

When embedding a dialog inside another, make sure you don't accidentally create duplicate control IDs

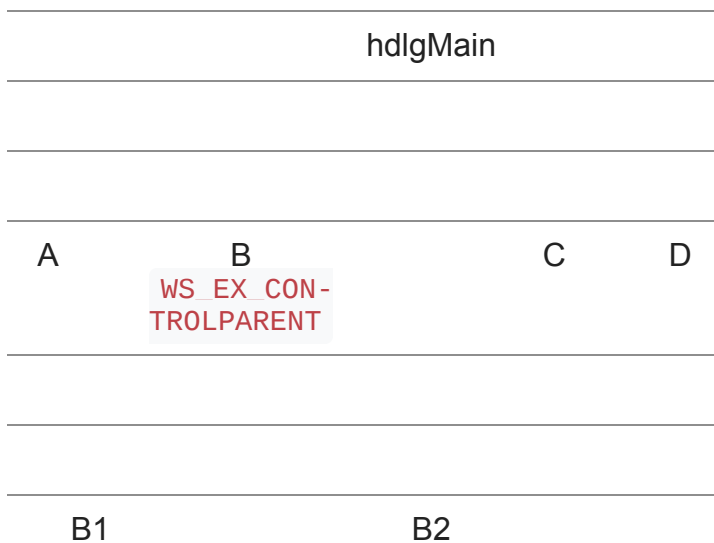
 devblogs.microsoft.com/oldnewthing/20120620-00

June 20, 2012



Raymond Chen

The `WS_EX_CONTROLPARENT` extended style (known in dialog templates as `DS_CONTROL`) instructs the dialog manager that the dialog's children should be promoted into the dialog's parent. This is easier to explain in pictures than in text. Given the following window hierarchy:



The result of the `WS_EX_CONTROLPARENT` extended style being set is that the children of B are treated as if they were direct children of the main dialog, and the window B itself logically vanishes.



A B1 B2 C D

The `WS_EX_CONTROLPARENT` extended style means “Hello, I am not really a dialog control. I am a control parent. In other words, I have children, and *those children* are controls.” (Some people erroneously put the `WS_EX_CONTROLPARENT` extended style on the main dialog itself. That’s wrong, but fortunately it also has no effect.) Okay, this is all stuff you already know. So why am I bringing up this topic? I sort of gave it away in the subject line: When you use `WS_EX_CONTROLPARENT`, you need to be careful that the controls that you are promoting into the parent don’t conflict with controls already in the parent, or with controls promoted into the parent by another sibling. Suppose, in the above scenario, that window C also had the `WS_EX_CONTROLPARENT` extended style, and it had children C1 and C2. Not only do you have to watch out that B1 and B2 don’t conflict with the controls A or D, you also have to watch out that it doesn’t conflict with C1 or C2 either. “But Mister Wizard, the property sheet control hosts multiple child dialogs, and since they can be provided by third parties, it’s entirely possible (and likely) that there will be conflicts between B1 and, say, C2. Why doesn’t this create a problem?”

Well, Timmy, most of the time, it doesn’t, because notifications are fired to the control’s parent, and in the case of child dialogs, the child dialog’s child controls fire their notifications to the child dialog. So as long as the identifiers are unique within the child dialog, you won’t have a problem. This isn’t the entire answer, however, but to understand it further, we’ll need to look at another consequence of control ID conflicts, which we’ll take up next time.

Raymond Chen

Follow

