

If `InitCommonControls` doesn't do anything, why do you have to call it?

 devblogs.microsoft.com/oldnewthing/20050718-16

July 18, 2005



Raymond Chen

One of the problems beginners run into when they start using shell common controls is that they forget to call the `InitCommonControls` function. But if you were to disassemble the `InitCommonControls` function itself, you'll see that it, like the `FlushInstructionCache` function, doesn't actually do anything.

Then why do you need to call it?

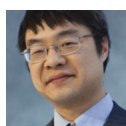
As with `FlushInstructionCache`, what's important is not what it performs, but just the fact that you called it.

Recall that merely listing a lib file in your dependencies doesn't actually cause your program to be bound to the corresponding DLL. You have to call a function in that DLL in order for there to be an import entry for that DLL. And `InitCommonControls` is that function.

Without the `InitCommonControls` function, a program that wants to use the shell common controls library would otherwise have no reference to `COMCTL32.DLL` in its import table. This means that when the program loads, `COMCTL32.DLL` is not loaded and therefore is not initialized. Which means that it doesn't register its window classes. Which means that your call to the `CreateWindow` function fails because the window class has not been registered.

That's why you have to call a function that does nothing. It's for your own good.

(Of course, there's the new `InitCommonControlsex` function that lets you specify which classes you would like to be registered. Only the classic Windows 95 classes are registered when `COMCTL32.DLL` loads. For everything else you have to ask for it explicitly.)



Raymond Chen

Follow

