
Subject: Re: New dark theme thread

Posted by [Tom1](#) on Wed, 05 Jan 2022 12:18:25 GMT

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

mirek wrote on Wed, 05 January 2022 10:13OK, removed those unsupported colors.

There is one more complication: At least in Win10, Win32 actually does not return dark colors in dark mode, so we are basically "flipping" normal colors to dark (we detect this situation where the theme is dark, but COLOR_WINDOW is light, see CtrlLib/ChWin32.cpp:377). How does it behave in Win11? Do they return dark colors now?

As to actual scheme, I am ready to accept any suggestions. Not sure what to do with all this mess...

Mirek

Hi Mirek,

The Windows 10 complication remains in W11... The colors do not change when the theme is changed between Light and Dark.

(Actually, previously I thought I was looking at windows system colors through SColor*(), but indeed they were something already fixed in U++.)

Let's start with the current situation. Here's my test code:

```
RDUMP(Color::FromCR(GetSysColor(COLOR_WINDOW)));
RDUMP(Color::FromCR(GetSysColor(COLOR_WINDOWTEXT)));
RDUMP(Color::FromCR(GetSysColor(COLOR_HOTLIGHT)));
RDUMP(Color::FromCR(GetSysColor(COLOR_GRAYTEXT)));
RDUMP(Color::FromCR(GetSysColor(COLOR_HIGHLIGHT)));
RDUMP(Color::FromCR(GetSysColor(COLOR_HIGHLIGHTTEXT)));
RDUMP(Color::FromCR(GetSysColor(COLOR_3DFACE)));
RDUMP(Color::FromCR(GetSysColor(COLOR_BTNTEXT)));
```

```
RDUMP(SColorDisabled());
RDUMP(SColorDkShadow());
RDUMP(SColorFace());
RDUMP(SColorHighlight());
RDUMP(SColorHighlightText());
RDUMP(SColorInfo());
RDUMP(SColorInfoText());
RDUMP(SColorLabel());
RDUMP(SColorLight());
RDUMP(SColorLtFace());
RDUMP(SColorMark());
RDUMP(SColorMenu());
RDUMP(SColorMenuMark());
RDUMP(SColorMenuText());
```

```
RDUMP(SColorPaper());  
RDUMP(SColorShadow());  
RDUMP(SColorText());
```

Here are the results for Windows 11 in Light mode:

```
Color::FromCR(GetSysColor(COLOR_WINDOW)) = Color(255, 255, 255)  
Color::FromCR(GetSysColor(COLOR_WINDOWTEXT)) = Color(0, 0, 0)  
Color::FromCR(GetSysColor(COLOR_HOTLIGHT)) = Color(0, 102, 204)  
Color::FromCR(GetSysColor(COLOR_GRAYTEXT)) = Color(109, 109, 109)  
Color::FromCR(GetSysColor(COLOR_HIGHLIGHT)) = Color(0, 120, 215)  
Color::FromCR(GetSysColor(COLOR_HIGHLIGHTTEXT)) = Color(255, 255, 255)  
Color::FromCR(GetSysColor(COLOR_3DFACE)) = Color(240, 240, 240)  
Color::FromCR(GetSysColor(COLOR_BTNTEXT)) = Color(0, 0, 0)  
SColorDisabled() = Color(109, 109, 109)  
SColorDkShadow() = Color(63, 63, 63)  
SColorFace() = Color(240, 240, 240)  
SColorHighlight() = Color(0, 120, 215)  
SColorHighlightText() = Color(255, 255, 255)  
SColorInfo() = Color(255, 255, 192)  
SColorInfoText() = Color(0, 0, 0)  
SColorLabel() = Color(0, 0, 0)  
SColorLight() = Color(255, 255, 255)  
SColorLtFace() = Color(247, 247, 247)  
SColorMark() = Color(0, 0, 255)  
SColorMenu() = Color(240, 240, 240)  
SColorMenuMark() = Color(0, 0, 255)  
SColorMenuText() = Color(0, 0, 0)  
SColorPaper() = Color(255, 255, 255)  
SColorShadow() = Color(128, 128, 128)  
SColorText() = Color(0, 0, 0)
```

Here are the results for Windows 11 in Dark mode:

```
Color::FromCR(GetSysColor(COLOR_WINDOW)) = Color(255, 255, 255)  
Color::FromCR(GetSysColor(COLOR_WINDOWTEXT)) = Color(0, 0, 0)  
Color::FromCR(GetSysColor(COLOR_HOTLIGHT)) = Color(0, 102, 204)  
Color::FromCR(GetSysColor(COLOR_GRAYTEXT)) = Color(109, 109, 109)  
Color::FromCR(GetSysColor(COLOR_HIGHLIGHT)) = Color(0, 120, 215)  
Color::FromCR(GetSysColor(COLOR_HIGHLIGHTTEXT)) = Color(255, 255, 255)  
Color::FromCR(GetSysColor(COLOR_3DFACE)) = Color(240, 240, 240)  
Color::FromCR(GetSysColor(COLOR_BTNTEXT)) = Color(0, 0, 0)  
SColorDisabled() = Color(184, 184, 184)  
SColorDkShadow() = Color(213, 213, 213)  
SColorFace() = Color(24, 24, 24)  
SColorHighlight() = Color(97, 217, 255)  
SColorHighlightText() = Color(1, 1, 1)  
SColorInfo() = Color(26, 26, 0)  
SColorInfoText() = Color(255, 255, 255)  
SColorLabel() = Color(255, 255, 255)
```

```

SColorLight() = Color(1, 1, 1)
SColorLtFace() = Color(12, 12, 12)
SColorMark() = Color(214, 214, 255)
SColorMenu() = Color(24, 24, 24)
SColorMenuMark() = Color(214, 214, 255)
SColorMenuText() = Color(255, 255, 255)
SColorPaper() = Color(1, 1, 1)
SColorShadow() = Color(172, 172, 172)
SColorText() = Color(255, 255, 255)

```

So, it is easy to see that Windows no longer takes any active part in color selection in Dark theme. It is just the Light/Dark -theme selection and then one accent color selection somewhere in Windows settings. The rest is up to the application.

However, the game changes when a 'Accessibility > Contrast theme', e.g. 'Night sky' in Windows 11 is selected:

```

Color::FromCR(GetSysColor(COLOR_WINDOW)) = Color(0, 0, 0)
Color::FromCR(GetSysColor(COLOR_WINDOWTEXT)) = Color(255, 255, 255)
Color::FromCR(GetSysColor(COLOR_HOTLIGHT)) = Color(128, 128, 255)
Color::FromCR(GetSysColor(COLOR_GRAYTEXT)) = Color(166, 166, 166)
Color::FromCR(GetSysColor(COLOR_HIGHLIGHT)) = Color(214, 180, 253)
Color::FromCR(GetSysColor(COLOR_HIGHLIGHTTEXT)) = Color(43, 43, 43)
Color::FromCR(GetSysColor(COLOR_3DFACE)) = Color(0, 0, 0)
Color::FromCR(GetSysColor(COLOR_BTNTEXT)) = Color(255, 238, 50)
SColorDisabled() = Color(166, 166, 166)
SColorDkShadow() = Color(213, 213, 213)
SColorFace() = Color(0, 0, 0)
SColorHighlight() = Color(214, 180, 253)
SColorHighlightText() = Color(43, 43, 43)
SColorInfo() = Color(26, 26, 0)
SColorInfoText() = Color(255, 255, 255)
SColorLabel() = Color(255, 255, 255)
SColorLight() = Color(1, 1, 1)
SColorLtFace() = Color(0, 0, 0)
SColorMark() = Color(214, 214, 255)
SColorMenu() = Color(0, 0, 0)
SColorMenuMark() = Color(214, 214, 255)
SColorMenuText() = Color(255, 255, 255)
SColorPaper() = Color(0, 0, 0)
SColorShadow() = Color(172, 172, 172)
SColorText() = Color(255, 255, 255)

```

Finally I have attached two images that portray Windows 11 light and dark theme colors. I guess these colors should be used for light and dark themes, and GetSysColor() -system colors for 'Contrast themes'. Windows apps also seem to handle some darkening and lightening of default colors by a few (+/- 3) steps for some uses.

I guess we could start with emulating dark mode just like it is now, but picking the colors from Windows apps directly. It is probably worth ensuring that the 'Accessibility > Contrast themes' are kept working through current eight system colors.

Probably only Windows 10 and 11 should detect the light/dark theme and adapt its colors accordingly. Older Windows should always use system colors. But then again, if 'Contrast theme' is selected in Windows 10/11, they should also use Windows system colors.

Best regards,

Tom

EDIT: Found the color picker on Linux Mint. Removed sample images as the colors did not match the ones on my desktop with the default dark theme on Windows 11. Now I will try to pick some colors from my desktop to see if they could be used.
