Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Sat, 30 Oct 2010 11:45:30 GMT
View Forum Message <> Reply to Message

There was found, I think, a good way to get away of this "issue" (the temporary addition to an extended style of windows a flag WS_EX_KOMPOSITED *) - but I want to find the reason of it.
Do not judge strictly an amateur in programming.
After a series of experiments with MS Spy++ utility I made some conclusions, this "issue" has two causes:
1. Child windows of traditional Windows applications have a flag of style CS_SAVEBITS. Child windows of UPP apps, this style does not have (or have, but not all).
After the forced set of this flag to all windows, "issue" only occurs if UPP application is busy, for example when I opening a package in TheIDE or doing "Rescan Code".
2. If UPP app does not busy (idle) and it creates the child window, the parent window is not repainting. But if UPP app is busy and it creates the child window, in this case, for some reason is starting repainting of the parent window or region of it, and CS_SAVEBITS is no longer helps.
If in this case the parent window or region of it does not repaint and all the child windows would have a flag CS_SAVEBITS (like in traditional applications), then the "issue" would not appear (IMHO).
This was only my best guess, do not take very seriously.

* With WS_EX_COMPOSITED set, all descendants of a window get bottom-to-top painting order using double-buffering. Bottom-to-top painting order allows a descendent window to have translucency (alpha) and transparency (color-key) effects, but only if the descendent window also has the WS_EX_TRANSPARENT bit set. Double-buffering allows the window and its descendents to be painted without flicker.