

The gradual erosion of the `SEM_NOOPENFILEERRORBOX` error mode

devblogs.microsoft.com/oldnewthing/20170330-00

March 30, 2017



Raymond Chen

The `SEM_NOOPENFILEERRORBOX` flag has a long history.

It goes back to 16-bit Windows and the way modules were loaded from floppy disks. If the loader could not find the file it needed, it prompted you to insert the disk and gave you the option to retry or cancel. The `SEM_NOOPENFILEERRORBOX` flag lets you disable the prompt and force the system to act as if you had selected *Cancel*.

The 16-bit kernel also exported a function called `OpenFile` that lets you access the routine in the module loader that tries to open a file and prompts for retries if the file isn't found. Therefore, the `SEM_NOOPENFILEERRORBOX` flag controlled the `OpenFile` function, too.

Forward to 32-bit Windows NT. The Windows NT module loader doesn't work anything at all like the 16-bit module loader. Memory management is page-based rather than segment based. Images are memory-mapped and operate on page faults rather than being manually loaded from the disk and operating on segment faults (or worse, segment reload thunks).

This means that the old model of prompting the user to reinsert the floppy disk makes no sense. There is never any prompting for failed module loads. The `SEM_NOOPENFILEERRORBOX` flag has no effect for modules, which means that you won't see any error boxes from `LoadLibrary` on Windows NT. (On Windows 95, if the DLL you were trying to load had a chain of dependencies that led to a missing 16-bit DLL, then you still got the error box from the 16-bit loader.)

This means that today, the `SEM_NOOPENFILEERRORBOX` flag has no effect.

Well, almost.

Remember that `OpenFile` function? The one that let you open a file using the same retry logic as the 16-bit module loader? The function was ported to 32-bit Windows, but of course it doesn't actually use the 16-bit module loader. It just replicates the behavior of the 16-bit module loader.

Including the error prompt if the file cannot be found.

So that's where we are today. The `SEM_NOOPENFILEERRORBOX` flag has been eroded away. The only place that still respects the flag is the old `OpenFile` function, a function you probably didn't even realize existed, and which exists only for backward compatibility with 16-bit programs.

Bonus chatter: I submitted a documentation change request to MSDN to clarify this. We'll see which happens first: This article gets posted or the MSDN change request gets processed.

Raymond Chen

Follow

