

Is Chicken Little Right This Time?

Whil Hentzen

Earlier this year, a number of articles in the trade press described, in various fashions, "The Death of FoxPro." The first article claimed that Microsoft had announced that FoxPro would be discontinued after version 4.0; followup articles proclaimed "Xbase is dead" and other tripe. The immediate result was a firestorm of messaging on CompuServe's three FoxPro forums (I think I read more than 2,000 messages on the subject during the three-week period). The aftermath is still being played out in offices across the world as anti-Fox developers makes copies of the articles and shove them into the faces of their FoxPro associates; the more politically astute slide these articles by upper management in an attempt to torpedo the use of FoxPro in the organization.

To compound matters, the public responses by Microsoft haven't been encouraging. "Lukewarm" is the term I've heard most often (well, the one I can repeat in public) used to describe public postings and quotes by Tod Nielson and Jon Sigler. I'm reminded of the admonitions that the upperclassmen used to give newly accepted pledges at the fraternity: "Oh, no, don't throw the Pledge Educator in the lake! No, that would be bad! No, please, stop, please . . . And whatever you do, don't throw him in with his clothes on. They'd get all wet!" If I was responsible for marketing a product that was slighted the way FoxPro has been, I'd be screaming at the top of my lungs 24 hours a day. Tod and Jon's attitudes have been more along the lines of "Oh, I guess I gotta post something on one of the Fox forums. Like I don't have enough to do today . . ." <<Whil, I have nothing against taking MS to task for their lukewarm support for FP, but I also assume that we rely on Tod and Jon for a lot of help. Need we make this critique so personal, especially at a time when we're discussing PR opportunities at the Fox DevCon?--rw I've gone back and forth on this myself, and my personal (admittedly still emotional) opinion is that they're getting away with murder. Everyone is so scared of offending them that they can do anything they want and everyone just treats them with kid gloves. Furthermore, if this is the kind of help we get from them, then what do we care? They've done so little and are so ineffective... My company has quit the Solution Provider Program after going around with Sigler and getting absolutely nowhere. I think it's time to talk about the Emperor having no clothes...OTOH, if you think this reflects badly on Pinnacle, then I can be reasoned with <g>>>

Due to publication lead times, this happened just after the May issue of *FoxTalk* was put to bed, so I've had nearly a month to reflect on the situation. I've also attended a number of user group meetings, talked to countless people on the phone and answered well over a hundred e-mails about this topic. And, yes, I've had to discuss things with some of our customers as well.

As a result, I think I have a pretty good grasp on what you're going through. In short, three questions are asked over and over again. I'll paraphrase them as they've been posed to me:

1. *"I just had my boss (if you want, substitute another developer, a customer, or a good friend who just starting learning Access), toss the article (or, even worse, a company memo that picks selected quotes from one of the articles) in front of me. They claim that I'm an idiot for using FoxPro, that I should have known that Microsoft was going to kill the product, and that I shouldn't go out of doors anymore because I obviously don't know enough to even get in out of the rain. How do I respond?"*

2. *"We've just spent <fill in the blank> months rewriting our vertical market/mission critical application in Visual FoxPro (or) we're considering a total rewrite of our <name of product or application> in Visual FoxPro. But with the bad press surrounding it, we're wondering if we should consider another tool that has a better chance of being around in five years. We're concerned that Microsoft isn't going to support VFP in the future."*

3. *"I'm starting to learn VFP but now I'm wondering if I should be learning Delphi or PowerBuilder. Their companies are passionate and evangelical about their products; Microsoft's FoxPro marketing team is invisible. I don't want to waste 200 hours of my own time in a dead end."*

"Whil, you run a FoxPro development shop—what are you going to do?"

Let's put this ruckus in perspective. We've been hearing this stuff for five years. Microsoft was rumored to be "killing FoxPro" since its purchase many Junes ago. Second, the same rumors float around the other forums—you should have seen the threads on MSACCESS when the "12M to load Access95" requirement came out, or the howl that arose when Visual Basic 4.0 (released months after VFP3) contained nary a hint of object orientation. Third, while it's not a panacea (Go to the head of the class, Whil <g>), Visual FoxPro is still the best all-around desktop database development tool available. And remember, the industry rags want to sell newspapers and magazines <<Not true! Certainly not here at Pinnacle <bg>!>>. Evidently there isn't always time to check all the facts and make sure they're presented objectively.

Finally, the arguments that follow don't make an assumption on whether or not Visual FoxPro will be around past version 4.0. My personal bet is that we'll see VFP merged into the Developer's Studio just as Microsoft announced at DevCon in 1992, and that we'll be extremely pleased at the additional tools and capabilities we'll be given over the next few

years. But the following arguments are equally valid whether VFP stands on its own through 2015 or is merged into Visual Basic or Access in 1998.

“How do I answer the naysayers?”

Typically, the folks who dump a copy of the article or sneeringly hint at the lack of wisdom of your choice of development tool are a few bricks short of a load when it comes to having all the facts. Nothing dissuades someone like facing someone who has the facts at their disposal. Elsewhere in this issue is a compilation of facts that should help you defend FoxPro. Written by Editorial Board member Walter Loughney, you'll find this piece useful to counteract half-truths and innuendos.

Obviously, you won't be able to change the mind of someone who has already made their decision, but I've found that many people who are asking for information are interested in sticking with FoxPro, but just need some reassurance. The objective information Walter has compiled will help you set their minds at ease.

“What should we write our next version in?”

The primary fear here is that an investment in Visual FoxPro will be wasted. “If they discontinue it, how will we upgrade our application?” When you come down to it, this fear is moot. How many of you “ported” an application from FoxBASE to FoxPro 1.0? Or FoxPro 2.6 to Visual FoxPro? Sure, you could RUN the application in the newest version (and most of the time it would work). But why bother? The tools provided for any conversion have proved unusable for all but the smallest of applications. You couldn't take advantage of the 2.x Screen Builder capabilities if you moved an application from 1.0 or FoxBASE, nor could you do anything with the object oriented or local/remote view capabilities of Visual FoxPro with a conversion from 2.x.

The expectation of converting an application written this year into a new version released in three or four years comes from the use of mainframe and mini-computer development environments that changed very little from release to release. So choosing a language on the premise that you'll be able to convert it to a new version in several years is a bad idea. Visual Basic developers aren't ‘converting’ their VB 2.0 apps to VB 4.0, and Access programmers who built apps in version 1.0 (yes, there are a few) are rewriting them from the ground up to take advantage of the new features in the latest release.

The factors in determining whether or not to use Visual FoxPro to develop applications should include the stability of the vendor, the technical capability of the product, and knowledge of the product by the development team. *Every development tool in use today is going to change and systems written today aren't going to convert well, no matter which tool is used.*

“What should I spend my time learning?”

A while back, one of those office documents that floats around the world described how adherents of various development languages would hunt elephants, the point being to make fun of each tool in the appropriate fashion. The joke for FoxPro developers was that they'd spend years becoming expert in the use of the rifle used to shoot elephants, and just as they had mastered the tool a new model of the rifle would be released, and the developer would put everything aside to learn how to use that new model.

FoxPro *does* have a significant learning curve in order to use it productively, and we've been spoiled over the past half-decade as many of the fundamental concepts (native SQL, Screen and Menu Builders, and the Foundation Read) haven't changed significantly since the release of FoxPro 2.0. We've had a lot of time to become extremely proficient with this tool, and now we're back near the starting line with Visual FoxPro 3.0. Compounding matters, our expectations are much higher than they were five years ago when it comes to the capabilities of our applications and the speed at which we want to be able to write them. Investing several hundreds of hours in a new version isn't an encouraging thought—we want to deliver a new VFP app NOW!

So, faced with this background, does it make sense to climb the learning curve with Visual FoxPro despite the uncertainty? In a word, yes. First, VFP has the best object model available in a desktop database development environment. If you want to start fights with Visual Basic developers, ask them the definition of “subclassing” in their tool. When they can't answer, tell them “Ctrl-C.” <G> The concepts you learn in VFP will be applicable no matter what development tool you end up using in five years. And the best part is that you don't have to learn a whole new language—you already know how to use FoxPro and the Xbase syntax, so learning OO is less arduous than if you tried to do so with a completely new language like Smalltalk. The various development tools are quite complementary—many Delphi developers have indicated that their development tools are easy to pick up once you've used VFP.

Second, Visual FoxPro is still the best desktop database development tool, and will be for a number of years. Step back for a second and compare FoxPro 2.x and Visual FoxPro. The development team at Microsoft has done an incredible, absolutely mind-blowing job. VFP 4.0 is under development and popular opinion has it due for release yet in 1996. You won't have to do a paradigm shift when you move from version 3.0 to version 4.0, so the investment you make in VFP 3.0 will translate nicely to version 4.0. And as those of you who are still writing new apps in FoxPro 2.6/DOS know, you can

expect at least a minimum of three or four years of productive use of a tool after release. So that means that we can make money with VFP for at least five years. That sounds like a good investment to me.

Finally, in terms of the bigger picture of “What do I do?”, I suggest that you start to focus on the process of software development. While cutting code isn’t a trivial part of software development, it isn’t the only part of writing high quality software. People want solutions, not just clever code. Learn how to spec out your projects, design properly, test your apps, make them maintainable, and so on. Language syntax is just one part of the deal. We’ll talk about this more in future months.