
Subject: Re: Ctrl Transparency, backpaint, ...
Posted by [mirek](#) on Fri, 26 Mar 2021 10:33:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

Didier wrote on Fri, 26 March 2021 10:41Great !!!
i will look into this tonight :)
Thanks

What is the general strategy to deal with these matters in widgets ?
There are some methods existting in Ctrl:
* Transparent() ==> set a bool that can be used in other places

Transparent was the flag that was originally intended as a way for widget to say to CtrlCore painting code that it should be treated as transparent. Over time, it evolved to be the flag for the widget as well - widget sets its default state in constructor and user code can change that. Widget however needs to honor the flag in its painting code (most of them do now, TabCtrl did not, that is what lead to the problem).

Note that from the CtrlCore perspective Transparent flag is mainly the optimization - it reduces the number of widgets that have to be repainted.

Quote:

* Backpaint() : not sure what this is intended for (although I expected using TransparentBackpaint() would have fixed my problem ... but it didn't)

You can safely forget about these. These were in the past methods to optimize things, today everything is set by default to backpaint.

Mirek
