What's the deal with the EM_SETHILITE message?

devblogs.microsoft.com/oldnewthing/20071025-00

October 25, 2007

Raymond Chen

If you look up the documentation on the `EM_SETHILITE` and `EM_GETHILITE` messages, they just say “not implemented”. What’s going on here? The `EM_SETHILITE` and `EM_GETHILITE` messages were added back in 2002 for the breadcrumb bar to use. Back in those days, the breadcrumb bar wasn’t what you see in Windows Vista today, a series of buttons with drop-down arrows, each representing a level in the hierarchy. The breadcrumb bar back in those days was just a fancy-looking address bar (i.e., an edit control). Instead of having a button for each level in the hierarchy, the breadcrumb bar would highlight a path component if you moved the mouse over it (or used the keyboard to select it). Instead of having drop-down arrows, the breadcrumb bar would wait for you to hover your mouse and the highlighted path would auto-open to show the menu of sibling items. The design of the breadcrumb bar changed and the “fancy-looking address bar” design was abandoned. The functionality of the `EM_SETHILITE` and `EM_GETHILITE` messages was ripped out, but the message numbers were left behind to avoid build breaks in case anybody was using those messages. Of course the messages have no effect any more, but at least the product still builds. The people who were using the message can then wean themselves from it on their own schedule. This is one of the realities of developing both a library and the library’s clients simultaneously. If you are the person who is responsible for the library and you intend to make a breaking change, you have to do it in stages so that all the clients can make the transition smoothly. It’s not like you can go in and change all the clients yourself since they aren’t yours to change, and you might not have access to the source code anyway. The clients belong to other teams, or even other companies. You have to coordinate with those other teams and let them know what your transition plan is so they can work it into their schedule.

The fact that the `EM_GETHILITE` and `EM_SETHILITE` messages still linger is just leftover dirt that never got cleaned up. What’s in the header files now is the penultimate stage of the transition plan. The final stage would have been to remove the messages from the header file entirely, but the “no changes to public header files allowed” deadline came before the final stage could be executed, and now that the messages are in a public header file, they have to stay in there forever. Even though they don’t do anything.

Raymond Chen

Follow