

How am I supposed to create children of the Win32 tab control?

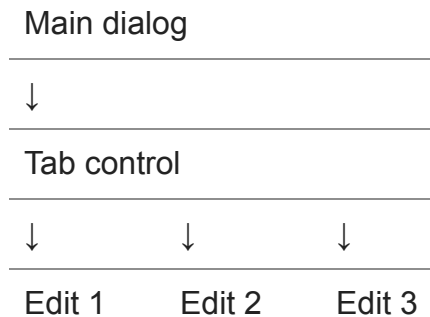


Raymond Chen

Say you're trying to write your own tabbed dialog. You start with the Win32 tab control to get the tabs at the top of the window. But how do you position the controls that belong to each of the pages?

Let's say, for simplicity, that your tab control has three tabs, and they let you select among three different edit controls.

Your natural inclination would be to make the page controls children of the tab control:

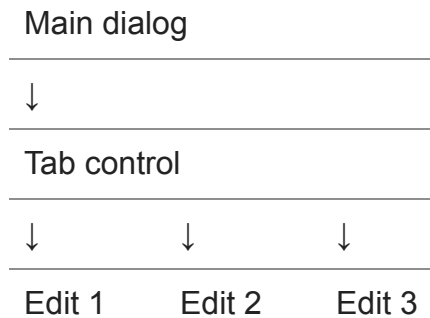


When the user is on a particular tab, you show that tab's edit control and hide the other two.

The problem with this arrangement is that navigating the dialog with the **Tab** key doesn't work.

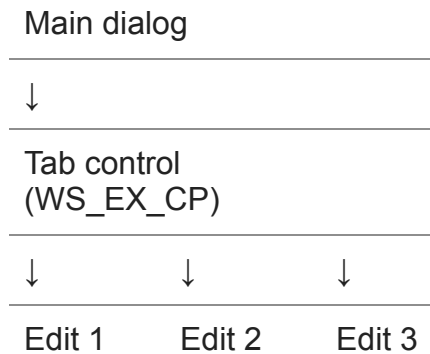
Recall that the windows which participate in the tab order are those which are immediate children of the dialog frame, but with the special rule that if any of those children is marked `WS_EX_CONTROLPARENT`, then it disappears from the tab order, replaced by its children. The `WS_EX_CONTROLPARENT` rule applies recursively, so if any of the children have the style, then they too disappear and are replaced by their own children.

In the above diagram, if we assume that none of the windows is marked as `WS_EX_CONTROLPARENT`, then the tab order consists only of the tab control itself. You can't tab into the children.



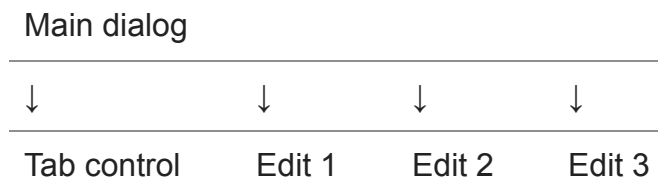
I've highlighted in blue the windows that participate in the tab order. By default, the windows that participate in the tab order are the immediate children of the main dialog, which in this case is the tab control.

On the other hand, if you mark the tab control as `WS_EX_CONTROLPARENT`, then you can tab into the children, but the tab control itself disappears from the tab order. The `WS_EX_CONTROLPARENT` style means "Replace me in the tab order with my children."



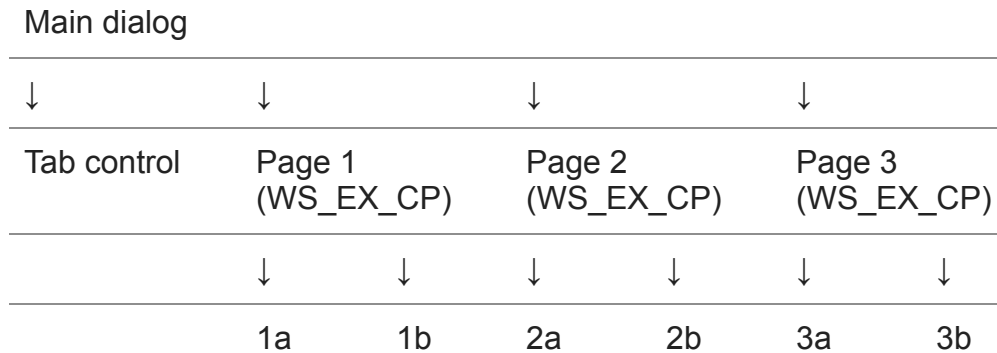
Win32 doesn't let a control and its children all participate in the tab order. You have to choose whether you want the control or its children, but not both.

The solution is to take the page controls out from under the shadow of the tab control.



Setting the controls as siblings of each other gets them all into the tab order. We set the edit controls at a higher z-order, so they draw on top of the tab control.

In practice, you probably have a lot of controls on each page, not a single edit control. Hiding and showing all of those controls can be a hassle, but you can use `WS_EX_CONTROLPARENT` to save you some effort:



Wrap each group of controls inside a parent page control, and mark the page control as `WS_EX_CONTROLPARENT`. This is easy to arrange by making the page controls themselves be dialog boxes. That also gives you the convenience of having a separate dialog procedure for each page.

You can then hide or show an entire group of controls by hiding or showing the page parent. This also has the advantage of preserving the hide/show states of the inner controls, so if the rule was that control 1b is shown only if control 1a is checked, then hiding page 1 will hide everything, and then showing page 1 will bring them back in the same state they were in as they were previously.

Raymond Chen

Follow

