

If your callback fails, it's your responsibility to set the error code

 devblogs.microsoft.com/oldnewthing/20060123-12

January 23, 2006



Raymond Chen

There are many cases where a callback function is allowed to halt an operation. For example, you might decide to return `FALSE` to the `WM_NCCREATE` message to prevent the window from being created, or you might decide to return `FALSE` to one of the many enumeration callback functions such as the `EnumWindowsProc` callback. When you do this, the enclosing operation will return failure back to its caller: the `CreateWindow` function returns `NULL`; the `EnumWindows` function returns `FALSE`. Of course, when this happens, the enclosing operation doesn't know why the callback failed; all it knows is that it failed. Consequently, it can't set a meaningful value to be retrieved by the `GetLastError` function. If you want something meaningful to be returned by the `GetLastError` function when your callback halts the operation, it's the callback's responsibility to set that value by calling the `SetLastError` function.

This is something that is so obvious I didn't think it needed to be said; it falls into the "because computers aren't psychic (yet)" category of explanation. But apparently it wasn't obvious enough, so now I'm saying it.

Raymond Chen

Follow

