What were Get/SetMessageExtraInfo ever used for?

devblogs.microsoft.com/oldnewthing/20101129-00

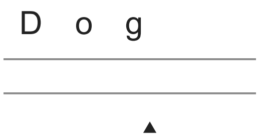
November 29, 2010

Raymond Chen

KJK::Hyperion asks, “Could you shed some light on Get/SetMessageExtraInfo? It’s almost like nobody on earth used them, ever, and I can’t get some sample code.” Yup, that’s about right. Nobody on earth (to within experimental error) ever used them. These functions were introduced on July 20, 1990 (I’m looking at the change history right now) at the request of what was then called the Hand-Writing Windows group, which shipped the first version of Windows for Pen Computing in 1992. The idea was that each input event from the custom pen hardware would have this extra information associated with it, and the software that converted pen input into strokes (and ultimately into gestures or characters via handwriting recognition) would use this extra information to guide the conversion process. Seeing as Pen Windows died a hasty death, I think it’s fairly accurate to say that nobody on earth will admit to having used these functions. For those of you fortunate enough never to have been exposed to Pen Windows, here are some random tidbits of information. First, applications needed to be modified to support pen input. In particular, edit controls did not accept text input from the pen. You had to replace them with one of the following:

- Handwriting edit control (hedit). This accepted free form handwriting input.
- Boxed edit control (bedit). This accepted handwriting input, but you had to write one letter per box. This constraint resulted in much better character recognition.

Both of these controls were significantly larger than the standard edit control. They needed to be, in order to give enough room for the user to write. This in turned means that you had to edit all your dialog templates and custom window layout to take into account the larger pen-aware controls. And just changing your controls wasn’t enough. You also had to write extra code to call various character recognition functions to get the user’s pen input converted and recognized. Here’s an artist’s conception of what the boxed edit control looked like:



That weird triangle-shaped thingie was, I believe, called the *dinky*. What did it do? Beats me. There are still vestiges of the old Pen Windows product in the `GetSystemMetrics` function: Check out `SM_PENWINDOWS`. (Note that the old Pen Windows product is unrelated to the current Tablet PC product, even though they both do handwriting recognition.) **Bonus chatter**: The Windows touch team saw their opportunity and commandeered the extra info (perhaps resurrecting the ghost of Pen Windows) and use the extra info to specify the source of the input.

[Raymond is currently away; this message was pre-recorded.]

Raymond Chen

Follow