

---

Subject: Re: U++ 2019.1.rc4 released  
Posted by [mirek](#) on Thu, 18 Apr 2019 12:54:08 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

cbpporter wrote on Thu, 18 April 2019 14:29OK, let's go in order.

Before the dark mode support, setting dark mode in Windows didn't work. So clearly it is not a "simple" issue of applying a system theme, because most applications ignore that windows option.

Well, the real secret is that Windows does not really support dark theme for Win32 apps....

So what we do is to detect the fact that win10 is in dark theme, then roll out our own based on DarkTheme versions of system colors.

Quote:

So my questions are:

- how do you activate dark theme in any app once, and have it work forever? Even if Windows is in light mode. Same for light mode.

Do not call DarkThemeEnabled. It does something else.

Create dark theme skin. That must be a function that sets everything to your desired colors.

Use SetSkin.

Quote:

Case in point:

```
void Ctrl::SetSkin(void (*_skin)())
```

It is not clear what it does and there are 100 more ways to touch skins.

There are 100 ways to touch the skin, BUT you are supposed to do them in skin routine.

So basically it does what you would want from SetTheme(Theme \*). But the advantage is that you inherit some default skin which you are changing and you can even call "subskin" routines there.

I agree that more docs and some cleanup would be nice, but the basic idea is IMO sound.

Mirek

---