January, 1996

Editor's View

We're Not in Kansas Anymore

The new world of computing can be a big, bad place

My computer contracted its first virus recently. I follow the usual precautions—check every floppy I put in, download files only from sources like CompuServe that screen every file, and so forth. Nonetheless, one day recently, my system just wasn't right.

The symptoms were weird and seemed confined to Word for Windows. Nothing I could think of made sense, so I started asking around. What I learned was far more frightening that having to reinstall Windows or Word, two possibilities I'd considered. Some joker has found a way to attach a virus to Word templates. The virus is called "winword.concept."

As I talked to people about this virus, I learned that it first surfaced this summer and had spread through Microsoft's Redmond campus like wildfire. And with good reason.

Until this one, viruses could only be attached to programs. There was never a danger in looking at a document, reading a spreadsheet, opening a table. You didn't need to check those kinds of files.

But macro languages aren't just keystroke recorders any more either. WordBasic and Visual Basic for Applications are full-blown application languages with tremendous capabilities.

The winword.concept virus consists of a group of macros which attach themselves to Word's normal.dot template. I took a look at the code. I'm not a WordBasic programmer, but I could read it without any trouble. It's really basic stuff, nothing you'd need to be a skilled programmer to write.

Microsoft has provided a cure for this virus. It's a document that installs some macros of its own. The document both cures your system for now and inoculates it against future outbreaks of *this* virus.

Our little world has changed in a fundamental way. The cure for winword.concept doesn't protect you against other viruses using the same techniques. Many other applications are equally vulnerable to this approach.

What a shame. Just as it's gotten really easy to share information world-wide, we have to become wary again.

(To see if you're infected with winword.concept, open a Word document, choose Tools-Macro and make sure the "Macros Available in" dropdown lists "Normal.dot." If you're infected, you'll see a pair of macros right at the top named AAAZAO, AAAZFS, plus one called PayLoad. To clean up your system, get a file called WD1215.EXE available from Microsoft in a number of places including the MSWord forum on CompuServe.)

Rocky Marriage

I've been hearing a lot of talk about the interaction of FoxPro 2.x for Windows and Windows 95. Since I haven't installed Windows 95 (and don't have any immediate intention of doing so—I'm incredibly conservative about operating systems), I couldn't check these things out for myself. So I did the next best thing—spoke to some of my Microsoft contacts. Here's what I learned.

First, there definitely are some problems. FoxPro for Windows is not a true Windows product for several reasons. First, it's not a true Microsoft product. FoxPro for Windows was nearly done when Microsoft acquired Fox Software. Next, in order to have cross-platform capabilities, some items do not do things the usual Windows way. These, of course, are the things that come back and bite you when you move a FoxPro 2.x app to Windows 95.

Microsoft has prepared a white paper listing the acknowledged problems and workarounds for some of them.

The good news is that most of the problems are visual—involving icons or window characteristics. This means that the meat of your application should continue to work.

Of course, the bad news is that most of the problems are visual—users can see them and it makes your app look less than professional.

The next good news is that there are workarounds for most of them. Unfortunately, there are no workarounds for others, including the one most likely to annoy your users—that the Windows 95 close button is always disabled.

What does all this mean for you? If you're developing custom applications, either in-house or for a particular client, your users are probably willing to live with these inconveniences if the application serves their needs otherwise. For developers of shrink-wrapped applications, the problem is more serious. It's hard to imagine marketing a Windows 95 application that doesn't use the Close button.

So what can you do? First, get your hands on the Microsoft document discussing the problems. By the time you read this, it should be available as a Knowledge Base article.

Second, let Microsoft know about any other problems you run into. They are monitoring the situation.

Finally, Microsoft has not yet decided whether they'll be updating FoxPro 2.6 to address these problems. It's a matter of where to put their resources. Let them know how you feel about it—would you rather they perfected FoxPro 2.6 under Windows 95, improved Visual FoxPro 3.0 or focussed on the next version coming down the pike?