

June, 1999

## Advisor Answers

### Properties as Private Variables

VFP 3.0/5.0/6.0

Q: I'm using a listbox with a RowSourceType of 5-Array. I finally figured out that I had to make the array (aComments) a property of the form and then use THISFORM.aComments for the RowSource. But I don't really understand why it works this way.

Why does one have to assign an array property to a form? Suppose I want to access this same array at some other point in the application? If an array is a variable, how can it also be a property?

–Quincy Howe (via CompuServe)

A: You've asked a big philosophical question that really points out the big difference between FoxPro 2.x and Visual FoxPro and, more generally, between procedural and object-oriented programming.

In FP2.x, when you generated an .SPR from an .SCX, anything in the Setup snippet became a "main" program of sorts for the screen. That is, the code there was at a higher calling level than any of the other snippets for that screen. So when you ran the screen, private variables created in Setup were scoped to the whole .SPR and, therefore, were visible in the other routines contained in the screen (such as Valid code for the screen's controls).

In VFP, there is no "main" program for a form. Load and Init are methods (triggered by events) just like all the others. So there's nowhere to declare variables that are visible to all methods. Instead, you use properties. Think of a property as a private variable scoped to the object. The only difference is that you have to identify the scope when you use it, so instead of referring to a property as cSomething, you need to say THIS.cSomething or THISFORM.cSomething (or whatever is appropriate to the situation) to indicate which object the "variable" is scoped to.

If you want the same array to be accessible elsewhere in your application, scoping it to the form isn't the right answer. It needs a higher scope than that. You could create the array in the program that calls the form, or in your main program, or make it a property of your application object (in which case you have to use the appropriate object reference, something like oApp.aComments).

There are some consequences of doing it this way, though. One of the benefits of OOP is encapsulation, having data and the code that manipulates it stored in one place. When you make the array a property of the form, you can be sure that when you run the form, the array is available. If the array is created elsewhere, you have to be sure it's available before you run the form. (We had this problem in 2.x, too. There, the issue was using variables from other programs without passing them as parameters.) In most cases, you don't want one form or program to depend on having another one run first.

In fact, in your situation, the best thing to do is create a listbox subclass and add the property aComments there. Set RowSource for the subclass to THIS.aComments. Then, when you use the listbox class in a form, make sure to add code to populate the array.

-Tamar