

Figures:

File for Subscriber Downloads:

Whither... Reprise

Whil Hentzen

My March editorial, "Whither Our Fox—Part 2," generated lots of e-mail—even more than my April explanation of Quantum Buffering. Most of the feedback has been complimentary—the objective look at where we stand now, and the proposed game plan for Microsoft generated confidence and settled a lot of people's uncertain nerves.

But a few whiners have taken the time to write in. To a person, these are folks who couldn't hack it as Fox developers, and have opted for a strategy of playing with the technology *du jour* instead of developing actual competence with a specific tool. Yesterday they played around with Java. Today they're enthralled with .NET. Tomorrow it'll be something different.

Typical is the following introduction to an e-mail exchange I've had over the past month. I thought you'd find it both interesting in terms of the muddled thinking that passes for analytical thought these days, and well worth examining because it's likely that you will run into a similar situation yourself. (By the way, the entire exchange is included as part of this month's Subscriber Downloads.)

I read your editorial "Whither Our Fox", and as I suspected it would be, it basically sounded like an desperate cheer to aid the defense of a fallen hero that everyone knows is in his death-throws (sic) but is afraid to say as such.

It's true that the nay-sayers have been "saying" for years. I feel that VFP could be around for years, but the reality is that this lone tool will not support a "growing" need no matter how badly you think MS has mucked up VFPs marketing. Falling VFP 3rd-party tool sales, magazine subs, books, etc., etc. paint a telling story. MS pulling VFP from .NET is MAJOR whether you play it down or not. VFP getting it's own identity is a rather silly "cheer" isn't it? Have you worked with .NET? It is MAJOR and it will define Microsoft's future in terms of development environments. Debugging SQL Server stored procedures from the development environment and deriving Visual Basic classes from C# classes are things that make the world go 'round. There's also the little software tool called SQL Server 2000 and MSDE for that matter. SQL Server 2000 is a highly scalable yet affordable (sic) database solution that is even beating Oracle in taste tests of late. It's easy to use, robust and inexpensive. (sic) It's too good and too easy to get not to use. It's everything that FoxPro will never be.

If anything VFP will probably be sold off to a Company like Computer Associates who would carry it for a few years at best ala Clipper. VFPs ultra kludge-ridden interface building tools and archaic code conventions will not build a market. MS could not sell VFP as a middle-tier tool because it's a ridiculous (sic) proposition. How can it possibly survive against C#, VB, C++, ADO+, etc. Answer? It can't.

Today's developer must be a master of many tools and languages. SQL, VB.NET, C#, ADO+, ASP+, Javascript, XML, etc. are but a handful of technologies that are routinely mastered by a senior enterprise developer today. Gone are the days of the old timer developer who has been clutching onto the VFP manuals for 20 years and is afraid to master new technologies and migrate to superior toolsets. If they weren't afraid, they'd drop VFP like a rock, because it just doesn't stack up to what is available today. Not by a long shot.

I found it interesting that the writer was willing to denigrate "a desperate cheer" for Visual FoxPro while in the same breath, sticking his neck out for a brand new application development environment that is still in it's first beta.

I found it surprising that the writer could flatly claim that SQL Server 2000 was "inexpensive." It all depends on the installation. Against Oracle, who made \$6 billion in profits last year – on \$9 billion in sales? Sure, I'll agree - Oracle might, ahem, be a bit overpriced. But I found it difficult to believe that the writer

took an apples and apples comparison of Fox and SQL Server to a potential customer, and found the licenses for SQL Server to be inexpensive, compared to the free license for a similar Fox back end. It all depends on the specific solution. One broad, overreaching statement can't be accurate. As the saying goes, "Once is not a trend."

And I found it fascinating that a corporate developer today must be a master in at least a half-dozen tools and languages. Did you know that? I sure didn't. Acquainted with, yeah, I'll buy that. But an expert? Wow, people sure must be smart these days to be able to master a half-dozen or dozen complex, ever-changing tools. I'm not sure where they find the time. Maybe they don't have to actually produce anything with that mastery.

Then, in a follow up e-mail, the writer continued:

What's corporate America's IT industry demanding today?

Single-user and very limited multi-user desktop xBase apps? Not.

Scalability. Availability. Fail-over support. Clustering. Load balancing. Component/services driven design. Multi-tier partitioned apps. Browser-based user services. Heterogenous (sic) disparate architectures. B2B. Etc.

Go on a sales call with a noteworthy Fortune-500 prospect and mention "FoxPro" as a potential solution. You probably wouldn't get more laughs doing stand up comedy.

I was even more puzzled by this one. Evidently, this fellow wants to play in the enterprise computing arena. That's all well and good. I don't know that anyone has ever argued that Fox should be used for running airline reservations systems, global banking transaction networks, or automated factories. Remember, Fox's roots come from a fellow who just wanted to analyze sports statistics so he could participate in his office football pool in the late 70's.

That said, there's a time and place for lots of different tools. Fox has always been the best tool for building database applications on the desktop, as well as client-server front ends. It is also great for creating components for Web applications.

So it's not going to be used as the primary tool for tying together the internals for \$50 billion corporations or forecasting the weather. There aren't that many of those types of applications around. There are, however, 30,000,000 businesses that can use Fox as their primary application development tool. It's just that this type of system isn't hot, new and sexy.

Fox isn't always the best tool. When it isn't, you look for other tools, or you turn down the work. But using Fox over and over and over again doesn't mean that you're afraid to learn to use new technology. Maybe the job just doesn't require it.

At the same time, Fox is the best tool for a very, very large variety of work. The native database engine, the object oriented tools (that's real object orientation, not just marketing lingo on the side of a box), the hooks with Windows, Automation server capabilities, internal string processing... the list just does not stop. Take the lowly macro expansion operator – something we've had in Fox since before I can remember. It gave us the ability to compile a line of code at runtime – it's a feature that distinguishes Fox from many other development environments. Indeed, many of the third party frameworks that exist for Fox got their start because of this capability.

But this attitude that Fox is on its way out because it's not getting the front cover ink isn't new. I remember engaging a well-known developer on CompuServe back in 1996 or so. He had just jumped to another language—I forget which one—and occasionally stopped by to tell us we were all idiots for sticking with FoxPro. Several years later, an order for a half-dozen books showed up from the developer, who had returned to Fox...

So what's the point? You'll run into folks who will downgrade your choice of development tools. It's important to be able to distinguish the valid arguments from the nonsense. It's true that Fox is a niche tool, and the apps it's best at don't make the front cover of *Time* and *InfoWorld*. At the same time, your use of Fox is a strategic decision. Do you want to be a little fish in a big pond, or a big fish in a smaller pond?

There are dozens of “niche” languages that all have very specific needs. We don’t all need to learn VB and C++ just because they’re popular.

And, very possibly, when you cut through the clutter, it may well just be sour grapes from a former Fox developer who couldn’t get going when the going got tough.