

Why not just block the apps that rely on undocumented behavior?

 devblogs.microsoft.com/oldnewthing/20031224-00

December 24, 2003



Raymond Chen

Because every app that gets blocked is another reason for people not to upgrade to the next version of Windows. Look at all these programs that would have stopped working when you upgraded from Windows 3.0 to Windows 3.1.

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Compatibility
```

Actually, this list is only partial. Many times, the compatibility fix is made inside the core component for all programs rather than targeting a specific program, as this list does.

(The Windows 2000-to-Windows XP list is stored in your C:\WINDOWS\AppPatch directory, in a binary format to permit rapid scanning. Sorry, you won't be able to browse it easily. I think the Application Compatibility Toolkit includes a viewer, but I may be mistaken.)

Would you have bought Windows XP if you knew that all these programs were incompatible?

It takes only one incompatible program to sour an upgrade.

Suppose you're the IT manager of some company. Your company uses Program X for its word processor and you find that Program X is incompatible with Windows XP for whatever reason. Would you upgrade?

Of course not! Your business would grind to a halt.

“Why not call Company X and ask them for an upgrade?”

Sure, you could do that, and the answer might be, “Oh, you're using Version 1.0 of Program X. You need to upgrade to Version 2.0 for \$150 per copy.” Congratulations, the cost of upgrading to Windows XP just tripled.

And that's if you're lucky and Company X is still in business.

I recall a survey taken a few years ago by our Setup/Upgrade team of corporations using Windows. Pretty much every single one has at least one “deal-breaker” program, a program which Windows absolutely must support or they won’t upgrade. In a high percentage of the cases, the program in question was developed by their in-house programming staff, and it’s written in Visual Basic (sometimes even 16-bit Visual Basic), and the person who wrote it doesn’t work there any more. In some cases, they don’t even *have* the source code any more.

And it’s not just corporate customers. This affects consumers too.

For Windows 95, my application compatibility work focused on games. Games are the most important factor behind consumer technology. The video card that comes with a typical computer has gotten better over time because games demand it. (Outlook certainly doesn’t care that your card can do 20 bajillion triangles a second.) And if your game doesn’t run on the newest version of Windows, you aren’t going to upgrade.

Anyway, game vendors are very much like those major corporations. I made phone call after phone call to the game vendors trying to help them get their game to run under Windows 95. To a one, they didn’t care. A game has a shelf life of a few months, and then it’s gone. Why would they bother to issue a patch for their program to run under Windows 95? They already got their money. They’re not going to make any more off that game; its three months are over. The vendors would slipstream patches and lose track of how many versions of their program were out there and how many of them had a particular problem. Sometimes they wouldn’t even have the source code any more.

They simply didn’t care that their program didn’t run on Windows 95. (My favorite was the one that tried to walk me through creating a DOS boot disk.)

Oh, and that Application Compatibility Toolkit I mentioned above. It’s a great tool for developers, too. One of the components is the Verifier: If you run your program under the verifier, it will monitor hundreds of API calls and break into the debugger when you do something wrong. (Like close a handle twice or allocate memory with GlobalAlloc but free it with LocalAlloc.)

The new application compatibility architecture in Windows XP carries with it one major benefit (from an OS development perspective): See all those DLLs in your C:\WINDOWS\AppPatch directory? That’s where many of the the compatibility changes live now. The compatibility workarounds no longer sully the core OS files. (Not all classes of compatibility workarounds can be offloaded to a compatibility DLL, but it’s a big help.)

Raymond Chen

Follow



