

---

Subject: Re: BackPaint question

Posted by [hojtsy](#) on Fri, 20 Jan 2006 23:27:48 GMT

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

---

Invoking `BackPaint()` without parameters is equivalent to `BackPaint(FULLBACKPAINT)`. Any and all documentation about this enum value is "Whole area of `Ctrl` is backpainted". This short description can be understood in a whole lot of ways. My understanding was that it instructs the `Ctrl` class to paint the full area of the widget to the background color before calling the overloaded `Paint` method. I tried to search in the up sources to find where and how this `FULLBACKPAINT` is used. I found that it is used in the `Ctrl::CtrlPaint` method. I tried to decrypt how that method works, and it seems that it uses the undocumented `BackDraw` class to buffer the drawing operations done in `Ctrl::Paint`. Now I see that this can avoid flickering, but still don't understand what is that connection with "Whole area of `Ctrl` is backpainted"?! It is quite possible to do multiple drawing operations (like overlapping images or polygons) in a sequence which could result in flickering even if you do not backpaint the whole `Ctrl`. In such case you would also activate this mode, and not because of backpainting. So is it possible that both the name and the documentation of this mode is misleading? Or am I misunderstanding something?

---