

Anders Hjelsberg tries not to talk about FoxPro while talking about FoxPro

Malcolm Greene posted a link today on Profox to a [Mary Jo Foley interview](#) with Anders Hejlsberg about future data-handling features for C# 3.0, rumored to be based on FoxPro (you'll have to wade through a good bit of the interview before finding the references to FoxPro on the second and third pages). Here is Malcolm's comment:

After reading this article I remain convinced that Anders and his C# yes-team "just don't get it". They are sitting in their ivory tower, too far removed from real world business needs, and with (IMO) a not-invented-here type of attitude towards FoxPro.

Here is my response to Malcolm, which bears repeating here:

I agree that probably Anders and the C# team don't "get it" as much as the VB.NET guys, who are probably much more influenced by Alan Griver and his VS Data team.

I think these comments from Anders in the interview are pretty funny, actually:

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.

It's almost like he's holding his nose while admitting that the FoxPro model of working with data is superior to C#, while not really admitting it. :-)

I'm expecting that the PDC sessions in September that are dedicated to new models for working with data in future versions of Visual Studio will be quite interesting. The widely-circulated quote from VS-Live recently indicated a rumor that VB.NET would be getting Fox-like features, and that Anders was also working on some new approaches to data-handling for C# 3.0. That made it sound like VB.NET might end up being more Fox-like than C# -- but it's too early to tell yet.

If they do it right and put cool new data stuff into the .NET framework, maybe all .NET languages will benefit. I'm seriously considering [attending PDC](#) this year (September 13-16 in Los Angeles) just to see the sessions likely to focus on future data innovations and check out the reaction to them:

Visual Basic: Future Directions in Language Innovation from Paul Vick
Visual Basic is designed to be the most productive language for writing data-centric, solution-focused applications. Meet with the designers of VB and learn about

upcoming language features that will improve developer productivity, including new features that enable optimized queries over objects, XML, and databases in a consistent way.

Session Level(s): 300

Track(s): Tools & Languages

C#: Future Directions in Language Innovation from Anders Hejlsberg

Join Anders Hejlsberg, Distinguished Engineer and chief architect of the C# language, for an in-depth walkthrough of the new language features in C# 3.0. Understand how features like extension methods, lambda expressions, type inference, and anonymous types make it possible to create powerful APIs for expressing queries and interacting with objects, XML, and databases in a strongly typed, natural way.

Session Level(s): 300

Track(s): Tools & Languages

The following sessions also sound intriguing, but may or may not be directly related to the rumored Fox-like features:

Future Directions: Data Access and Storage

Abstract pending

Session Level(s): 200

Track(s): Data & Systems

The .NET Language Integrated Query Framework: An Overview

Modern applications operate on data in several different forms: Relational tables, XML documents, and in-memory objects. Each of these domains have profound differences in semantics, data types, and capabilities, and much of the complexity in today's applications is the result of these mismatches. The "Orcas" release of Visual Studio aims to unify the programming models through integrated query capabilities in C# and Visual Basic, a strongly typed data access framework, and an innovative API for manipulating and querying XML. This session introduces each of these areas and walks through how they are related.

Session Level(s): 200

Track(s): Tools & Languages

Using the .NET Language Integrated Query Framework with Relational Data

Database-centric applications have traditionally had to rely on two distinct programming languages: one for the database and one for the application. This session introduces advances Microsoft is making for the "Orcas" release of Visual Studio in programming languages and frameworks to help integrate relational data

and queries with C# and Visual Basic. These advances enable developers to express queries and updates in terms of their local programming language without sacrificing the server-side execution model of today's high-performance SQL-based approaches. Using these advances, database queries that previously were stored as opaque strings now benefit from static type checking, CLR metadata, design-time type inference, and of course IntelliSense.

Session Level(s): 300

Track(s): Data & Systems

Using the .NET Language Integrated Query Framework with XML Data

One of the key challenges to working with XML data has been the impedance mismatch between XML and programming languages. This session introduces advances Microsoft is making for the "Orcas" release of Visual Studio in programming languages and frameworks to help integrate XML and queries with C# and Visual Basic. The advances include a framework for navigating, querying, and transforming XML that is both easier to use and more efficient than current XML programming techniques. This framework marries the capabilities of XPath, XQuery, and the DOM with the language integrated query framework planned for C# and Visual Basic.

Session Level(s): 300

Track(s): Data & Systems

You can find all of the PDC session descriptions [here](#)

Source: <http://talkingfox.blogspot.com/2005/07/anders-hjelsberg-tries-not-to-talk.html>

PDC: <http://commnet.microsoftpdc.com/content/sessions.aspx>