

# Where did WIN32\_LEAN\_AND\_MEAN come from?

---

 [devblogs.microsoft.com/oldnewthing/20091130-00](http://devblogs.microsoft.com/oldnewthing/20091130-00)

November 30, 2009



Raymond Chen

Commenter asdf wonders where WIN32\_LEAN\_AND\_MEAN came from. The `WIN32_LEAN_AND_MEAN` symbol was introduced in the Windows 95 time frame as a way to exclude a bunch of Windows header files when you include `windows.h`. You can take a look at your `windows.h` file to see which ones they are. The symbol was added as part of the transition from 16-bit Windows to 32-bit Windows. The 16-bit `windows.h` header file didn't include all of those header files, and defining `WIN32_LEAN_AND_MEAN` brought you back to the 16-bit Windows philosophy of a minimal set of header files for writing a bare-bones Windows program. This appealed the programmers who liked to micro-manage their header files, and it was a big help because, at the time the symbol was introduced, precompiled header files were not in common use. As I recall, on a 50MHz 80486 with 8MB of memory, switching to `WIN32_LEAN_AND_MEAN` shaved three seconds off the compile time of each C file. When your project consists of 20 C files, that's a whole minute saved right there.

Moore's Law and precompiled headers have conspired to render the `WIN32_LEAN_AND_MEAN` symbol relative useless. It doesn't really save you much any more. But at one point, it did.

[Raymond Chen](#)

**Follow**

