

September, 2006

## Advisor Answers

### A better New Property dialog

VFP 9

Q: Now that we can tell VFP to remember the capitalization of custom properties and methods using `_MemberData`, is there any way to have it pick up that information when I create the property, instead of having to go into the MemberData Editor every time?

A: Another new feature of VFP 9 means you're in luck. One of the more intriguing changes to the product is the ability to replace items on the VFP menu with our own code. You can use the IntelliSense table to indicate that when a particular menu item is chosen, something other than VFP's default behavior occurs. This mechanism uses a script called MENUHIT to divert the menu behavior.

I'll leave the particulars of implementing such a script for another time because the Fox team (with the help of Doug Hennig of Stonefield Systems Group) already took care of your particular problem. The Solution Samples for VFP 9 include a replacement Add Property/Method dialog that not only lets you specify capitalization when you add a property or method, but takes care of a bunch of other annoyances in the native dialogs.

To use the replacement dialog, open the Solution Samples. (The easiest way to do this is to open the Task Pane Manager -- Tools > Task Pane -- and click the Solution Samples button.) Then, expand the New in Visual FoxPro 9.0 group and scroll down to find the Foxcode menu scripts items. Run it, and the form shown in figure 1 opens.

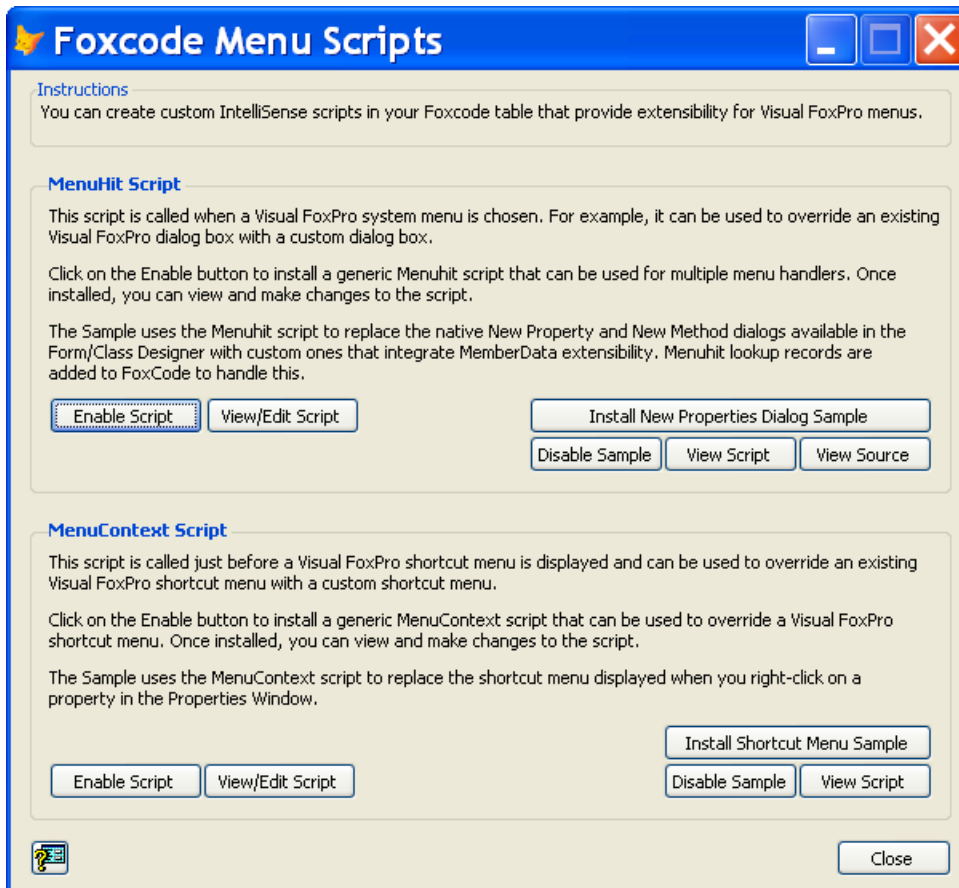


Figure 1: Customizing VFP menus—This Solution Sample lets you install a replacement for the New Property and New Method dialogs.

First, click on the Enable Script button to add a record to your IntelliSense table that allows custom menu scripts. Then, click on Install New Properties Dialog Sample to add additional records that are called when you choose New Property or New Method from the Form or Class menu. Close the form and exit the Solution Samples.

From here on out, the menu items Form > New Property..., Form > New Method..., Class > New Property... and Class > New Method... will all call the same form. When you choose New Property from the menu, the form looks like figure 2. If you choose New Method, it instead looks like figure 3. However, you can easily switch between the two by choosing the appropriate option button at the bottom of the form, a nice feature if, like me, you often choose the wrong menu item.

Figure 2: A better dialog—This replacement for both the New Property and New Method dialogs is non-modal, handles both properties and methods, and sets up `_MemberData` for capitalization.

Figure 3: Adding methods—When you choose New Method from the menu, the form is prepared for methods.

A feature that's not so obvious is that the new form is not modal. That is, you can keep it open while you work, which is convenient if you're not sure what properties and methods you'll need.

The form also lets you add a new property or method to the Favorites tab when you create it, again saving you from having to remember to do so afterward.

There are two things you need to be aware of if you make the switch to the new dialog. First, it is called only from the menu. If you open the Edit Property/Method dialog then click New Property or New Method from there, you get the native dialog. (You could, of course, build your own replacement for the Edit Property/Method dialog that calls this New Property/New Method dialog, but that's definitely beyond the scope of this column.)

Finally, the replacement dialog doesn't work for adding methods to ActiveX control classes. If you need to do that, you can go through the Edit Property/Method dialog to get to the native New Method dialog. This bug is fixed in an updated version of the dialog, available from the VFPX (formerly known as the SednaX) project. By the time you read this, you should be able to download it from <http://www.codeplex.com/Wiki/View.aspx?ProjectName=VFPX> .

I use the new dialog for my daily development. When I have to work in an earlier version of VFP without it, I find I miss it a lot.

-Tamar