

What is the DS_CONTROL style for?

 devblogs.microsoft.com/oldnewthing/20040730-00

July 30, 2004



Raymond Chen

The DS_CONTROL dialog box style indicates that the dialog being created is being used as the child of another dialog rather than being a top-level dialog in and of itself. Nested dialogs are hardly a scary sight. You see them in property sheets, for example. Each page on a property sheet is a separate dialog; they all live inside the outer dialog, the property sheet itself. Nested dialogs are sometimes used in the common file dialogs: You can see one in action if you do a Save As from Notepad. The extra options at the bottom come from a nested dialog. When you set the DS_CONTROL style on a dialog template (or set the WS_EX_CONTROLPARENT extended style on a regular window), a bunch of new rules kick in. First of all, the WS_CAPTION and WS_SYSMENU styles in your dialog template are ignored. Because you're a child window now, not a top-level window, so you don't get a caption bar or a system menu. (The caption bar and system menu come from the outer window.) Next, the dialog navigation functions like [GetNextDlgTabItem](#) will recurse into windows marked WS_EX_CONTROLPARENT when they inspect the controls on a dialog box (in [GetNextDlgTabItem](#)'s case, because it is looking for a control to tab to). Without the extended style, the control search treats the embedded dialog box as one giant control rather than as a container for other controls. When you create a dialog with the DS_CONTROL style, you invariably use one of the creation functions like [CreateDialogParam](#) rather than one of the dialog box functions like [DialogBoxParam](#), because the modal loop is controlled by the outer dialog, not the inner one. The recursive behavior is important to know in order to avoid sending the dialog manager into an infinite loop. When you ask [GetNextDlgTabItem](#) to look for the previous item, what it does is take the starting control, then walk through the controls on the dialog until it comes back to the starting point, at which point it returns the one it saw before that one. If you forget to mark your dialog as DS_CONTROL, and focus started out in the sub-dialog, then the control enumeration will not recurse into the sub-dialog and consequently the starting point will never be found. The dialog manager will just keep looping, hunting for that starting-point control and never finding it.

(This problem exists even without DS_CONTROL. If you start out on a disabled or invisible control, then the walk through the controls will again never find the starting point, since disabled and invisible controls are skipped over when tabbing through a dialog.)

[Raymond Chen](#)

Follow

