

FOXROCKX

Visual FoxPro Strategy at Microsoft

*Historical overview of the 15 year evolution
and strategy of Visual FoxPro at Microsoft*

Ken Levy

Microsoft will never release VFP source code into open source because there is no business reason to do so and a list of reasons not to. The evolution of VFP goes on into the next decade with new and enhanced products that work with VFP, community projects such as VFPX, and of course the VFP community itself.

Microsoft's Early FoxPro Strategy

The purchase of Fox Software for \$173 million in 1992 was very strategic for Microsoft, and was the biggest corporate purchase Microsoft had ever made up until that time. Aston-Tate's dBase was still popular, Borland had Paradox, and there was PowerBuilder as the king of client/server tools. Microsoft needed three things from the Fox Software deal - the Fox developer team, the Fox technology, and the customer market share of FoxPro/FoxBase. Microsoft was just starting work on Access and it was more targeting power users, but there was still some overlap. Visual Basic was still in its early days.

There was an estimated 500,000 FoxPro developers at the peak around 1995, and millions of computers with FoxPro apps running (either DOS or Windows based). It took almost 4 years, not until after VFP

November 2009

Number 11

- 1 Editorial**
Visual FoxPro Strategy at Microsoft
Ken Levy
- 2 VFPX**
VFPX: Control Renamer
Rick Schummer
- 7 Deep Dive**
Introduction to GDIPlusX, Part I
Doug Hennig
- 13 New Ways...**
Collections instead of Arrays
Tamar E. Granor, PhD
- 17 Best Practices**
Best Practices Part VI
Jim Booth
- 19 VUProjectTools**
Beauty Studio
Venelina Jordanova
Uwe Habermann

5.0, for Microsoft to focus more strategy around VB and less for VFP. Basically, the VFP customer base and sales went from increasing to decreasing, as the saying goes in business: if you aren't growing, you're dying.

In the initial years after the Fox Software merger, Microsoft put a huge effort and lots of resources into creating VFP 3.0. There were about 50 people on the Fox team with a big marketing budget. In the following years, both Access and VB grew in market share and also competed in ways with the VFP market (and messaging), and by the time VFP 5.0 was released, many upper managers wanted Microsoft to just end VFP there. In fact, they did for a short time. I was there, in a meeting with 40 people, and the formal announcement was made to the Fox team that VFP was dead. It was very early 1996, and that meeting led to the Gartner Group releasing their report that VFP was dead, which had a major impact on future VFP sales. But the Fox team members along with the community made so much noise, combined with people like Eric Rudder (architect of VFP 3.0, who became Bill Gates' technical assistant), who convinced developer tools management to keep VFP evolving while decreasing the resources. In fact, the primary reason VFP lasted another decade with 4 more versions released was more about Windows sales than VFP sales. There are many Windows machines running VFP apps. When Steve Ballmer jumps around like monkey boy and yells "developers, developers, developers", he's thinking about selling Windows more than sales of developer tools.

History of Visual FoxPro at Microsoft

For each new version of VFP like 6.0 and 7.0, there were less resources, team members, and marketing budget. When VFP was included the Visual Studio box, it was just a bundle, no integration. When the VFP community saw VFP running inside Visual Studio and the possibility of running on the .NET framework, they also learned that if VFP went that path, it would seriously break FoxPro code backward compatibility and the VFP IDE would be gone eventually. Microsoft never had a goal to work on both VFP stand-alone and VFP for .NET. Having a new VFP for .NET would just take away resources from evolving VB.NET and C#, make it harder to sell Visual Studio and the .NET framework, and not really result in anything useful since it would not run old VFP code as-is.

Then after VFP 7.0, it was decided to keep VFP as a stand-alone product outside of the Visual Studio bundle, since it would not be part of the .NET platform. This made sense, and allowed VFP to ship on its own new version timeline. Each new version released, it was expected that was the last version.

Nobody on the Fox team, not a single person, expected a version after VFP 8.0 released. Sales continued to decline annually, and so did the marketing budget. The only way to grow/increase sales of VFP would have been to compete with Visual Studio and take away budget and resources from Visual Studio. In reality, the biggest competitor to VFP was Visual Studio (not Delphi or any non-Microsoft product). Most marketing and management folks at Microsoft would have preferred Fox developers use Borland's Delphi.NET rather than Microsoft VFP, since they would be building on the .NET platform rather than the old legacy COM. COM became the enemy, to move people away from it, just like HTML/JavaScript is a current enemy to the Microsoft platform today as well.

For VFP 9.0, there were only about 8 people on the Fox team, and even so, VFP 9.0 was a better release than VFP 7.0 and VFP 8.0 according to the community. Soon after VFP 9.0 released, Microsoft decided to create an Xbase add-on to give away in order to maintain initial upgrade sales and an up-beat perception of VFP. The result was a plan for Sedna, a download of useful sample apps and utilities that focused on VFP interop with other Microsoft products (.NET, SQL Server, Office, Windows, etc.). The other reason for Sedna was to delay the announcement of the end of VFP in order to save sales of VFP 9.0 as well as to protect the Fox community and the job market for VFP developers.

No Open Source of Visual FoxPro

The request for Microsoft to make the Visual FoxPro code base open source is a common (and logical) one. Here is some insight to why Microsoft will never release Visual FoxPro source code into open source. There is technology in VFP, like Rushmore optimized indexing, that is used in other Microsoft products SQL Server and Access. It's not the same C/C++ codebase, but many techniques and algorithms originated from VFP. Microsoft considers this intellectual property, an asset Microsoft does not want to be released.

But the two more significant reasons have nothing to do with the reason above. They have to do with business. While Microsoft is not focused on sales of VFP, it is focused on sales of Visual Studio and adoption of the overall Microsoft platform (stack of products and services). Releasing VFP into open source would result in less VFP developers using the current (modern) Microsoft platform of products, but may also result in someone or some company creating a competitive product against Microsoft. Microsoft would not want to see the code used to enhance a competitive product nor would they want to see a new product created that interferes with Visual Studio sales or .NET platform adoption.

The Final Strategy of Visual FoxPro at Microsoft

In the final 5 years of Visual FoxPro at Microsoft, while I was the last Product Manager (marketing/community) for VFP, the strategy was to market VFP to the existing community (mainly via upgrades), to do what was possible to keep the Fox community as strong, and to get VFP developers to adopt additional Microsoft products (.NET and SQL Server). In my role, I always viewed myself in 2 positions - one representing Microsoft as an employee, and the other as a FoxPro community member doing everything I could for Fox within the walls of Microsoft to evolve, save, promote, and help VFP and the community as much as possible. I spent nearly half of my time marketing VFP within Microsoft, at the Redmond headquarters and to the field offices, on messaging and keeping the VFP message positive/alive.

Management at Microsoft above the core Fox team were the decision makers for things related to VFP strategy. Nobody on the core Fox team had decision making ability around marketing budget or resources for the team, or what was done after each version shipped. There were a few key people on the team who, working together, probably extended the life of VFP an least one additional version and several years.

I think the customer base loyalty was yet another factor in how long VFP was extended beyond 6.0. But by the time VFP 9.0 was released, the amount of sales for all versions of VFP combined annually was less revenue than Microsoft sales of Visual Studio in only one day. The cost to evolve VFP relative to the amount of money it generated (ROI) was far less than putting more resources into Visual Studio and .NET languages. Plus, some Fox team members were ready to move on or leave Microsoft, and it was nearly impossible to find qualified people to replace them. It helps to put this all into perspective if you think of Visual Studio as a competitive product to VFP, even though it was owned by the same company. Remember when Apple worked on both the Mac and the Lisa computers at the same time, only one survived. In the case of VFP, it survived an entire decade after it was essentially killed (by it no longer being strategic).

Into the Next Decade

On January 15th, 2010, VFP 9.0 standard support ends. While paid extended support will exist for 5 more years, I don't expect any additional hotfixes or anything to be done for VFP, unless in the rare case it impacted VFP runtime on Windows 7 preventing customers with VFP based apps to upgrade to the latest version of Windows. Some suggest that

Microsoft killed VFP before it should have, and another way to look at the behind the scenes history is to see that VFP lived many years and versions beyond what it was planned. While Microsoft could have done more for VFP, it just really couldn't happen with Microsoft promoting and giving resources to Access, VB, and then Visual Studio at the same time. Only developers who have used FoxPro really appreciate it for what it was and what it still is.

Author Profile

Ken Levy is the president and founder of MashupX, LLC, specializing in consulting for community building around products and services, guerilla marketing techniques, multimedia creation, and software technology. Ken is the co-host of CodeCast, a podcast show associated with CODE Magazine. Prior to starting MashupX, Ken worked at Microsoft as the product manager for Visual FoxPro, a product planner on the Windows Live Platform team, and the community program manager for VSX (Visual Studio Extensibility). Ken is a long time recognized member of the FoxPro community, created GenScrnX for FoxPro 2.x, and many VFP components including the Class Browser. You can find Ken on Twitter @KenLevy, Ken's blog at <http://mashupx.com/blog>, or contact Ken at klevy@mashupx.com.