Which windows appear in the Alt+Tab list?

devblogs.microsoft.com/oldnewthing/20071008-00

October 8, 2007

Raymond Chen

Commenter Phil Quirk wants to know what the rules are for determining which windows appear in the Alt+Tab list. It's actually pretty simple although hardly anything you'd be able to guess on your own. **Note**: The details of this algorithm are an implementation detail. It can change at any time, so don't rely on it. In fact, it already changed with Flip and Flip3D; I'm just talking about the Classic Alt+Tab window here.

For each visible window, walk up its owner chain until you find the root owner. Then walk back down the visible last active popup chain until you find a visible window. If you're back to where you're started, then put the window in the Alt+Tab list. In pseudo-code:

```
BOOL IsAltTabWindow(HWND hwnd)
{
 // Start at the root owner
 HWND hwndWalk = GetAncestor(hwnd, GA_ROOTOWNER);
 // See if we are the last active visible popup
 HWND hwndTry;
 while ((hwndTry = GetLastActivePopup(hwndWalk)) != hwndTry) {
  if (IsWindowVisible(hwndTry)) break;
  hwndWalk = hwndTry;
 }
 return hwndWalk == hwnd;
}
```

The purpose of this algorithm is to assign the most meaningful representative winow from each cluster of windows related by ownership. (Notice that the algorithm doesn't care whether the owned window is modal or non-modal.)

At least that's the simple rule if you're not playing crazy window style games. The `WS_EX_TOOLWINDOW` and `WS_EX_APPWINDOW` extended styles were created so people can play games and put their window in the Alt+Tab list or take it out even if the simple rule would normally have decided otherwise. This is one of those "Okay, if you think you're smarter than Windows, here's your chance to prove it" options. Personally, I would avoid them since it makes your window behave differently from the rest of the windows in the system.

A window with the `WS_EX_TOOLWINDOW` extended style is treated as if it weren't visible, even if it is. A window with the `WS_EX_APPWINDOW` extended style is treated as if it has no owner, even if it does.

Once you start adding these extended styles, you enter the world of "I'm trying to work around the rules" and the result is typically even worse confusion than what you had without them.

I'm not sure what the original commenter is getting at. The window hierarchy described in the suggestion (which doesn't make it so much a suggestion as it is a request for me to debug their problem) says that window C is modal on both windows A and B, which doesn't make sense to me, since a window has only one owner.

The algorithm for choosing the Alt+Tab representative from each cluster of windows may not be the best, but it's what we have. I wouldn't be surprised if the details are tweaked from time to time. No, wait, let me rephrase that. I *know* that the details are tweaked from time to time. The spirit of the operation is preserved (to show the windows the user can switch to, using the most "natural" candidate for each cluster of windows), but the specific details may be fined-tuned as the concept of "naturalness" is refined.

Raymond Chen

Follow