WindowFromPoint, ChildWindowFromPoint, RealChildWindowFromPoint, when will it all end?

devblogs.microsoft.com/oldnewthing/20101230-00

December 30, 2010

Raymond Chen

Oh wait, there's also `ChildWindowFromPointEx`. There are many ways of identifying the window that appears beneath a point. The documentation for each one describes how they work, but I figured I'd do a little compare/contrast to help you decide which one you want for your particular programming problem. The oldest functions are `WindowFromPoint` and `ChildWindowFromPoint`. The primary difference between them is that `WindowFromPoint` returns the deepest window beneath the point, whereas `ChildWindowFromPoint` returns the shallowest. What do I mean by deep and shallow? Suppose you have a top-level window P and a child window C. And suppose you ask one of the above functions, "What window is beneath this point?" when the point is squarely over window C. The `WindowFromPoint` function looks for the most heavily nested window that contains the point, which is window C. On the other hand `ChildWindowFromPoint` function looks for the least nested window that contains the point, which is window P, assuming you passed `GetDesktopWindow` as the starting point. That's the most important difference between the two functions, but there are others, primarily with how the functions treat hidden, disabled, and transparent windows. Some functions will pay attention to hidden, disabled, and/or transparent windows; others will skip them. Note that when a window is skipped, the entire window hierarchy starting from that window is skipped. For example, if you call a function that skips disabled windows, then all children of disabled windows will also be skipped (even if the children are enabled). Here we go in tabular form.

Function	Search	Hidden?	Disabled?	Transparent?[1]
`WindowFromPoint`	Deep	Skip	Skip	It's Complicated[2]
`ChildWindowFromPoint`	Shallow	Include	Include	Include
`ChildWindowFromPointEx`	Shallow	Optional	Optional	Optional
`RealChildWindowFromPoint`	Shallow	Skip	Include	Include[3]

The return values for the various `...FromPoint...` functions are the same:

- Return the handle of the found window, if a window was found.
- Return the handle of the parent window if the point is inside the parent window but not inside any of the children. (This rule obviously does not apply to `WindowFromPoint` since there is no parent window passed into the function.)
- Otherwise, return `NULL`.

The entries for `ChildWindowFromPointEx` are marked *Optional* because you, the caller, get to specify whether you want them to be skipped or included based on the `CWP_*` flags that you pass in. [1]There is a lot hiding behind the word *Transparent* because there are multiple ways a window can be determined transparent. The `...ChildWindowFromPoint...` functions define transparent as *has the `WS_EX_TRANSPARENT` extended window style*. [2]On the other hand, `WindowFromPoint` defines transparent as *returns `HTTRANSPARENT` in response to `WM_NCHITTEST`*. Actually, that's still not true. If the window belongs to a ~~process~~ thread different from the one calling `WindowFromPoint`, then `WindowFromPoint` will not send the message and will simply treat the window as opaque (*i.e.*, not transparent). [3]The `RealChildWindowFromPoint` includes transparent windows in the search, but has a special case for group boxes: The `RealChildWindowFromPoint` function skips over group boxes, *unless* the return value would have been the parent window, in which case it returns the group box after all. Why is `RealChildWindowFromPoint` so indecisive? The `RealChildWindowFromPoint` function was added as part of the changes to Windows to support accessibility. The intended audience for `RealChildWindowFromPoint` is accessibility tools which want to return a "reasonable" window beneath a specific point. Since group boxes usually enclose other controls, `RealChildWindowFromPoint` prefers to return one of the enclosed controls, but if the point belongs to the group box frame, then it'll return the group box. One place I see confusion over the various `...WindowFromPoint...` functions is code which uses one of the functions, and then massages the result, unaware that there is already a function that returns the pre-massaged result for you. For example, I've seen code which calls `WindowFromPoint` followed by `GetAncestor(GA_ROOT)`. This does a pointless down-and-up traversal of the window tree, searching for the deepest window that lies beneath the specified point, then walking back up the tree to convert it to a shallow window. This is the Rube Goldberg way of calling `ChildWindowFromPointEx(GetDesktopWindow(), ...)`.

Next time, a look at the mysterious `RealGetWindowClass` function. What makes this function more real?

Raymond Chen

Follow