Why 16-bit DOS and Windows are still with us

devblogs.microsoft.com/oldnewthing/20040301-00

March 1, 2004

Raymond Chen

Many people are calling for the abandonment of 16-bit DOS and 16-bit Windows compatibility subsystems. And trust me, when it comes time to pull the plug, I'll be fighting to be the one to throw the lever. (How's that for a mixed metaphor.) But that time is not yet here. You see, folks over in the Setup and Deployment group have gone and visited companies around the world, learned how they use Windows in their businesses, and one thing keeps showing up, as it relates to these compatibility subsystems: Companies still rely on them. Heavily. Every company has its own collection of Line of Business (LOB) applications. These are programs that the company uses for its day-to-day business, programs the company simply cannot live without. For example, at Microsoft two of our critical LOB applications are our defect tracking system and our source control system. And like Microsoft's defect tracking system and source control system, many of the LOB applications at major corporations are not commercial-available software; they are internally-developed software, tailored to the way that company works, and treated as trade secrets. At a financial services company, the trend analysis and prediction software is what makes the company different from all its competitors. The LOB application is the deal-breaker. If a Windows upgrade breaks a LOB application, it's game over. No upgrade. No company is going to lose a program that **is critical to their business**. And it happens that a lot of these LOB applications are 16-bit programs. Some are DOS. Some are 16-bit Windows programs written in some ancient version of Visual Basic. "Well, tell them to port the programs to Win32." Easier said than done.

- Why would a company go to all the effort of porting a program when the current version still works fine. If it ain't broke, don't fix it.
- The port would have to be debugged and field-tested in parallel with the existing system. The existing system is probably ten years old. All its quirks are well-understood. It survived that time in 1998 when there was a supply chain breakdown and when production finally got back online, they had to run at triple capacity for a month to catch up. The new system hasn't been stress-tested. Who knows whether it will handle these emergencies as well as the last system.

- Converting it from a DOS program to a Windows program would incur massive retraining costs for its employees ("I have always used F4 to submit a purchase order. Now I have this toolbar with a bunch of strange pictures, and I have to learn what they all mean." Imagine if somebody took away your current editor and gave you a new one with different keybindings. "But the new one is better.")
- Often the companies don't have the source code to the programs any more, so they couldn't port it if they wanted to. It may use a third-party VB control from a company that has since gone out of business. It may use a custom piece of hardware that they have only 16-bit drivers for. And even if they did have the source code, the author of the program may no longer work at the company. In the case of a missing driver, there may be nobody at the company qualified to write a 32-bit Windows driver. (I know one company that used foot-pedals to control their software.)

Perhaps with a big enough carrot, these companies could be convinced to undertake the effort (and risk!) of porting (or in the case of lost source code and/or expertise, rewriting from scratch) their LOB applications. But it'll have to be a really big carrot. Real example: Just this past weekend I was visiting a friend who lived in a very nice, professionally-managed apartment complex. We had occasion to go to the office, and I caught a glimpse of their computer screen. The operating system was Windows XP. And the program they were running to do their apartment management? It was running in a DOS box.

Raymond Chen

Follow