Why does Win32 fail a module load if an import could not be resolved?

devblogs.microsoft.com/oldnewthing/20030916-00

September 16, 2003

Raymond Chen

Because we tried it the other way and it was much worse.

In 16-bit Windows, a module that didn't satisfy all its imports would still load. As long as you didn't call a missing import, you were fine. If you did try to call a missing import, you crashed pretty spectacularly with the dreaded Unrecoverable Application Error dialog.

The Win32 folks decided that this was a bad design, because often people would take Fred App, designed for Windows 3.1, and run it on Windows 3.0, and it would run great for about an hour, at which point Fred App would call a function that was available only in Windows 3.1 (like, say, `GetSaveFileName`) and crash as a result.

So the Win32 folks decided that if an import could not be resolved, the app should fail loading. If the makers of Fred App wanted to run on Windows 3.0 after all, they could indicate this by using `GetProcAddress` explicitly. Because if you have to call `GetProcAddress` explicitly, it'll probably occur to you to check the return value.

This issue comes up occasionally when people wish out loud, "Gosh, there should be a way I could mark an import as 'optional' – if it couldn't bind, the load should not fail. It would be the app's responsibility to verify that the bind succeeded before calling it." These people are unwittingly asking for history to repeat itself.

Raymond Chen

Follow