

How do I suppress the error box that appears when a LoadLibrary fails?

 devblogs.microsoft.com/oldnewthing/20240208-00

February 8, 2024



Raymond Chen

If there is a problem loading a DLL via `LoadLibrary` or a related function, you sometimes get a system error box that says “Contoso.dll is either not designed to run on Windows or it contains an error. Try installing the program again using the original installation media or contact your system administrator or the software vendor for support.” How can a program suppress this error message and just have `LoadLibrary` return `NULL` without fanfare?

The so-called “error mode” controls how things behave when various categories of bad things happen. One category of failure goes under the name `SEM_FAILCRITICALERRORS`. This name sounds really vague, but at the time it was introduced, it wasn’t vague at all.

The term “critical errors” refers to MS-DOS interrupt 24h, known as the “critical error” interrupt. These errors were things like “No disk in drive” or “I/O error”, and your recovery options were the famous trio of Abort, Retry, and Ignore, later revised to Abort, Retry, and Fail. Saying that you want to “fail critical errors” obviously means that when these critical errors occur, you don’t want the system to prompt the user for action but rather act as if the user had selected “Fail”.

Windows expanded the category of things that were classified as “critical errors” to include “DLL not found”, so that the user could reinsert the floppy disk with the missing DLL. And in the intervening decades, various other miscellaneous file errors got tossed into the “critical errors” bucket. (Sound familiar?)

And that leads to the answer to the question.

To suppress error dialogs in functions like `LoadLibrary`, you can call `SetErrorMode(SEM_FAILCRITICALERRORS)`¹

¹ More precisely, you want to add the `SEM_FAILCRITICALERRORS` flag to whatever flags are already set.