

The importance of error code backwards compatibility

 devblogs.microsoft.com/oldnewthing/20050118-00

January 18, 2005



Raymond Chen

I remember a bug report that came on in an old MS-DOS program (from a company that is still in business so don't ask me to identify them) that attempted to open the file "". That's the file with no name. This returned error 2 (file not found). But the program didn't check the error code and thought that 2 was the file handle. It then began writing data to handle 2, which ended up going to the screen because handle 2 is the standard error handle, which by default goes to the screen. It so happened that this program wanted to print the message to the screen anyway. In other words, this program worked completely by accident. Due to various changes to the installable file system in Windows 95, the error code for attempting to open the null file changed from 2 (file not found) to 3 (path not found) as a side-effect. Watch what happens. The program tries to open the file "". Now it gets error 3 back. It mistakenly treats the 3 as a file handle and writes to it. What is handle 3? The standard MS-DOS file handles are as follows:

handle	name	meaning
0	stdin	standard input
1	stdout	standard output
2	stderr	standard error
3	stdaux	standard auxiliary (serial port)
4	stdprn	standard printer

What happens when the program writes to handle 3? It tries to write to the serial port. Most computers don't have anything hooked up to the serial port. The write hangs. Result: Dead program.

The file system folks had to tweak their parameter validation so they returned error 2 in this case.

[Raymond Chen](#)

Follow

