Subject: Re: DarkTheme function parameters changed
Posted by Lance on Wed, 09 Oct 2024 21:39:20 GMT
View Forum Message <> Reply to Message

Maybe we can do dynamic color within the current Color. I don't know the full picture, but here are some thoughts:

1. upto 254 System colors are store in a c array that is indexable with a integer i (0<=i<=254); These are, of course, SColorPaper, SColorInk, SBlack, SWhite, etc.

2. End user(programmer) are deprived of alpha value of 255. With alpha=255
  a. if r=g=b=255, that's the Null Color;
  b. If g=b=255, r<255, r is the color index of a system color;
  c. if b<=254, the Final Color = Blend(ColorFromIndex(r), ColorFromIndex(g), b);

3. If a programmer's Color choice conflicts with dynamic color ones, a warning will be logged but a color close enough will be silently supplied instead.

4. Color are finalized on the site. So AttrTxt, etc, only calculate the final Color before it actually renders text.

5. UPP supports light theme(normal theme), dark theme out of box, free of charge. If a programmer wants his/her program sticks to light theme, sticks to dark theme, stick to host theme (between dark and non-dark, any non-dark host theme will be interpreted to light theme by upp, before further themes are developed, if ever). Remove the chamalion Style * style;(a further 8 byte reduction on x64 platforms) from Ctrl definition, but add a GetChStyle()const virtual function, which will return light theme, dark theme, or host-dependant theme according to programmers' choice. If anyone want to Style a Ctrl, he/she should override GetChStyel() to return pointers to his/her own full set of theme values. Don't use, don't pay.

5. Icon will remain to be an issue.