

What's the difference between EM_UNDO and WM_UNDO?

 devblogs.microsoft.com/oldnewthing/20070911-00

September 11, 2007



Raymond Chen

Daniel Chait wonders why we have both EM_UNDO and WM_UNDO. You know, I wonder the same thing. But I'm going to make an educated guess. Actually, most of what I write is just a lot of educated guessing. Like my explanation of why GetWindowText has such complicated rules? A guess. Why address space granularity is 64KB? A guess. Why most EM * messages are in the system message range? A guess. Mind you, it's logical guesswork, usually strongly guided by the principle of "Imagine if this were possible." Today's guesswork: The history of `EM_UNDO` and `WM_UNDO`. It seems obvious that `EM_UNDO` came first. After all, why would there need to be a `EM_UNDO` message if a `WM_UNDO` already existed? At some point, somebody decided, "Hey, this sounds like something that people might want to do more generally." New messages `WM_CUT`, `WM_COPY`, `WM_PASTE`, `WM_CLEAR`, and `WM_UNDO` were added, and the first control to implement them was the edit control. Therefore, the answer to the question is that for edit controls, the difference is that there is no difference.

The window manager does not provide a default implementation for any of these new messages. (Obviously, because the window manager is not psychic.) If you want your control to support those operations, you'll have to respond to the messages yourself.

Raymond Chen

Follow

