

July, 2000

Editor's View

Tools Make it Easier

But We Have to Make the Time to Keep Learning

by Tamar E. Granor, Technical Editor

I spent several days recently on site at a new client, a development shop in the process of moving their flagship products to VFP. They brought me in to answer questions, teach them some things and generally give them an assessment of how they're doing.

As I reviewed their questions ahead of time and prepared for my visit, I was struck by both the quality and quantity of tools available to VFP developers. My client's questions ranged from source control to installation to reporting and I found myself jumping from website to website, collecting information on various products, downloading demos and pasting bookmarks into a document that we could revisit on site. While on site, the list continued to grow. We'd be discussing a topic and I'd remember a tool I'd used or heard about. Fortunately, this company is very open to the idea of using outside tools.

Not every company has that attitude. Two recent online incidents brought that home to me. In one case, a fellow (still using VFP 5) was looking for a way to retrieve the version number of an .EXE file. FoxTools.FLL, included with VFP, makes that a breeze—all it takes is a single function call, but he didn't want to distribute FoxTools with his application. Instead, he preferred to write his own version, making the necessary API calls. (In VFP 6, retrieving the version of an .EXE is even easier – the `AGetFileVersion()` function hands you the information you need without the need for any outside libraries.)

The second incident involved a developer forced to cut-and-paste useful code from a freeware tool and pretend it was his own because his boss insisted that everything be developed in-house. The boss believed that he retained more control that way and couldn't see how much money such an attitude was costing his company.

My new clients already own and use several of the most widely recommended tools for VFP developers. Their need is to learn to take full advantage of the tools they have. It's very easy to learn just the minimum functionality of a tool and never take it any farther. The

challenge for all of us is to find the time to go beyond that and let the tools do what they're designed for.

Making time to keep learning may be the biggest challenge we face as developers, in fact. But unless we do so, the one we hurt the most is ourself. Pamela Thalacker, who'll be joining the Advisor Answers team starting next month, put it this way, "I would just like to say from my experience, every five minutes I spend reading good instructional material saves me an hour of development time. If I am in a development crunch then I don't have time *not* to read."

Pamela's absolutely right; the hard part is having the discipline to follow her advice. I advised my client to make regular seminars a part of their schedule. I'm trying hard to follow my own advice and make reading new material part of my schedule, as well.

What about those tools?

While I'm a big believer that buying the right tools is extremely cost-effective, there are also plenty of free tools for VFP developers. A number of them come with the product; others are available either public domain or freeware.

Let's start with the goodies that Microsoft includes with Visual FoxPro. I won't discuss the ones on the menu, like the Class Browser, Component Gallery, Debugger and Coverage Profiler, except to say that all of them deserve serious study. They're far more capable than they appear at first glance.

But VFP comes with some tools that are hidden from plain sight. If you don't know they're out there, you may not find them. Probably the best known among this group is the FoxTools library, which includes a number of useful functions. Although the Tools folder contains FoxTools.CHM (in VFP 6; look for FoxTools.HLP in older versions), not every function in FoxTools is documented there. ToolHelp.Hlp, compiled by FoxPro Advisor writer George Tasker, documents every function in FoxTools – you'll find it (zipped) on this month's Professional Resource CD and at Advisor.COM.

Most of the goodies are in sub-directories of the Tools directory. Run GenDBC against an open database to create a program that, when run, recreates the original database (without the data, of course). Use it to document your databases.

Many of us lamented when Filer was removed from VFP; our cries were heard. VFP 6 includes Filer.DLL and a form, also called Filer, that together reproduce much of the functionality of the old Filer applet.

While I won't look at them here, don't forget about the FoxPro Foundation classes and all the sample code to be mined in the Solutions samples.

Finally, let me mention one tool that's included with VFP 6, but is available elsewhere in an expanded version. When you're working in the method editing window, Ken Levy's SuperCls toolbar lets you see the code for the same method of the parent class of the form or class you're editing. The toolbar appears automatically whenever you open the Class Browser and you can find the source code for it in XSource.ZIP in the Tools\XSource. However, I've put SuperCls.ZIP on this month's PRD and Advisor.COM. This version includes some additional tools and has instructions so you can make the toolbar appear when you start VFP.

The next time I write this column, I'll talk about free tools written by others. There's a long tradition of this kind of help in the FoxPro community. I've already accumulated quite a list of them, but if you have a favorite you think should be included, drop me a note at foxpro@advisor.com. Be sure to tell me where to find the tool.