April, 2003

Editor's View

Time to Upgrade

VFP 8 offers an incredible mix of new functionality and productivity tools

Although we've been writing about Visual FoxPro 8 in FoxPro Advisor for several months, this month marks our official VFP 8 launch issue. All the feature articles in this issue look at new features of VFP 8.

VFP 8 was released to manufacturing at the end of January and became available to MSDN subscribers less than a week later. By the time you read this issue, it should be available through retail channels, as should the book Doug Hennig and I wrote to help you get up to speed on this version, *What's New in Visual FoxPro 8* (www.hentzenwerke.com).

Whatever version of VFP you're using, this is the time you should upgrade, for both technical and political reasons. I've written in this space already (8/2002 and 12/2002) about new features in VFP 8. So, rather than reeling off a feature list again, I want to focus on what VFP 8 offers that will change the way you work.

Changing the Way You Write Code

The first area that's really different in this version is dealing with data, especially remote data. The new CursorAdapter class lets you work with ADO, ODBC, XML and native VFP data all the same way. You can design and test an application with native data, then change a couple of properties and work with remote data. Same thing if you suddenly need to switch from ODBC to XML or ADO. CursorAdapters give you the best of remote views and SQL Pass-through and eliminate some of the problems of those approaches to client-server work.

The next big area you'll want to rethink is error handling. The introduction of structured error handling with the TRY-CATCH-FINALLY syntax lets you eliminate tons of nested IFs and CASE statements. This structure offers local solutions to local problems, and more global solutions to global problems. Even better, it results in more readable code. (For details, see Christof Lange's entry in the February ADVISOR Answers column.)

The ability to bind code to VFP events and to fire events offers new solutions to some old problems. The BindEvent() function lets you connect code to events dynamically, so for example, you can hook a progress control into an existing process without having to touch the original process's code. Similarly, you can fire code for events occurring in objects for which you don't have source.

Dealing with certain container classes, such as pageframes and grids, has always been a problem because they always used the appropriate baseclass for their contents. VFP 8 lets you indicate what page class is used in a pageframe, what column and header classes are used in grids, what data environment class is used in a form, and so forth. This means you can design things as they should be without having to work around artificial limits.

Changing the Way You Work

VFP 8's new tools change things, too. The Toolbox replaces the Form Controls toolbar, but you can use it not only to put controls on forms, but to manage web services, pieces of text, and other files you use frequently. (See Cindy Winegarden's articles in the January issue and this issue for more information.)

Now that we have Code References, it's hard to imagine how we ever lived without it. This new tool lets you search for and even replace strings or regular expressions in a project. I've used it extensively to examine the code for the new tools. It also offers the ability to quickly find the definition of an item. (For more information, see my ADVISOR Answers entry and Andrew MacNeill's article, both in the March FoxPro Advisor.)

The Task Pane Manager may just change the whole way you approach the development environment. It comes with built-in panes for a number of common tasks and can be extended in all kinds of ways. Expect third-party vendors for VFP to provide task panes for their products. See my articles in the February issue and this issue for details.

There are also significant enhancements to several of the existing tools. The Query/View Designer has the most changes. For the first time, it's reliable enough to handle almost any query you can throw at it. Best of all, it now features two-way editing, so you can change the code it generates directly.

There are a number of usability changes to the Form Designer and Class Designer, including a new View Parent Code button. Even the Report Designer has some changes, including a new _PAGETOTAL variable that makes the “Page x of y” formulation a breeze.

Supporting Future Development

If all these changes and the dozens I’m not writing about here aren’t enough to get you to upgrade, consider the future of VFP. Microsoft has been open about the fact that VFP 7 sales were disappointing. The FoxPro community needs to make it clear that we’re still enthusiastic about this product and want them to continue to enrich it. We can do that by buying VFP 8.

Another way you can support the VFP team is by sharing your successes with the world. Microsoft offers a template for writing case studies at http://www.gotdotnet.com/team/vfp/. The best will be posted at Microsoft.com.

I’m more excited about VFP 8 than any new version since the transition to object-orientation in VFP 3. This version is solid and offers an exciting mix of new functionality and tools that let us create better apps more easily. If you’ve worked with the public beta, you know what’s in store for you; if you haven’t, I think you’ll be quite pleased.

Microsoft says that 80% of the changes in VFP 8 came from community requests. You asked for it, you got it, now support it.