More notes on use of the DS_CONTROL style

devblogs.microsoft.com/oldnewthing/20231101-00

November 1, 2023

Raymond Chen

Here’s a follow-up on some old articles on the DS_CONTROL dialog style and avoiding accidental control ID collisions.

When you add the DS_CONTROL style to your dialog template, there are other changes to the dialog styles which come into play. Some are mandatory, some are automatic, and some are optional.

Mandatory: You *must* remove the WS_POPUP style and you *must* add the WS_CHILD style. If you omit the WS_CHILD style or accidentally specify the WS_POPUP style, then the dialog won’t be created as a child window of the parent window you specified.

Automatic: You *may* remove the WS_CAPTION and WS_SYSMENU styles. The dialog manager removes them automatically, so this isn’t a requirement, but you are welcome to remove them yourself, either to avoid confusion or out of a sense of cleanliness.

Optional: You have the option of adding or removing the WS_VISIBLE and DS_MODALFRAME styles, depending on the behavior you desire.

If you specify WS_VISIBLE, then the child dialog will be created visible. Otherwise, it will be created hidden.

If you specify DS_MODALFRAME, then the child dialog gets the WS_EX_WINDOWEDGE extended style, which gives it an extra outer edge border. If you want the child dialog to blend in seamlessly with its parent, then you probably want to remove that style.