

---

Subject: BackPaintHint

Posted by [mirek](#) on Mon, 05 Nov 2007 13:57:07 GMT

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

---

This is a special hint that says U++ that single next repaint of window which the widget belongs to should be done in backpaint mode to avoid flickering.

It is now used when splitter moves or window is resized.

(Also, most likely, it will be avoided in low-end machine - as soon as I will find heuristics that actually is able to tell me which machine is low-end...)

Mirek

---

---

Subject: Re: BackPaintHint

Posted by [mirek](#) on Tue, 13 Nov 2007 09:17:31 GMT

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

---

Ops, had to remove this feature.

To my surprise, I have found that Win32 produces different results when rendering text to off-screen bitmap DC and to the screen, at least with some videocards as you can check with this code:

```
#include <CtrlLib/CtrlLib.h>

using namespace Upp;

class Vera : public TopWindow {
public:
    virtual void Paint(Draw& draw);

    typedef Vera CLASSNAME;
};

void DrawIt(Draw& w)
{
    w.DrawRect(0, 0, 10000, 10000, White());
    w.DrawText(10, 10, "Vera::Paint", Font().FaceName("Bitstream Vera Sans Mono").Height(13));
    w.DrawText(10, 30, "Arial", Arial(18));
    w.DrawText(10, 50, "Roman", Roman(18));
}

void Vera::Paint(Draw& w)
{
    DrawIt(w);
}
```

```
ImageDraw iw(200, 200);  
DrawIt(iw);  
w.DrawImage(100, 0, iw);  
}
```

```
GUI_APP_MAIN  
{  
  Vera().Run();  
}
```

It is best visible with Bitstream Vera Sans Mono font, but with Arial or Roman too...

Means, we cannot mix backpainting with direct painting in a single widget as it looks weird.

Mirek

---