

May, 2001

Editor's View

Not in the .NET

Microsoft separates VFP from Visual Studio.NET

Microsoft announced at the end of February that Visual FoxPro 7 will not be part of Visual Studio.NET, but will be sold and marketed as an independent product. (You can see the announcement at <http://msdn.microsoft.com/vfoxpro/prodinfo/vfp7.asp>.) If you haven't been part of the ongoing discussions about this subject, you're probably a little stunned right now and trying to figure out what this means for the future. Let me talk about the pros and cons of this change, and then offer my overall feelings.

First, why remove VFP from Visual Studio.NET? One reason is that VFP doesn't participate in the Common Language Runtime (CLR), and, in the view of many in the VFP community, shouldn't do so. Making VFP use the CLR would require removing significant capabilities from the language, especially those that distinguish it from the other Visual Studio languages (like the native data engine).

But without CLR participation, VFP sticks out like a sore thumb in VS.NET. VFP is different from the other VS.NET products in other ways as well: while you can use the common VS.NET IDE for some VFP tasks, it has its own distinctive development environment; unlike the other products in VS.NET, VFP 7 is (almost) fully backward compatible. Mike Feltman of F1 Technologies expressed it this way, "Given that VS.NET is now a product in itself, versus a product bundle, does it make sense for Visual FoxPro to continue to be bundled with Visual Studio?"

The next argument for making VFP a stand-alone product is that VFP 7 is much closer to being done than the rest of Visual Studio.NET. The announcement indicates that VFP 7 will ship in late spring. Visual Studio.NET is expected sometime in the second half of the year, but no one really knows when it'll be ready. Another bonus in the separation is that service packs and future versions of VFP won't be tied to service packs for all of Visual Studio; instead, the VFP team can ship when they're ready.

The key arguments have to do with marketing issues. As a member of Visual Studio, VFP has to be marketed based on the way it can be used

in Studio's worldview, that of building enterprise, Web applications. If VFP is a stand-alone product, it can be marketed for its strengths – building desktop and client-server applications. The announcement makes it clear that this is the strategy the FoxPro team plans to pursue. This paragraph really lays it out:

"Visual FoxPro is an extremely powerful application development tool. Its data-centric, object-oriented language offers developers a robust tool set for building database applications deployed on the desktop, as client-server solutions, or on the Web via components and Web Services. Visual FoxPro is optimized for building fast database applications and components."

There's one final benefit from the change. Visual Studio.NET (and the whole .NET initiative) is a version 1 product. (If you don't think that's true, visit any online discussion group for Visual Basic. Those developers are not happy about the prospect of throwing away their existing applications and starting over.) It's not at all clear yet whether .NET will be successful, either technologically or in the market. VFP, on the other hand, is a stable, mature product.

With all these benefits to excising VFP from the Visual Studio package, what are the losses? Most of these seem to be political, and all need to be resolved through appropriate marketing. The biggest concern is the perception of others outside the FoxPro community. Will people see this move as clear evidence that FoxPro is dead?

Looking from the other side, over the last few years, there have been a number of developers who've started using VFP because it was in Visual Studio. That won't happen when it doesn't land on their desktops. This is somewhat mitigated by the fact that, of course, VFP will continue to be part of MSDN.

Finally, there are some VFP developers who've been able to fight corporate or client pressure to give up VFP because of the Visual Studio name.

How do we answer these concerns? With steady and powerful marketing that clearly communicates Visual FoxPro's role in application development. If VFP is properly marketed, none of these issues should matter. If it continues to get the weak marketing it's had over the last few years, they could all be serious problems, but it's hard to imagine that happening. As a member of Visual Studio.NET, VFP would certainly be at the bottom of the marketing totem pole. Now the VFP team is free to market the product for what it does well.

I asked Robert Green, Visual Studio Lead Product Manager, about the team's marketing plans. He responded, "We want to market VFP to the space where it shines, that is building high performing database applications, whether they work with Fox or SQL Server data. We want to boast about the product's highly productive set of tools, its data handling and OOP capabilities, and its ability to build and use components and Web Services. I don't have lots of specifics yet on how we will do this, but I can say we are listening to the community and looking at all of our options. We hope to have more details at the spring shows."

I'll offer Robert one suggestion. One of the first steps they should take is to educate Microsoft's field representatives about the product, letting them know where it's appropriate and what it can do.

Now that every other product has either been moved into VS.NET (VB, VC++ and C#) or has been dropped (VJ++), Visual FoxPro is Microsoft's only pure desktop development system. As such, it might get a lot of attention if the .NET strategy is unsuccessful. On the other hand, a successful .NET platform could move Visual FoxPro even further into isolation. On the third hand, not every application needs what the .NET platform brings to the table, so there's a tremendous opportunity for a strong product to meet those needs.

So what's the bottom line? I'm cautiously optimistic. In Visual Studio.NET, VFP was the odd man out. On its own, it's the same powerful development environment it's always been and now, maybe, Microsoft will tell other people.

More VFP news

Microsoft has released Service Pack 5 for Visual Studio 6. It contains a few bug fixes for VFP 6. You can check it out at <http://msdn.microsoft.com/vstudio/sp/vs6sp5/default.asp>.

Another must-read book

After I'd finished my March column, in which I recommended several books, I read one that I think every software developer who has anything to do with user interfaces must read. The book is *GUI Bloopers: Don'ts and Do's for Software Developers and Web Designers* by Jeff Johnson (Morgan Kaufmann). It's a practical guide to good interface design. As the title suggests, the approach of the book is to show interface problems and it does, but it backs each one up with the

reasons why it's a blooper and ways to do it right. Best of all, this book is well-written with enough humor to make it a real pleasure to read.