

.NET does Data? Not like the Fox!

Anders Hejlsberg (The Father of C#) was recently interviewed about the future direction of C# programming. In the excerpt from that interview that you can read at [this link](#), Anders discusses a number of data-related issues. He even uses the dreaded 'F' word <g>. No, not **F#**, although he does mention that, too, but **FoxPro**. Here's one quote (italics mine):

"No. I wouldn't say that it [C#] specifically borrows from FoxPro. As I have said before, the area where we're already laying some groundwork for in Whidbey (Visual Studio 2005) is this big and largely unexplored area of deeper language and data integration. And, of course, FoxPro is a language that has been there ... *I think they show how closeness to data is tremendously useful* for a certain class of applications." No disrespect intended, but, Duh!

If you read this interview, you'll note that Anders is mostly talking about features for "Orcas", the code name for the *next next* version of Visual Studio, the one after Whidbey, which is slated for release in November. So I'd guess a release of no sooner than 2007 for Orcas. If I read Anders right, he's saying that by the time .NET is about 7 years old, C# might possibly include features that make it 'close to data', which would be 'tremendously useful' for some applications. Such as any application that manages data. In the meantime, VFP has been doing that extremely well for longer than Microsoft has owned it.

A little further in the interview, referring to FoxPro and other dBase-d languages, Anders says "They lacked some capabilities that programmers,

generally speaking, want." Like maybe... marketing? Ken Levy has done the best job by far of anyone I can remember in his 'VFP evangelism' position, but his hands are clearly tied. Sure, VFP developers have always had a number of features on their wish lists (just check out the listings on the Fox wiki and the UT). But the number one 'capability' VFP developers have historically requested was a respectable marketing effort even in the same ballpark as VFP's technical capability.

As for the last FoxPro mention in the excerpt, check this out:

"Let's just say for the sake of argument that I want to make it as easy to program data in C# as it is in FoxPro. Or pick whatever goal you'd like.

I'm not saying that is a particular goal."

On the one hand, he accurately points out that it is easier to program data in VFP than in C#. No big revelation there. But then he is careful to deny any specific attempt to achieve the same ease of data management in C# as he knows is in VFP. But Microsoft wants all us VFP developers to move to .NET as soon as possible, even though everything we do is directly related to managing data. Again, keep in mind that we're not even talking VS 2005 here, but no sooner than the next version after that.

I did like everything Anders said regarding the possibility of an Integrated Query Framework that would make it possible to query data over multiple different types of data, including XML and objects, not just SQL/relational DBMS. It will certainly be interesting to see how that works out in Orcas. Although I find this Anders Hejlsberg quote at the end of the interview excerpt a little foreboding:

"You sort of declare what you want to have done, but not exactly how you want to have it done. And in many ways, we've educated generations of programmers to think not just about what, but about how, and to explicitly state how in their programs.

"In many ways, programmers have to gradually unlearn that and learn to trust that when they're just stating the 'what,' the machine is smart enough to do the 'how' the way they want it done, or the most efficient way."

Hmm... The goal is to improve the .NET languages so that the developer just has to tell the computer what they want done, and it will 'just happen' in an efficient, consistent (and correct) manner. Sounds quite a bit different from what I've come to expect in VFP, where I always have a number of ways to complete any given task, including querying data. Frequently, that ability to state the 'how' makes it possible for me to produce code that works – the first technique I use may not be fast/efficient, may not work quite as expected, or work at all. Or as I recently heard a developer put it, "There's almost always about a million ways to arrive at the same location. That's what keeps things interesting." And please forgive me for the chuckle, but why do I have such a hard time swallowing this concept of turning my programming commands into requests, from a high-level manager at a company whose operating system has such a hard time running all the Windows-certified software on my computer?

Hey, I'm not bashing Microsoft/C#/Anders Hejlsberg here. Really. I think his goals for C#/.NET in the Orcas timeframe are lofty, worthwhile, and beneficial for developers. But VFP developers surely have to read this kind of stuff and just shake their heads in incredulity over the lack of data-centric features in the sister .NET languages. We have to wonder how a company that owns VFP and has spent hundreds of millions of dollars (maybe into the billions?) developing and marketing .NET still has to say that its .NET languages don't natively handle data all that well. And, near as I can tell, features like Integrated Query Framework address querying issues, not data-update issues.

Rumors about the incorporation of VFP-like data-centric features in .NET have been circulating lately. For example, here's a [link to a David Stevenson blog](#). That would certainly make .NET more interesting to VFP developers, but we'll have to wait and see what form such features might take.

Oh, and did I mention that this is all contemplated for the *next* version of .NET? No, not Visual Studio 2005, but the one after that.

Source: http://blog.visionpace.com/2005/07/net_does_data_n.html