Taking Macros to the People: Hands-On For Beginners

Presented by Keith Rice, Technical Director, CADSharp LLC

View this presentation and more free API training at www.cadsharp.com

The Macro Toolbar

All of the macro commands are found under Tools→Macro, though you are encouraged to use the toolbar.

The Macro Recorder

The macro recorder is a tool built into SolidWorks that lets you record yourself performing certain actions within SolidWorks. Once you save the macro, you can replay it, and the steps you recorded should be repeated. To use the macro recorder, use the "Record" button on the macro toolbar. Use "stop" to stop recording and "play" to replay the recording. *Make sure that you begin with the same initial conditions as when you recorded the macro!*

The macro recorder can help you quickly determine what API calls (functions) are needed to accomplish a task, but it also has four big disadvantages: 1) it creates sloppy code, 2) it cannot properly record many actions, 3) it cannot handle user input, and 4) it cannot handle logic. If you do use the macro recorder, I recommend that you record in small chunks and then splice those pieces of code together.

Cleaning Up the Macro Recorder's Code

The macro recorder will produce poorly formatted code, much of which is unnecessary. Use the single quote to comment out lines in order to determine if a line of code is necessary.

Within the sub procedure (the code between "Sub main()" and "End Sub", it is helpful to indent in your code. You can also use the line continuation character, the underscore, to continue statements onto the next line.

VBA Programming Basics

In VBA programming, we are almost always interacting with a piece of information stored in a variable. This involves three steps:

1. Declaring the variable
    a. This is when we reserve a piece of memory for that variable.
    b. Variables come in different shapes in sizes depending on what type of information need to be stored
    c. Think of a variable as a container that we need to reserve for later use.
2. Assign a value or reference to the variable
    a. This is when we put something in the container.
3. Do something with the variable
    a. Use string and numerical operators
    b. If the variable is an object, use its members (which you might hear called "methods", "properties", "functions", or "API calls") to do something

Variables

Declaring a variable takes this form: Dim <variable name> As <data type>

- You choose the variable name. You cannot use a reserved (blue) keyword, however
- Variables can be created inside or outside of sub-procedures
- The following is a list of the variables most commonly used in VBA

Name	Used for…	Size	Range (approximate)	Default value
Integer	Small integers	2 bytes	-30,000 to 30,000	0
Double	#'s with decimal values	8 bytes	See API Help	0
String	Text	Varies	0 to 2 billion	"" (Empty)
Boolean	True or False values	2 bytes	True or False	False
Object	Object reference	4 bytes	N/A	Nothing
Variant	Anything	Varies	0 to 2 billion	Empty

- Good programming practice: put a prefix in front of your variable names so you know its data type as soon as you see the variable. For example, a string containing a name would be called "strName"
- Anything that cannot be described as a number or string will most likely be considered an object. Every "thing" in SolidWorks is considered an object—the application itself, a part document, a face, a vertex, the FeatureManager tree, a drawing view, etc.
- Objects must be assigned references using the Set keyword.

Researching API Calls

API programming is probably 5% typing out code, and the rest of the time is spent researching API calls or debugging. Here some tips for finding the SolidWorks API call you need:

1. Try recording yourself performing the task using the macro recorder, then look at the code to see which function was used
2. In the API Help, use the Index tab and type in a category. For example, typing in "Drawing views" will yield all kinds of topical help for working with drawing views. On some member pages you might find some helpful examples.

The Next Step: From Amateur to "Dangerous"

This hands-on class was a very basic introduction to the SolidWorks API. What you learned here will actually not take you very far, so look at it as dipping your toe in the API waters. If you want to take your API skills to a serious level, then you will need to understand how the SolidWorks API object model works—in other words, how the object interfaces relate to one another and how they are accessed. It takes a little time and practice, but for anyone with serious automation ambitions, it is well worth it.

Get immediate access to over 15 hours of beginner friendly, in-depth API training at www.cadsharp.com.