

What's the difference between WINVER, _WIN32_WINNT, _WIN32_WINDOWS, and _WIN32_IE?

devblogs.microsoft.com/oldnewthing/20070411-00

April 11, 2007



Raymond Chen

Okay, so there are [all these different ways you can specify what version of the Windows header files you want.](#)[†]

```
#define WINVER          0x0400
#define _WIN32_WINNT    0x0400
#define _WIN32_WINDOWS  0x0400
#define _WIN32_IE       0x0400
```

Let's take them in order.

The `WINVER` symbol is the earliest one. That's the symbol that 16-bit Windows used to control the versioning of its header files, and its use carried forward into the 32-bit header files, presumably from the people who did the initial conversion of the header files to 32-bit and who grew up with the `WINVER` symbol. This symbol is still used a lot in the header files that can trace their origins to 16-bit Windows, such as `winuser.h`, `wingdi.h`, and `mmsystem.h`.

The `_WIN32_WINNT` symbol came next. I'm not sure where it came from, but from its name it probably was invented by the Windows NT team in order to allow them to block off sections of the header file that are available only in the Windows NT implementation of Win32. Don't forget that in the early days, there was also Win32s, a subset of Win32 that could run on 16-bit Windows 3.1. The single `WINVER` symbol wasn't enough to specify exactly what you wanted to be compatible with. For example, a function available only in Windows NT 3.1 would be guarded with `#if _WIN32_WINNT >= 0x030A` so that programs that wanted to run on Win32s could set `_WIN32_WINNT` to zero and keep that function off-limits.

Similarly, both Windows 95 and Windows NT 4 identified themselves as Windows major version 4, so the `WINVER` symbol was insufficient to distinguish them. Functions that existed in Windows NT 4 but not in Window 95 were therefore guarded with `_WIN32_WINNT`.

On the other hand, there were also functions that were first introduced in Windows 95 and did not exist in the original version of Windows NT 4. The `_WIN32_WINDOWS` symbol let you specify that you wanted access to stuff that was new for Windows 95 and which would also be ported to Windows NT 4 and future versions of Windows NT.

The next symbol in this progression is `_WIN32_IE`, which lets you specify what version of Internet Explorer you require to be installed on the system. This was more important back in the days when Internet Explorer included updates to selected operating system components. For example, Internet Explorer 4 came not only with an updated `comctl32.dll` but also a new `shell32.dll` that gave you Active Desktop. (Wow, remember Active Desktop? That was when everybody thought that HTML was going to take over the world and people would write entire applications in HTML. People are still trying.)

And history repeated itself: We saw it before when we tried to puzzle out why some functions return NULL while others return INVALID_HANDLE_VALUE. Each time somebody added a new feature to Windows and had to add an `#ifdef` guard, it was pretty much a toss-up whether they would use `WINVER`, `_WIN32_WINDOWS`, or `_WIN32_WINNT`. Once Internet Explorer stopped including updates to shell components, `_WIN32_IE` fell into the “toss-up” bucket as well.

In an attempt to make some sense out of this disaster, the SDK and DDK teams came up with a new plan for Windows Vista header files: `sdkddkver.h`. There’s now just one symbol you define to specify your minimum target operating system: `NTDDI_VERSION`. Once you set that, all the other symbols are set automatically to the appropriate values for your target operating system. (And no, I don’t know what the letters NTDDI stand for, though there is one obvious candidate.) With any luck, everybody will standardize on `NTDDI_VERSION` and this article will become one of those “quaint historical novelties” like all the ones about 16-bit Windows. Just “a little story about what people had to do back in the crazy days of the early 21st century. Boy am I glad we don’t have to worry about that any more!”

Notes

I’d appreciate it if people would extend me the courtesy of not stepping on my announced topic. (I wonder if these are the same people who go to a comedy show and shout out the punch lines before the performer gets to say them.) I did say that I would pick up the topic today, after all. If you really want to steal my topic, at least be polite enough to post your essay on your own blog.

Nitpicker’s corner

†This list is not intended to be comprehensive.

Raymond Chen

Follow

