

Which message numbers belong to whom?

 devblogs.microsoft.com/oldnewthing/20031202-00

December 2, 2003



Raymond Chen

Valid window messages break down into four categories.

0 .. 0x3FFF (WM_USER-1): System-defined messages.

The meanings of these messages are defined by the operating system and cannot be changed. Do not invent new messages here. Since the meanings are defined by Windows, the operating system understands how to parse the WPARAM and LPARAM parameters and can marshal the messages between processes (or knows to refuse to do so).

0x400 .. 0x7FFF (WM_USER .. WM_APP-1): Class-defined messages.

The meanings of these messages is determined by the implementor of the window class. (Informally: By the person who calls RegisterClass for that window class.) For example, the WM_USER+1 message means TB_ENABLEBUTTON if the window is a toolbar control, but it means TTM_ACTIVATE if it is a tooltip control, and it means DM_SETDEFID if it is a dialog box. If you created your own control, it would mean something else completely different. Since anybody can create a message in this range, the operating system does not know what the parameters mean and cannot perform automatic marshalling.

0x8000 .. 0xBFFF (WM_APP ... MAXINTATOM-1): Application-defined messages.

The meanings of these messages is determined by the application that created the window. (Informally: By the person who calls CreateWindow.) This message region was created in Windows 95 to ensure that applications which subclass a window and generate custom messages will not interfere with new messages created by the window class in future versions. Again, since anybody can create a message in this range, the operating system does not know what the parameters mean and cannot perform automatic marshalling.

0xC000 .. 0xFFFF (MAXINTATOM .. MAXWORD): Registered messages.

The meanings of these messages is determined by the caller of RegisterWindowMessage. Note that the numerical value of registered messages can change from run to run, so you must use RegisterWindowMessage to obtain the message number. Once again, since anybody can create a message in this range, the operating system does not know what the parameters mean and cannot perform automatic marshalling.



Raymond Chen

Follow