OCX Fragmentation Hexes 32-Bit Migration

"A foolish consistency is the hobgoblin of little minds."
—Ralph Waldo Emerson

Jim Fawcette, Publisher and Editor

PUBLISHER'S NOTE

HOST INCOMPATIBILITY THREATENS PORTABILITY, THE ESSENCE OF THE COMMON OBJECT.

Any regular reader of this magazine knows that we are enthusiastic supporters of component-based software. We are also optimistic about the transition from 16-bit VBXs to 32-bit OLE Controls, also known as OCXs, although Microsoft frowns on that name. However, we aren't blindly optimistic. The numerous problems with OLE Controls not only threaten their adoption, but could slow the transition to Windows 95 and VB 4.0 alike. Because virtually all major VB applications require VBXs, the lack of 32-bit OCX counterparts will be a major impediment to corporations implementing Windows 95.

Commercial VBX/OCX developers are unanimous in the opinion that the OLE Controls architecture is superior to VBX's. But technical superiority is hardly sufficient for establishing OCXs as the standard object model for 32-bit software development. No one would argue that VBXs succeeded because of their superior implementation. In fact, VBXs are one of the most limited object models introduced. Academic purists blanch at even calling them objects. But they are the first and so far the only commercially successful component model, and more than one-half million developers use VBXs despite their limitations, or perhaps because of them.

VBXs succeeded because they were simple. They worked, and customers could figure out how to use them—plug them into VB's toolbar and drop them on a form. Also, I have to toot our own horn here: our magazine and VB catalogs provided very efficient media for VBX vendors to reach a focused audience of professional VB developers and create this market. Without that catalyst, VBX use wouldn't have exploded.

The launch of OCXs is much more confusing and uncertain. Many of the problems are understandable considering the fact that we're dealing with new technology and ambitious plans to implement OLE across so many development environments with overlapping delivery dates. My guess is that OCXs are so new that none of the hosts has even been thoroughly tested with a range of third-party controls. Those bugs will be worked out, and the spec is beginning to solidify.

But there are more fundamental problems. The promise of any common object model is that those components work across different environments. Yet, the implementation of OLE containers is so fragmented that this simply isn't true. Access, VB X.0, Visual FoxPro, and Excel all support OCXs in such varying fashions that true portability doesn't exist.

Access 2.0's support is so limited that there are virtually no commercial OCX products used in that market. In contrast, Visual FoxPro was introduced more recently, and support is far more complete in the beta copies we've seen to date. Yet Fox's OCX implementation is so different from VB's OCX implementation that most of the major UI-related tools won't work, despite Fox's excellent form engine.

For example, Visual FoxPro doesn't support all of the Control Development Kit's stock properties and events. Consider the impact of a few missing items: the stock Align or Tag properties, and the IContainer and ISimpleFrame interfaces. Their absence means that third-party frames, index tabs, panels, print previews, and picture boxes that work with VB can't work in Visual FoxPro without some workaround. I won't even touch the issue of data-bound controls that rely on the JET engine trying to use Rushmore or Delphi's Paradox engine. If OCX vendors are forced to create separate products for VB, Fox, Delphi, and Access, the benefits of the common object model will be decimated.

Another substantial problem that will be exacerbated is technical support. We've already written that the cost and difficulty of quality assurance and tech support is one of the biggest barriers facing small ISVs that hope to create a horizontal market for OCXs. This would be true even if OCXs worked transparently and smoothly across every host container. Add the incompatibilities that exist today and tech support will become impossibly expensive for these vendors. Look for the days of free tech support on $200 development tools to end soon.

Microsoft needs to establish a reference standard, a simple baseline, like the de facto VBX Level 1 support, that all containers must meet. This could be handled through an OLE Controls Host logo program, akin to the Windows 95 logo, with the participation of an ISV committee. Hosts should document the methods and events they do—and do not—support. Better yet would be a means to poll the container on supported methods. ■