

---

Subject: Re: U++ 2019.1.rc4 released  
Posted by [cbpporter](#) on Wed, 24 Apr 2019 10:04:32 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Edit:

After further testing, I think solution 2 is better.

Solution 1 was present in the attachment, so I am attaching solution 2 in a new post.

Issue #2: High contrast

I have spent a great deal of time looking over Win32 chameleon and now have a good idea on what the OS gives us when it comes to skinning. You weren't kidding when you said that dark mode was "emulation".

With so little information, inverting was pretty clever. I set up more nuanced processing for my skin, but it only works for a single windows skin. So your solution is more universal and probably the best we can do.

The problem is that it results in a high contrast themes.

The problem is that light skins is not black text on white background, only for some controls. There are gray backgrounds, shadows and overall a nice array of colors. On the other hand, if you look at our color palette colors, in dark mode, they are purely white on black background. A "normal" dark skin will be many shades of gray and some gray tinted colors. Pure white and black are taboo.

As I said above, this is not fixable with current support, but we can add some customization.

Let's start small: BlackBorder (and WhiteBorder).

The issues is that BlackBorder is SColorText, so black normally. And white in dark mode. Fair enough. But BlackBorder is used by BlackFrame in CtrlLib too as a border, and there it is similar but different from ViewBorder.

In some skins it might make sense for it to be black/white, but for a modern look, it would be probably SColorText/SColorShadow in light/dark. But this would ruin Draw.

Solution 1:

Leave things as they are, but allow us change BlackBorder color. This way I can skin this in my code.

This solution is a bit awkward since we already have:  
`const ColorF *BlackBorder();`

Maybe there is a better way?

I have attached this solution.

Solution 2:

Make BlackBorder truly black (might or might not be useful to customize) and change BlackFrame to not use black border, but a customizable Chameleon value.

So BlackFrame would have 4 colors to customize and maybe some extra goodies, like border width customization?

And BlackBorder would be used by Draw to draw black borders and never by GUI, while the GUI would use BlackFrame, a far more flexible solution? With image and hotspot support?

## File Attachments

---

1) [patch2.zip](#), downloaded 304 times

---