Trying Times

Andrew J. Brust, president of Progressive Systems Consulting in New York City, specializes in developing (and training others in developing) client/ server and other custom database applications.

VB3, VB4, OLD BUT QUICK VS. NEW BUT...WELL, YOU KNOW. SO WHAT'S A PROGRAMMER TO DO?

A t the VBITS conference last February, I gave a talk on RDO, which requires the VB4 32-bit Enterprise Edition. I started off by asking how many of the 450-plus VB developers present still used VB3 as their primary production development tool—and how many thought they'd switch to VB4 by year-end. Most said yes to both questions.

I told them that this poll indicated how stressful times are right now for VB developers. A new version of VB, with a slew of new features and tools, has been available for a while, but for the most part *we can't use it yet*. Meanwhile it's expected that we'll be experts by the time we're allowed to use it, presumably by writing our own remote automation-enabled reservation system, or other enterprise behemoth, in our "spare time." Heck, that's okay, I didn't need to sleep anyway...and maybe my nephew can wait until he's 14 for the bar mitzvah, since I won't be able to make it this year.

We were so *patient* with VB3, getting better and better at it, leaving no undocumented stone unturned, so that when the next version came out we could *pounce* on it. We were hungry; we were ready; we were desperate. After the release of a seemingly mature VB4 beta, we heard (announcement? rumor?) that VB4 wouldn't ship until Windows 95 did... Hmmm. When VB4 finally did arrive, we learned that we could only use its sexiest features (like remote automation, remote data objects, Jet database replication, and Windows 95 GUI controls) in the 32-bit version, while our beloved VBXs, and thus our apps, would only work in the 16-bit version.

But VB4/16 seems a bit buggy, a bit slower than VB3, and its run times require more resources. So, in spite of our thirst, we stay with VB3 during business hours, then come home to those haunting VB4 manuals... and books... and articles... and conference session abstracts. This *is* the fun part, right?

I waffle about the time I've devoted to studying the VB4/32 environment. I don't want to fall behind, to stagnate. Yet until we can do most or all of our production work in VB4, until we're able to reinforce in our work what we've picked up in our self-education, we *can't* really become experts. Are we just spinning our wheels with all this off-hours study? I feel like I'm playing in the sandbox—that my customers just want apps they can use today.

It seems like Microsoft wants us to keep one leg planted on the 32-bit island and the other on the 16-bit one (for now at least). But the islands are moving farther apart every day. Our seams are about to split! Is Microsoft just leveraging us to help them win a battle? I mean, what they're doing makes sense. If we can't set the 32-bit example, who can? If we keep insisting that we only want to develop 32-bit apps, 32-bit will become mainstream that much faster. If we gripe that our PCs are too slow to run the OLE-centric VB4 and the apps we develop with it, then the new hardware needed will surface as the corporate standard. So what if we're not expert at VB4 when that first project in it rolls around? We'll pick it up as we go along, and the stress will make us feel young (and powerless) again. That which doesn't kill us makes us stronger... I think.

Of course Windows 95 and NT do represent a significant evolution in OS interface and, more importantly, infrastructure. We really do need to migrate to them. We really can't blame Microsoft for not developing every feature of every tool in parallel for both 16- and 32-bit platforms. We can't expect Microsoft to make business decisions based on sympathy for us and nostalgia for 16-bit technology. In a way, Microsoft's dilemma mirrors our own. They had to come out with new technology to stay competitive and appease users. Yet in the short term they have to support the old stuff—again to appease users.

Yes, these are stressful times for VB developers in particular and Windows developers in general. Yes, one more interim VB3 release would have helped a lot with our current learning curve problems. But no, we shouldn't just bellyache that things are so hard. And no, we shouldn't continue to use VB3 just because VB4's OLE core slows things down right now. We can all certainly agree that our development community is going through a lot of stressful change. But honestly—would we want it any other way?

Don't worry. Be happy. The hardware will catch up, the users will catch up, the ActiveXs (there's another!) will catch up and, notably, *we'll* catch up. If we could weather the move to Windows from DOS, we can certainly cross the 16/32-bit divide. And when VB5 comes out, we'll be all limbered up! ⊠

Guest Opinion