
Subject: Re: BackPaint question

Posted by [mirek](#) on Sun, 22 Jan 2006 20:18:58 GMT

[View Forum Message](#) <> [Reply to Message](#)

FULLBACKPAINT just paints it in the "backpaint" buffer and transfers the result to the screen. Means, if some areas in the window are to be repainted (U++ always cumulates are "damaged" areas and defers it as much as possible, in Win32 that is done automatically and repainting is done when WM_PAINT for top-level Ctrl is recieved, in X11 a lot of additional code is involved).

In other words, without FULLBACKPAINT all Paint routines draw directly to the screen, which can result in flickering.

With FULLBACKPAINT, help buffer is used, Paint paints to it and then it is transfered to the screen.

Alternative mode backpaints areas covered by transparentCtrls only, as those are places where flickering is most visible. This mode is default (and it is also the only reason for "Ctrl::Transparent" flag to exist).

(Remaining option, EXCLUDEPAINT, is just "misuse" that reuses the flag to support special cases, like OLE controls that have WM_PAINT based painting).

Sorry for confusion. Please feel free to improve on docs if you are going to fix grammar there.
