Adventures in application compatibility: Querying for an internal interface

devblogs.microsoft.com/oldnewthing/20230113-00

January 13, 2023

Raymond Chen

An application compatibility report came in for a program that crashed during installation. The primary installer created a temporary directory and then ran a secondary installer from that temporary directory. So far, that's standard behavior to avoid operating from a disease-ridden hot tub. What made it more frustrating to debug is that the secondary installer in turn runs a *tertiary* installer, and it was the tertiary installer that was crashing. (At least it stopped at the tertiary installer. Otherwise it's installers all the way down.)

Some debugging of the tertiary installer revealed that it was trying to pin itself to the taskbar automatically on install. This is one of those things that installers love to do, probably second only to silently taking over popular file extensions. This particularly program's approach was to find the vtable for the internal interface that Explorer uses to pin items to the taskbar, and then call the fifth function in the vtable.

The application compatibility issue was that we changed the signature of the fifth function in the vtable, so they were calling the function with the old signature, and the implementation was therefore misinterpreting the parameters, and as a bonus insult, it also corrupted the stack because the number of parameters also changed.

The solution was to restore the function to its original signature and have it just return without doing anything, decorated with comments to ensure that the method stays at the same slot in the vtable for eternity. Meanwhile, the new functionality was moved to a different location. (I'm not telling where.)

Raymond Chen

Follow