
Subject: Ctrl Transparency, backpaint, ...

Posted by [Didier](#) on Wed, 24 Mar 2021 22:49:18 GMT

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

Hello,

I am struggling with some painting issues for some time now.

All I want to do is to have some transparent widgets or widgets displaying a background image with aplha in order to see the Topwindow background

Everything is drawn well on start and when resizing but when I hover over a widget THAT IS DISPLAYED over an alpha image, then some glitches appear.

The same happens when I refresh the layout

It looks like the Topwindow background is not redrawn as it should be.

I uploaded a test case:

- * just hover over the option in the tab ==> option glitch appears
- * Change selected row inside Grid : call Refresh() of the tab ==> whole background is glitch !
- * Once glitch is present, just change tab or resize window and everything comes back to normal

How can I correct this ?

[Edit] removed 7z file

Subject: Re: Ctrl Transparency, backpaint, ...

Posted by [Didier](#) on Wed, 24 Mar 2021 22:54:18 GMT

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

Here are three images of the test case

- * left : is what it should look like
- * middle : glitch on option widget
- * right : glitch on whole tab background

File Attachments

1) [TestTransparency.png](#), downloaded 535 times

Subject: Re: Ctrl Transparency, backpaint, ...

Posted by [mirek](#) on Thu, 25 Mar 2021 10:31:50 GMT

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

There is missing TestTransparency.upp in the package? why?

Subject: Re: Ctrl Transparency, backpaint, ...

Posted by [Didier](#) on Thu, 25 Mar 2021 10:52:07 GMT

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

Oups, made the zip using the export code function

I will send .upp file

Subject: Re: Ctrl Transparency, backpaint, ...

Posted by [Didier](#) on Thu, 25 Mar 2021 18:27:32 GMT

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

Here is the correct package

I tried everything I found : backpaint,transparent, Parent->Refresh(), ...

I am missing something :cry:

File Attachments

1) [TestTransparency.7z](#), downloaded 204 times

Subject: Re: Ctrl Transparency, backpaint, ...

Posted by [mirek](#) on Thu, 25 Mar 2021 19:52:46 GMT

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

Should be now fixed. There was a problem in TabCtrl.

Subject: Re: Ctrl Transparency, backpaint, ...

Posted by [Didier](#) on Fri, 26 Mar 2021 09:41:05 GMT

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

Great !!!

i will look into this tonight :)

Thanks

What is the general strategy to deal with these matters in widgets ?

There are some methods existing in Ctrl:

- * Transparent() ==> set a bool that can be used in other places
- * Backpaint() : not sure what this is intended for (although I expected using TransparentBackpaint() would have fixed my problem ... but it didn't)

So I am lost on how/when/why use these methods

I would like my widgets to be fully compatible with Upp, so I have to manage transparency correctly (and maybe I could then propose some corrections : Ex I found out that frame doesn't

manage transparency at all while Splitter does

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

Subject: Re: Ctrl Transparency, backpaint, ...

Posted by [mirek](#) on Fri, 26 Mar 2021 11:02:39 GMT

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

BTW, while working on it, I did not like how these WithXXXLayout<ParentCtrl> (interesting usage BTW) looked in the layout designer, so I spent a couple of hours fixing that. I would appreciate if you looked at it....

Mirek

Subject: Re: Ctrl Transparency, backpaint, ...
Posted by [Didier](#) on Fri, 26 Mar 2021 13:11:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Mirek,

1 - Ide doesn't compile du to some pen related code (I looks like accidentally pasted code) i
GtkEvent.cpp L428 so I commented it out and it compiled and worked after

```
void Ctrl::Proc()
{
#ifndef LOG_EVENTS
String ev = "?";
Tuple2<int, const char *> *f = FindTuple(xEvent, __countof(xEvent), CurrentEvent.type);
if(f)
    ev = f->b;
LOG(> PROCESS EVENT " << Upp::Name(this) << " " << ev);
ProcStop tm;
tm.ev = ev;
#endif
if(!isOpen())
    return;
Ptr<Ctrl> _this = this;
bool pressed = false;
int kv, hw;
static int clicktime = msecs() - 100000;
/* -----
pen = CurrentEvent.pen;
pen_barrel = CurrentEvent.pen_barrel;
pen_inverted = CurrentEvent.pen_inverted;
pen_eraser = CurrentEvent.pen_eraser;
pen_pressure = CurrentEvent.pen_pressure;
pen_rotation = CurrentEvent.pen_rotation;
pen_tilt = CurrentEvent.pen_tilt;
-----
*/
switch(CurrentEvent.type) {
case GDK_MOTION_NOTIFY:
    GtkMouseEvent(MOUSEMOVE, MOUSEMOVE, 0);
    break;
}
```

For the rest, transparency works fine now and layout designer also works fine now
Thanks :thumbup:

Quote:interesting usage BTW)
I don't like to code my GUIs so I use layout everytime I can
