

# Why are there two values for PSH\_WIZARD97?

[devblogs.microsoft.com/oldnewthing/20090423-00](http://devblogs.microsoft.com/oldnewthing/20090423-00)

April 23, 2009



Raymond Chen

*Welcome, Slashdot readers. Remember, this Web site is for entertainment purposes only.*

If you go cruising through the `commctrl.h` header file, you'll find that there are two different definitions for `PSH_WIZARD97` :

```
#if (_WIN32_IE >= 0x0400)
...
#if (_WIN32_IE < 0x0500)
#define PSH_WIZARD97          0x00002000
#else
#define PSH_WIZARD97          0x01000000
#endif
...
```

Why are there two values for `PSH_WIZARD97` ?

Set the wayback machine to 1997. (Note: Dates are approximate.) The user interface design team studied how Windows 95 wizards have been faring and have begun using what they've learned to develop the next version of the wizard interface, which they have code-named "Wizard 97". Development in `comctl32.dll` proceeds based their designs, what with exterior and interior pages, headers and sub-headers, watermarks, all that jazz. Meanwhile, work on Internet Explorer 4 reaches the homestretch, and it's a race to the finish line.

Internet Explorer 4 wins the race and is ready to ship before the work on Wizard 97 is complete. What do you do? Do you say "Um, hi, Internet Explorer team. Could you slip your schedule by about half a year so we can finish our work on this new wizard design?" Yeah, like the Internet Explorer team are going to say, "Oh, sure, no hurry, take your time. We'll wait. It's not like we're going to get ridiculed for being another six months late."

And monkeys might fly out of my butt.

Out the door it goes, with a Wizard 97 design that hadn't gone through a full usability review. To discourage programmers from using this half-finished wizard, put the sentence "Don't use this flag" in the documentation. Boy, that was a close call.

Since they already “missed the boat”, the urgency to complete the work on Wizard 97 vanished. One of the things that came up during the usability review was that the full-page watermark wasn’t such a great idea after all. It made the text hard to read, users had difficulty figuring out what they should do, various problems like that. The design for Wizard 97 changed; now the watermark was just a small picture that goes in the corner of the wizard. Of course, the changes weren’t just design changes; there were also bug fixes. All the work was done in time for Windows 2000, yippee.

And then the compatibility bugs started coming in.

Wait, compatibility bugs? How can there be compatibility bugs? Nobody is using Wizard 97 yet; there’s nobody to be compatible with!

Well, it turns out that programs were indeed using the `PSH_WIZARD97` flag from Internet Explorer 4 and expecting to get that old half-baked Wizard 97 design. (After some investigation, I believe we discovered that somebody somewhere had told the MFC folks, “Hey, there’s this cool Wizard 97 thing you should use. It’s pretty awesome.” That someone was overcome with excitement and didn’t realize that the work on Wizard 97 was not yet finished and that Wizard 97 was not yet ready for prime time. People who used MFC to write their programs then said, “Hey, look, MFC has support for this new type of wizard!” and started using it.)

Oh, great, how do you fix this?

It wasn’t pretty, but it had to be done. (The preferred first choice, using a time machine, was unavailable.) The fix was to have two types of Wizard 97; there’s the “Internet Explorer 4 version” and the “Windows 2000 version”. Of course, nobody wants to officially admit that the Internet Explorer 4 version exists, since during its brief lifetime, it was documented as “Don’t use me!” All the changes that were made after Internet Explorer 4 shipped had to be revisited. Each time there was a code change (either to support a new design or just to fix a bug), a test against the wizard flavor had to be made, and the old behavior restored if the old flag was used.

If new code was added for the Windows 2000 version of Wizard 97, whether it be due to a design change or a bug fix, then it had to be placed behind a check:

```
if (program asked for Windows 2000-style Wizard 97) {  
    // new behavior here  
}
```

Conversely, if code was deleted, it had to be brought back with a version check:

```
if (program asked for Internet Explorer 4-style Wizard 97) {  
    // old behavior here  
}
```

And if behavior changed, you need to have both versions:

```
if (program asked for Internet Explorer 4-style Wizard 97) {  
    // old behavior here  
} else {  
    // new behavior here  
}
```

Giving the old and new Wizard 97 flags different values allows the property sheet wizard code to tell whether the program was designed for the bogus Internet Explorer 4 pre-release version of Wizard 97 or the finished Windows 2000 version.

Raymond Chen

**Follow**

