

What was the purpose of the hPrevInstance parameter to WinMain?

 devblogs.microsoft.com/oldnewthing/20040615-00

June 15, 2004



Raymond Chen

Once your average GUI program picks itself up off the ground, control begins at your WinMain function. The second parameter, hPrevInstance, is always zero in Win32 programs. Certainly it had a meaning at some point? Of course it did. In 16-bit Windows there was a function called GetInstanceData. This function took an HINSTANCE, a pointer, and a length, and copied memory from that instance into your current instance. (It's sort of the 16-bit equivalent to ReadProcessMemory, with the restriction that the second and third parameters had to be the same.) (Since 16-bit Windows had a common address space, the GetInstanceData function was really nothing more than a hmemcpy, and many programs relied on this and just used raw hmemcpy instead of using the documented API. Win16 was actually designed with the possibility of imposing separate address spaces in a future version – observe flags like GMEM_SHARED – but the prevalence of tricks like hmemcpy'ing your previous instance reduced this potential to an unrealized dream.) This was the reason for the hPrevInstance parameter to WinMain. If hPrevInstance was non-NULL, then it was the instance handle of a copy of the program that is already running. You can use GetInstanceData to copy data from it, get yourself up off the ground faster. For example, you might want to copy the main window handle out of the previous instance so you could communicate with it. Whether hPrevInstance was NULL or not told you whether you were the first copy of the program. Under 16-bit Windows, only the first instance of a program registered its classes; second and subsequent instances continued to use the classes that were registered by the first instance. (Indeed, if they tried, the registration would fail since the class already existed.) Therefore, all 16-bit Windows programs skipped over class registration if hPrevInstance was non-NULL. The people who designed Win32 found themselves in a bit of a fix when it came time to port WinMain: What to pass for hPrevInstance? The whole module/instance thing didn't exist in Win32, after all, and separate address spaces meant that programs that skipped over reinitialization in the second instance would no longer work. So Win32 always passes NULL, making all programs believe that they are the first one.

And amazingly, it actually worked.

Raymond Chen

Follow

