If there is no 16-bit emulation layer in 64-bit Windows, how come certain 16-bit installers are allowed to run?

devblogs.microsoft.com/oldnewthing/20131031-00

October 31, 2013

Raymond Chen

Troy Martin is puzzled by the remark in this knowledge base article that says

> No 16-bit code can run, except for recognized InstallShield and Acme installers (these are hard-coded in Wow64 to allow them to work).

I agree that the sentence is rather confusingly written. It says "No 16-bit code can run, except for this code that runs." But that's not really what's going on. No 16-bit code running at all. What the article is trying to say (as briefly as possible) is that Windows has specific knowledge about certain InstallShield and Acme installers that allows it to parse the installer's data files directly. That's what the article is trying to say with the word *recognized*: When you try to run a 16-bit program, the application compatibility layer looks at the program and says, "Gosh, I wonder if I recognize this program." And maybe the answer is, "Yes, it is InstallShield version 5.0, and I have been taught very specific information about the data files that are used by that version of InstallSheield to the point that I know how to install them *without actually invoking InstallShield itself*." In that case, instead of playing a sad sound, the kernel hands the request to the application compatibility engine with the instructions, "You take care of this." The application compatibility engine then substitutes a 32-bit custom installer that knows how to open, parse, and apply the InstallShield data files. Another way of looking at it is that somebody sat down and ported InstallShield to 32-bit Windows, so that when a user tries to run a 16-bit installer, the request is redirected to the 32-bit version.

You can see that installer in your `C:\Windows\SysWOW64\InstallShield` directory.

Raymond Chen

Follow