Whither Our Fox – Part 3

Whil Hentzen

In February of this year, VFP 7 was pulled out of the Visual Studio box. This had caused a lot of angst – wondering if it signaled the oft-awaited death knell for Fox. At the time I penned an editorial that explained the facts, and offered my opinion that this was indeed a good thing, and, finally, offered my idea on what I would do if I was made King of Fox software.

Since then, a couple of things have happened. First, as you know, Fox 7 has shipped. Second, Ken Levy has been named VFP Product Manager, with Robert Green being moved up into a higher-level position, overseeing both VFP and Visual Studio.NET.

As soon as Ken told me the news, I made arrangements to head out to Redmond to meet with him and the other members of the Fox team. Here's what we discussed, as well as a sneak preview of what we're going to be seeing…

What have we seen over the last ten years (has it been TEN years already?) The June 9th, 1992 version of Fox 2.0 with the brand-new MSFT copyright notice. FoxPro 2.x – DOS, Mac, Windows, Unix. 3.0 and the tumult that many developers faced. Disastrous marketing by MSFT, bad press and a lot of rumours. Incremental improvements in 5.0 and 6.0. Robert taking over marketing and fixing all the stuff that was broke. Becoming part of VS. And many developers wondering if that was the beginning of the end – losing our identity. And now, 7.0 released, no longer in the box, those same developers wondering if **this** is the beginning of the end… which way do you want it?

So where are we now? In a nutshell, Visual FoxPro 7.0 is the best tool available for building rich desktop apps, for building front ends to client server applications, and for consuming web services from VS.NET. It's a solid player on a diverse team of tools.

And where are we headed over the next five years? You'll see a lot more press about how VFP is a rich, client-side application. How it consumes web services. How it will continue to get stronger and stronger with the rest of the world – interoperability. Delivering better error messages, even when the other app is helpless. Connecting to more applications, working with more ActiveX controls for a richer user experience, tighter connectivity and omnipresence.

It's NOT a replacement for VS.NET, folks. The push is on to distributed applications, using the Web as a delivery mechanism. While you can build web applications solely with VFP, it's not designed to be a one-stop solution. If you're new to the Web, you will find that building your first apps with VFP and a third party tool will get you up and running quickly, and get you comfortable and well grounded in the concepts you need for these types of apps. If you have specific needs, you may also find that a VFP only solution will fit your needs.

But VFP wasn't designed for this purpose, and won't necessarily scale as well, or as easily, as tools that are built with the Web in mind. VS – VB.NET, C# and C++ - is a better choice. VFP will provide a critical part in those apps – as the client side application. Just like SourceSafe plays a critical part – providing source code control to the developers. And SQL Server – providing 'big-iron' capabilities for data handling.

At the same time, there are a lot of Visual Basic and Access developers who feel like they're being left behind. For a while, I was seeing a lot of interest in topics like "Visual Basic for Fox developers". Since the second beta of Visual Studio.NET, I'm seeing something different. Visual Basic developers, in particular, are being split into three camps, much like Fox developers were six years ago with the introduction of VFP 3.

The first group is the early adopters of .NET – they've jumped in with both feet, feel comfortable with the new paradigm, and in some cases, have started delivering applications. The second group is taking a slower, wait-and-see approach, much like most Fox 2.x developers. They didn't start developing 3.0 apps in June of 1995 – it took them a while to move – both themselves as well as their customers. But eventually they made the transition. And the third group is easily identified by the heel marks in the ground…. "I will give up VB 6 when you pry my cold dead hands off of the keyboard." These folks are actually candidates for converting to Fox – they're used to talking about objects, they want an easier way to connect to data,

and, they are interested in a richer, more robust command and language feature set. (Well, that's until they see the 7 lb Hacker's Guide… <s>).

I'm probably overstepping my bounds in even mentioning VFP 8, but I can assure you that the feature set is already being worked – I had a couple of hour long conversations about the next version of Fox while in Redmond. That's the sixth major release of VFP since Microsoft's acquisition. I think it's pretty obvious that Fox is going to be around for a while.

As I was writing this editorial, I received a phone call from a fellow who, while not new to Fox, is new to the community – he'd just discovered the Wiki and the Universal Thread earlier in the week, and was interested in learning how to contribute. And that's a hallmark of our Fox – it's not just the language and the tool, but it's the community that has congregated around it as well. The passion (some would call it zealotry) we have for Fox has certainly played a part in keep Fox visible in the eyes of upper management at Microsoft. Indeed, I'd argue that, per dollar of sales revenue, Fox has probably garnered more attention at Microsoft than any other product they've produced. (Well, perhaps except for Microsoft Bob, which garnered negative sales revenue… hey, now everyone has something else to pick on!)

And in the coming years, one of Ken's primary goals will be to marshal the community – enhancing the knowledge base for Fox – certifying ActiveX controls, writing white papers, articles, and books, speaking at user groups, providing tool samples, creating third party products, and dozens of other activities. He's not going to be able to do it himself – he'll need our help. I've regularly received offers from fellow developers, asking what they can do to help. When I was in Redmond, we talked a lot about leveraging the talent and energy of you – so now's the time to write in – to me, at whil@hentzenwerke.com, or to Ken, at klevy@microsoft.com, to offer your help.

Finally, one last thing you can do right now for the Fox. If you've been sitting on the fence up to this point, now's the time - upgrade to VFP 7 now!!!!! Even if you're not actively developing VFP apps (yes, there are a few folks still using Fox 2.x for Windows, or FoxPro 2.0 DOS, or even FoxBase…), pick up a copy and start working with it as part of your R&D effort. And if you're a current 5.0 or 6.0 developer, grab the 7.0 release and start getting used to the features.

You're going to see a LOT of new things come out of Redmond, and specifically, the Fox team. There's never been a brighter time to be a Visual FoxPro developer.