

Pitfalls in handling the WM_CONTEXTMENU message

 devblogs.microsoft.com/oldnewthing/20040921-00

September 21, 2004



Raymond Chen

Before we continue with our IContextMenu discussion, I need to take a little side trip and discuss the subtleties of the WM_CONTEXTMENU message.

First, a correction to the existing <windowsx.h> header file:

```
#undef HANDLE_WM_CONTEXTMENU
#define HANDLE_WM_CONTEXTMENU(hwnd, wParam, lParam, fn) \
    ((fn)((hwnd), (HWND)(wParam), GET_X_LPARAM(lParam), GET_Y_LPARAM(lParam)), 0L)
```

Apparently, HANDLE_WM_CONTEXTMENU was overlooked when the <windowsx.h> header file gained multimonitor support.

The second subtlety of the WM_CONTEXTMENU message is the recognition that context menus can be invoked from the keyboard, not just by the mouse. If you have a 104-key keyboard, you will probably have a menu key to the right of your space bar. (Laptop owners: You're on your own. Laptop keyboards are hardly standardized.) Alternatively, you can type Shift+F10 to get the same effect.

When the user invokes a context menu from the keyboard, the x and y coordinates are both -1. In this case, you should display the context menu for the currently-selected item (or items, if a multiple selection is active). If you miss this detail, then you will end up hit-testing against (-1, -1) and probably not find anything.

Okay, now that these remarks on the WM_CONTEXTMENU message are out of the way, we can return to our discussion of the IContextMenu interface next time.

Raymond Chen

Follow

