

Visual FoxPro 5.0: A Sneak Preview

Whil Hentzen

As most of you know, the next version of Visual FoxPro has been in beta testing for a number of months. I'm finally allowed to talk about it, so this month I'll give you a sneak preview of what you'll be using in a few months.

Most *FoxTalk* readers won't get their hands on the new release for a few more months, so I'm not going to spend this entire issue flooding you with information that you can't use for a while. Nonetheless, you're curious and want information that can help you plan for the future, so I'll do a preview in this month's editorial and then cover these new features over the next months—and when Visual FoxPro 5.0 hits the street, we'll get more detailed.

First of all, you might be asking, "5.0? Did I fall asleep? What happened to 4.0?" Nope, the version of VFP that follows 3.0 is going to be named 5.0—just as Microsoft Word jumped from version 2.0 to 6.0 a couple years ago. Going from 3.0 to 5.0 will keep VFP's numbering in sync with the next releases of Visual Basic 5.0 and Visual C++ 5.0 (both of which are due out "real soon now"). That, of course, brings to mind two questions: "First, why is it important to keep version numbers in sync?" And, from the more thoughtful of you: "Will 5.0 suck up significantly more resources like Word did when it went to version 6.0 from 2.0?"

Let's address the second question first: No, 5.0 won't require more resources—in fact, one of the primary design goals for version 5 is improved performance—especially in areas that affect the user interface, such as form instantiation, and in situations with 8M and 12M of RAM as well. As far as the version numbers are concerned, remember that all three tools, Visual C, Visual Basic, and Visual FoxPro, are going to be merged into a common development environment called the Developer's Studio. We can look forward to concurrent releases of these products in the future, so it will be easier to keep them straight if they're numbered similarly. Furthermore, as each of these tools begins to use the same components, such as the Debugger or Report Writer, it will again be easier to keep the revisions straight.

Now to the \$64,000 question—what's new in VFP 5.0? The fundamental goal of version 5.0 is to provide the developer tools that we need in order to keep up with the amazing improvements in the FoxPro language. When you fire up 5.0, the first thing you'll see is that the editor has been colorized—keywords, variables, comments, and the like all have their own colors—and yes, you can tailor those colors to suit your preferences. If you think quickly, the second thing you'll notice is that by right-clicking you'll get context menu that extend the capabilities of the editor—the ones I like the most are the ability to clear the Command Window and the ability to execute the highlighted expression.

Touring around the interface some more will bring you to the Debugger, which is a whole new animal. If you've seen the Debugger in Visual Basic or Visual C++, you'll be reasonably at home. It's a completely separate process from Visual FoxPro, which means that it's a bit disconcerting to see the entire window disappear behind the FoxPro window. Then you'll realize that this means that using the Debugger is cleaner now, since it doesn't participate in any FoxPro events.

The Debug window actually has five pieces: Trace, Watch, Locals, Call Stack, and Output. You can select which of these components are displayed, and they can all be positioned at will. You'll soon find that the 17-in. monitor that you've been eyeing is now a requirement—and an even larger one might not be that bad of an idea. Besides the obvious capabilities that these windows give you, there are three more functions that I really like. Event Tracking allows you to send the results of events you specify (such as particular methods) either to the Output window or to a text file. Coverage logging allows you to see what code is being executed. You can also save a

particular Debugging configuration to a file to save time when setting up your development environment. (Remember what a pain it was, having to type in variable and variable in Debug, only to lose them if you had to exit out of FoxPro?) OK, and now that your appetite's been whetted, we now have assertions in the language!

The next great feature that everyone wanted has to do with the way that classes are handled. For instance, if you drag fields from the Data Environment to a form, you've been stuck with the Visual FoxPro base classes—there hasn't been a way, short of writing a builder, to tell VFP to use your own classes instead. In version 5.0, however, the new default class and library properties will allow you to assign a control type to a field: when you add the field to a form, you'll create the exact control you want in one easy step.

The database container has been enhanced as well. For example, it will allow multiple users to simultaneously create and modify objects within the same database, and the engine now supports changes to values that violate rules. Both changes will solve a lot of headaches for you.

Probably the most requested feature for this version of Visual FoxPro has been outer joins, and you won't be disappointed. You can create a variety of outer joins, queries that return "Top N" results, and alias columns, and you can do this all in the Query and View Designers to boot.

The Form Designer has been enhanced as well. The Properties window has a ton of new interface capabilities, including Property Zoom and keyboard navigation. There are additional properties, events, and methods with many controls, and there's a whole slew of new OLE controls as well. Other interface goodies include the ability to flip back and forth between Run and Design modes with a click on the toolbar, the ability to manipulate multiple controls at one time, and additional alignment capabilities.

You'll now be able to create SDI applications so that your application windows can be children of the Windows desktop. And you'll be able to create context menus with the Shortcut Menu Designer (your controls will be able to respond correctly to their new RightClick event. And for those of you who've completely immersed themselves into the Windows environment, you can create OLE servers with Visual FoxPro and other applications will be able to query FoxPro just as Fox has been able to access other applications such as Word and Excel.

Remember that this was just a brief tour. I could have spent several pages (instead of one paragraph) on the Debugger alone—but it should give you a taste of what's going to be in your hands in short order. You think VFP 3.0 was exciting? With 5.0, it'll just get better.