

# How can I detect which menu my item was invoked from?

 devblogs.microsoft.com/oldnewthing/20241121-00

November 21, 2024



A customer had a Win32 program with a menu bar, a context menu that they display on mouse right-click, and the same context menu that they display on a keyboard-initiated context menu (**Shift** + **F10** or the dedicated context menu key on some keyboards), and they all have the same menu item, call it *Create bookmark*. For good measure, let's say that they also had a keyboard accelerator for *Create bookmark*.

The customer wanted the *Create bookmark* command to work like this:

- If invoked from the menu bar, a keyboard-initiated context menu, or a keyboard accelerator, then create a bookmark at the current selected item.
- If invoked from a mouse-initiated context menu, then create a bookmark at the position that the user right-clicked.

The `wParam` of the `WM_COMMAND` message indicates whether the message was initiated from a menu (0) or a keyboard accelerator (1). But nothing that lets you distinguish the three menu sources. So what can we do to tell us the source of the message?

This is just a variation of "[If I register the same shell extension as both a context menu extension and a drag/drop extension, how do I know which one the system is using?](#)" Just that you're using menu items instead of COM class IDs.

The solution is the same: Since the system doesn't tell you the source of the message, you can just give each source a different ID number. The one on the menu bar could be `IDM_CREATE_BOOKMARK_MENUBAR`, the one on the mouse-triggered context menu could be `IDM_CREATE_BOOKMARK_MOUSEMENU`, the one on the keyboard-triggered context menu could be `IDM_CREATE_BOOKMARK_KEYBOARDMENU`, and the one from the keyboard accelerator could be `IDM_CREATE_BOOKMARK_KEYBOARDACCELERATOR`. Now you can use that number to distinguish each of the sources.<sup>1</sup>

Now, since you have only two behaviors, you could collapse it to just two IDs:

- `IDM_CREATE_BOOKMARK_AT_SELECTION`.
- `IDM_CREATE_BOOKMARK_AT_MOUSE`.

But wait, this whole problem could be solved in a simpler way: Make right-clicking the mouse select the clicked-on item! This is what Explorer does, it's what edit controls do, it's what list view controls do, it's pretty much standard behavior across most container controls, analogous to the principle that "double-clicking should be an extension of single-clicking."

Once you make right-clicking select the clicked-on item, then the single menu item can simply create a bookmark at the selection. Furthermore, you no longer have to remember the mouse click location somewhere, to refer back to when the user selects the "create bookmark at mouse" command.

<sup>1</sup> I called this the "Fred Tire" solution, but you see this often on URLs: If you want to know which advertising campaign sent a potential customer to your Web site, you can give each campaign a slightly different URL, like <https://contoso.com/?source=1>, <https://contoso.com/?source=2>, and <https://contoso.com/?source=3>. All three of the URLs show the same landing page, but you can use the `source=` to figure out which ad campaign the URL came from.<sup>2</sup>

<sup>2</sup> I used to work for a very small company that printed slightly different versions of each advertisement they placed. When somebody called the company, the receptionist asked, "Can you read me the number in the bottom right corner of the advertisement you saw?" This was surprisingly effective.