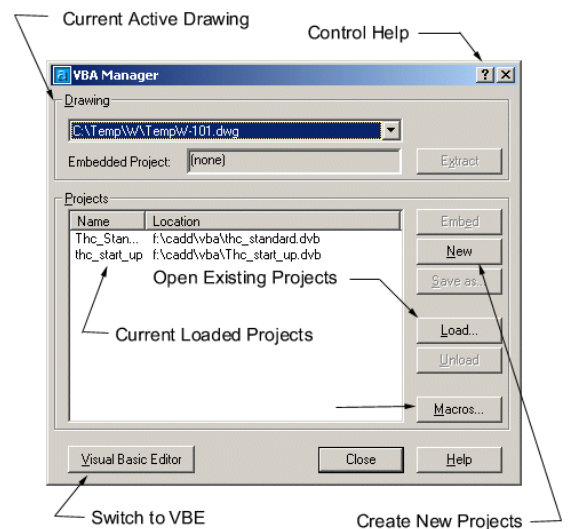
A Tutorial in VBA for Beginners—The Third in a Twelve Part Series

Richard L. Binning / rbinning@mediaone.net

other windows that do not have focus will have a much more subdued or faded color displayed in their main title bar as shown in the example below.



Now that you have a better understanding of the concept of focus, lets return to our new project creation. After you have returned focus to the AutoCAD window you can open the VBA Manager by selecting Tools>Macros>VBA Manager from the AutoCAD menu (“Old Salts” may proceed to type “vbaman” at the command line). Now select the New button as shown in the VBA Manager dialog box. You’ll utilize this dialog box frequently in the coming months, so lets explore it now. While we are exploring the VBA Manager Dialog let me take this opportunity to show you a built-in tool which can guide you as you discover and explore this dialog and the rest of the VBE now and in the future. Look at the top right of the VBA Manager dialog just to the left of the small button with the x. When you click on the button with the question mark, notice that a large bold question mark now appears anchored beside the cursor and moves with the cursor wherever it goes. Now direct your cursor to some area, label, button, or control on the dialog box and click your mouse. Do you see a small help description appear adjacent to the item you clicked on? Anytime you are working in the VBE you can follow this process to refresh your memory on what a particular control or area is used for. Additional controls available on the VBA Manager are shown in the graphic above right. Using the button on the lower left, you can open up the VBE and click the help button at the lower right to call up general Microsoft help with regards to VBA. (Note: this is generic VBA help not AutoCAD specific VBA help)



Clicking on the “Macros” button will access the Macros dialog, shown on the left, where you can select a project and run the available macros found within. You can also run a macro and begin stepping through your code one line at a time in “debug” mode. If the macro you wish to run is not listed in the Macros dialog then you may click on the button at the bottom right, which is labeled “VBA Manager...” to return to the previous dialog box shown above. You can also access the previously mentioned “control help” to jog your memory, sort the list of macros shown in the main list window by showing only a filtered list of macros available in all drawings, all projects, active drawings or projects, or the default as shown of “All active drawings and projects”. Since the macro name is listed at the top of the graphic to your left, please pay particular attention to the format shown in calling a particular macro contained inside

(Continued on page 5)

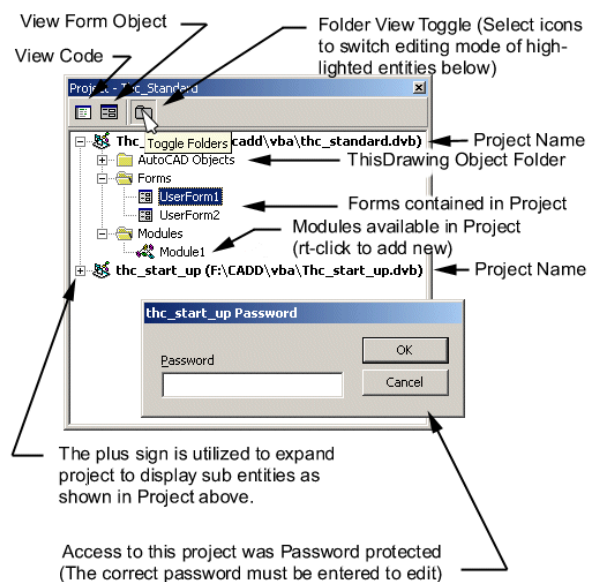
VBA Foundations, part 3

A Tutorial in VBA for Beginners—The Third in a Twelve Part Series

Richard L. Binning / rbinning@mediaone.net

a project. This format for calling individual macros or subs will be utilized in VBA and in AutoCAD custom menus. A similar construction is used in visual lisp when we are creating hot keys or two letter shortcuts for running our macros.

While working in the VBE you will need to become intimate with the Project Explorer Window. This window, shown to the right, is utilized much like the familiar Microsoft Windows Explorer. It provides the interface allowing us to manage, add, and delete all the parts and pieces (forms, modules, etc.) of the project we are editing. It is manipulated in standard the Windows Interface fashion like most of the “explorer dialogs” you will encounter such as Windows Explorer, AutoCAD’s Properties Dialog, AutoCAD Architectural Desktop Style manager, etc. In it each category can be expanded or minimized by double-clicking the name shown or clicking the plus sign to the left of the category. There are also many additional short keys/hot keys for accessing this functionality. For example: when a form’s name is highlighted in the Project Explorer you can select the Code View icon located at the top left of this dialog to access the code behind the form or hold the “shift” key and press enter to access the code behind the form. To directly access a particular portion of the code related to a control, simply double click the control desired shown on the form to access the particular code behind it.



If you would like an extensive spreadsheet detailing all the short cut / hot keys available for use in the VBE, please send me an email with the word “VBEKEY” in the subject line.

In this issue we have explored the Visual Basic Editor in depth and learned how to navigate through it, expose different properties and objects in our projects, load, create, delete, and close our projects. We have learned that we can password protect our projects, use hot keys for rapid access to entities contained within our projects, and came to a better understanding of the concept of “focus” and how it is used throughout Windows programs and the VBE in particular. In our next installment we will cover key Visual Basic concepts such as the anatomy of a module, begin writing simple macros, learn about some of the enhancements contained in the code editor windows, and explore “IntelliSense” which can save time, eliminate spelling errors, and reduce the amount of typing required while inputting your code. Stay tuned to begin your active participation in this learning adventure.