Subject: Re: DarkTheme function parameters changed
Posted by Tom1 on Tue, 17 Sep 2024 18:31:45 GMT
View Forum Message <> Reply to Message

mirek wrote on Tue, 17 September 2024 18:00Tom1 wrote on Sat, 14 September 2024 22:32Now that DarkTheme() 'color inversion' works so great, maybe a slight tuning of default TheIDE text colors (along the lines laid out in material.io Dark theme documentation) is in place. I'm thinking about:

Default scheme should always be "host". However, I can add alternative dark theme, if you supply one. It is quite easy, check

CtrlLib/Ch.cpp:716

Mirek
I was actually just thinking about TheIDE accent colors for code editing and other views... just to adjust the dark variants to view nicely on dark backgrounds. But yes, this involves dark theme tuning as well to get contrast optimized.

I remember working on Win32 emulated dark theme colors. The trick was (and I think still is) that many definitions come from platform and mostly only colors need to be adjusted for the dark theme. I recall that the color changes had to be written in-line within ChHostSkin(). If I just create a function called e.g. ChMyCustomSkin(), similar to e.g. ChDarkSkin(), I will not get platform shapes/styles for widgets, but something else instead. Therefore, some reorganization would be helpful:

It would be nice to have theme loading clearly split to two parts:

1. Load system colors (custom or platform) and store with "SColor*_Write();" functions like before.
2. Load shapes/styles from host like in ChHostSkin() (or load custom shapes/styles), and combine with preloaded SColor*() colors for a complete theme.

Also live theme changes would be nice... We have talked about this before and at that time you warned about a potential problem that arises with some controls caching colors internally. Still, it would be an interesting improvement. (I can even imagine a generic Theme(Color)EditorDialog after live theme changes feature has been introduced...)

Best regards,

Tom