

How do the menu functions find items?

 devblogs.microsoft.com/oldnewthing/20070319-00

March 19, 2007



Raymond Chen

Most of the menu item functions such as `GetMenuItemInfo` allow you specify the menu item either by position or by command. Some of them use the `MF_BYPOSITION` and `MF_BYCOMMAND` flags. Others separate the search algorithm into a separate `fByPosition` flag. Searching for menu items by position is straightforward: The specified position is used as a zero-based index into the menu. In other words, the first item in the menu is item zero. Searching for menu items by command is trickier. The menu manager searches the entire menu hierarchy, **including submenus**, for an item with the command you specify. If more than one menu item has the identifier you requested, then one of them is chosen arbitrarily. Searching the hierarchy for a command means that you can, for example, remove or disable a menu item by just passing the root menu (which you typically have easy access to) and the item identifier. If the submenus were not searched, then synchronizing menu states would be a much more cumbersome affair. But what if your menu has multiple items with the same identifier? Well, the short answer is, “Then don’t use `MF_BYCOMMAND`.” You can still use `MF_BYPOSITION` to access your menu items. But why would you have multiple items with the same identifier in the first place? When the user selects the menu, a `WM_COMMAND` is posted to the window with the menu identifier as one of its parameters, and none of the other parameters gives you the menu handle. If you have multiple menu items with the same identifier, you won’t be able to tell which of them the user picked! There was [an emotional discussion a while back](#) that generated far more heat than light. But I can use my psychic powers to explain what that person was seeing, even though not enough information was provided in the original problem description, and it’s not a bug in Windows. Addressing the original complaint: If you have a menu with more than one item with the same identifier, then `MF_BYCOMMAND` is ambiguous, and all that is promised is that **some** item with that identifier will be found. It might not be the one you wanted, but since you gave multiple items the same name, the menu manager did the best it could. This is analogous to other searching functions like `FindWindow` and `GetDlgItem`, which operate on the first item they find. If multiple items match the criteria you specify, it just returns one of them.

As for the specific problem: It so happens that there is no cache, at least not yet. (But who knows, there might be a cache in the future if we discover that lots of applications query for the same item in rapid succession.) But what this person didn’t realize is that the unnamed

custom GUI library they're using doesn't create submenus until the `WM_INITMENUPOPUP` message is received. The reason why the item isn't found before the submenu is opened is that until the submenu is opened, **the submenu doesn't exist**. And naturally, you can't find something that doesn't exist.

Raymond Chen

Follow

