Broadcasting user-defined messages

devblogs.microsoft.com/oldnewthing/20040505-00

May 5, 2004

Raymond Chen

When you broadcast a message (via HWND_BROADCAST) remember that the message you broadcast must have global meaning. I discussed earlier what the various message ranges mean. Notice that only the system-defined range (0..WM_USER-1) and the registered message range (MAXINTATOM .. MAXWORD) have global meaning. The other two ranges have class-specifc or application-specific meanings. In other words, you can't broadcast a message in the WM_USER range since that message has a different meaning for each window class. Similarly, a message in the WM_APP range has a different meaning for each application. We ran into this problem in Windows 95. There were programs that decided to broadcast private messages like WM_USER+0x0100, intending them to be delivered to other instances of that program. Of course, when those messages reached some other windows, they interpreted WM_USER+0x0100 as some other private message and either acted funny or crashed. On the other hand, the programs really wanted the message to reach the windows of other copies of itself, so we couldn't just block the broadcast of the programs would stop working. Progams were relying on the system not trying to stop them from crashing other programs! The solution was to split the difference. If you broadcast a message that was not safe to broadcast, Windows 95 would send it only to old-style programs. New-style programs (marked as version 4.0 or higher) would not receive the messages. That way, old programs continued to affect each other as they always did, but new programs followed the new rules.

Moral of the story: When you broadcast a message, make sure it's one that every receiving window will be able to handle.

Raymond Chen

Follow