

What is the story behind multimon.h?

devblogs.microsoft.com/oldnewthing/20091207-00

December 7, 2009



Raymond Chen

Commenter asdf wonders what the deal is with that multimon.h header file. Let's set some context. That header file was written back in the time when Windows 98 was still under development. Windows 98 was the first version of Windows to support multiple monitors. At the time, most application authors had Windows 95 as their target platform. And even after Windows 98 shipped, the expectation was that programmers would target both Windows 95 and Windows 98 for at least a little while longer. The problem then is convincing programmers to write their code in a manner that works well in the face of multiple monitors. If you say, "Well, you need to call these functions, but only if running on a version of Windows which supports multiple monitors. If running on an older version of Windows, then you'll need to keep doing things the old way because the functions you want to call don't exist there." Remember, if you want people to do something that doesn't directly benefit them, you need to make it easy or they won't bother. Therefore, the `multimon.h` header file was created which did the dirty work of checking whether it was running on a version of Windows which supported multiple monitors. If so, then it called those functions, and if not, then it emulated them. Applications could then use the `multimon.h` header file and write code on the assumption that the operating system supported multiple monitors: If it didn't, then the stub functions in `multimon.h` would kick in and implement "fake multimonitor support". For example, if you called `GetSystemMetrics(SM_CMONITORS)`, the stub function would simply return 1. If you asked for `GetSystemMetrics(SM_CXVIRTUALSCREEN)`, the stub function just gave you the width of the (one) monitor. Of course, those old versions of Windows which don't support multiple monitors are long obsolete, and there is no longer any need for the `multimon.h` header file. It long ago accomplished its intended goal. Give it a nice pat on the back.

But the header file still lingers in the Platform SDK. If you have a program written back in the days when you couldn't count on multiple monitor support, then removing that header file would result in a build break in your program. That wouldn't be fun.

Raymond Chen

Follow

