



Is Visual FoxPro obsolete?

Visual FoxPro is the database and programming language that was used to create Vision Point and Pro Series. In 2009 Microsoft “froze” development of Visual FoxPro and announced that it would be maintained but no longer upgraded. A few years later, they ended direct live support, but they made the documentation and other support materials available on their web site.

Users may understandably be concerned as to the continued viability of VFP as Windows is updated and more and more computing is done on the web. This concern is also often raised by vendors seeking to replace VFP-based software with another solution. This document is intended to address claims of VFP’s technological obsolescence, and to explain the business imperatives that offer assurance that it will continue to be a viable solution.

Is VFP technology obsolete?

Visual FoxPro really has two components – the **programming language**, and the **DBF file format** standard that it uses to store and preserve your data.

The **language** was designed not only to be powerful, but to be extensible - meaning that programmers can build on it to add capability that was not part of the original feature set. It’s an easy language to learn as well. VFP VARs have developed web access tools, e-commerce solutions, sophisticated report writers, links to Microsoft Excel, machine interfaces for point-of-sale or the shop floor, and mobile apps for smartphones, to name just a few.

The **DBF file format** is also an industry standard, similar to text or Excel formats. Data is easily exchanged with those and other standard formats so that solutions can be developed that bridge the gap between different products and enable a wider choice of options for the user. And for users with large scale data needs, VFP can use a SQL Server back end database along with its native format.

With all of these options, almost any solution can be deployed in a VFP-based environment.



Will Microsoft “break” VFP in future versions of Windows?

Well, they haven’t done that in the last four versions. But if they did, they would be shooting themselves in the foot. Believe it or not, VFP continues to be one of the most widely used programming languages on the planet. In addition to the commercial software that is based on VFP, it’s also embedded throughout corporate America, in departmental applications and single-purpose solutions by “power users” who don’t need top level programming skills to develop a useful solution to an ad hoc requirement. And interestingly, VFP is used as the basic teaching language in computer science programs throughout Europe. There are hundreds of thousands of people using VFP software every day. Microsoft can’t leave them hanging without an alternative, and every one of their alternatives is both more cumbersome and more expensive to use.

Microsoft’s former VFP Product Manager, Ken Levy, offered two reasons why Microsoft stopped further development of VFP. One is, after almost three decades of development, the product was pretty well optimized. There’s not a lot of value left to add, because anything a programmer needs that’s not included can be written. You can even integrate code written in other languages.

The second reason is that there was no licensing model built into the design, so they were not able to control distribution in the same way as their other products. VFP developers can distribute their products royalty-free to their end users, without paying any fees to Microsoft.

Yet they have to maintain the DBF standard file format, because so many products depend on it. They considered putting it into the public domain, but because many corporate IT policies forbid the use of public domain software, they opted to stop selling it but maintain passive support so that it didn’t disappear completely from their domain.

Will it last forever?

Well, maybe not “forever”. But given the size of its installed base, VFP will survive for decades to come, just as the mainframe-based COBOL language, originally developed in the 1950s, is still in use today despite not having been updated for many years.

The maturity and stability of these two languages is what gives them their staying power. Solutions can be developed at the user level, in response to business needs, without the ground shifting under their feet. So in this sense, the “frozen” code actually provides the **advantage of certainty** that is lacking in Microsoft’s other constantly evolving development tools. The benefit to the end user is that development is less expensive, support needs are reduced, and the longevity of the solution – and thus the long term value of the investment – is maximized.

As you can see, the death of Visual FoxPro has been greatly exaggerated!