

# Why is my program terminating with exit code 3?

 [devblogs.microsoft.com/oldnewthing/20110519-00](http://devblogs.microsoft.com/oldnewthing/20110519-00)

May 19, 2011



Raymond Chen

There is no standard for process exit codes. You can pass anything you want to `Exit-Process`, and that's what `GetExitCodeProcess` will give back. The kernel does no interpretation of the value. If you want code 42 to mean "Something infinitely improbable has occurred" then more power to you.

There is a convention, however, that an exit code of zero means success (though what constitutes "success" is left to the discretion of the author of the program) and a nonzero exit code means failure (again, with details left to the discretion of the programmer). Often, higher values for the exit code indicate more severe types of failure. The command processor `ERRORLEVEL` keyword was designed with these convention in mind.

There are cases where your process will get in such a bad state that a component will take it upon itself to terminate the process. For example, if a process cannot locate the DLLs it imports from, or one of those DLLs fails to initialize, the loader will terminate the process and use the status code as the process exit code. I believe that when a program crashes due to an unhandled exception, the exception code is used as the exit code.

A customer was seeing their program crash with an exit code of 3 and couldn't figure out where it was coming from. They never use that exit code in their program. Eventually, the source of the magic number 3 was identified: The C runtime `abort` function terminates the process with exit code 3.

Raymond Chen

**Follow**

