
Subject: Re: New dark theme thread

Posted by [mirek](#) on Thu, 06 Jan 2022 07:44:38 GMT

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

Tom1 wrote on Wed, 05 January 2022 16:15Mirek,

I thought I could apply custom colors through SColor*_Write(), so I tried this for UWord dropping the following code into the place where default SColors are initilized in ChWin32.cpp:

...

```
for(sysColor *s = sSysColor; s < sSysColor + __countof(sSysColor); s++) // this also resets all  
imls via SColorPaper_Write!!!
```

```
(*s->set)(sAdjust(Color::FromCR(GetSysColor(s->syscolor))));
```

```
SColorDisabled_Write(Color(0x95, 0x95, 0x95));  
SColorDkShadow_Write(Color(0x20, 0x20, 0x20));  
SColorFace_Write(Color(0x20, 0x20, 0x20));  
SColorHighlight_Write(Color(GetSystemAccentColor()));  
SColorHighlightText_Write(IsDark(SColorHighlight())?White():Black());  
SColorInfo_Write(Color(0x20, 0x20, 0x20));  
SColorInfoText_Write(White());  
SColorLabel_Write(Color(0x20, 0x20, 0x20)); // White();  
SColorLight_Write(Color(0x20, 0x20, 0x20));  
SColorLtFace_Write(Color(0x20, 0x20, 0x20));  
SColorMark_Write(Color::FromCR(GetSysColor(COLOR_HOTLIGHT)));  
SColorMenu_Write(Color(0x20, 0x20, 0x20));  
SColorMenuMark_Write(Color::FromCR(GetSysColor(COLOR_HOTLIGHT)));  
SColorMenuText_Write(White());  
SColorPaper_Write(Color(0x19, 0x19, 0x19));  
SColorShadow_Write(Color(0x20, 0x20, 0x20));  
SColorText_Write(White());
```

having first implemented this:

```
RGBA GetSystemAccentColor(){  
    RGBA color=Gray(); // Default
```

```
#ifdef WIN32
```

```
    HKEY SWKey;  
    HKEY MSKey;  
    HKEY WINKey;  
    HKEY DWMKey;
```

```
    LONG Status;
```

```
// "Computer\HKEY_CURRENT_USER\Software\Microsoft\Windows\DWM"
```

```
    Status =
```

```
    RegOpenKeyEx(HKEY_CURRENT_USER,TEXT("Software"),0,KEY_READ,&SWKey);  
    if(Status == ERROR_SUCCESS){
```

```

Status = RegOpenKeyEx(SWKey,TEXT("Microsoft"),0,KEY_READ,&MSKey);
if(Status == ERROR_SUCCESS){
Status = RegOpenKeyEx(MSKey,TEXT("Windows"),0,KEY_READ,&WINKey);
if(Status == ERROR_SUCCESS){
Status = RegOpenKeyEx(WINKey,TEXT("DWM"),0,KEY_READ,&DWMKey);
if(Status == ERROR_SUCCESS){
DWORD RegType=REG_DWORD;
DWORD BufferSize=4;
byte pStr[16];

Status=RegQueryValueEx(DWMKey,TEXT("AccentColor"),NULL,&RegType,(LPBYTE)pStr,&BufferSize);

    if(Status == ERROR_SUCCESS){
        color.a=pStr[3]; // This is ARGB
        color.b=pStr[2];
        color.g=pStr[1];
        color.r=pStr[0];
    }

    RegCloseKey(DWMKey);
}
RegCloseKey(WINKey);
}
RegCloseKey(MSKey);
}
RegCloseKey(SWKey);
}
if(Status == ERROR_SUCCESS) return color; // Success

#endif
// TODO solve other sources

return color; // Default
}

```

Some parts of the GUI follow these colors, but many don't. So, the results are mixed with partly unreadable GUI. While the suggested colors are definitely not optimized, could you try the above with both light and dark theme in Windows? I could use some help here in order to get onto the right track in tuning the default theme. UWord seems to be a good place to test as it has many different Ctrl's in e.g. Style manager.

Best regards,

Tom

I will, but this frankly seems wrong from outset - you are mixing fixed color values with those got from GetSysColor.

I think we need something really simple here, something like

```
if(sEmulateDarkTheme) {
    SColorPaper_Write(Color(...));
    SColorFace_Write(Color(...));
    SColorText_Write(Color(...));
    SColorHighlight_Write(Color(...));
    SColorHighlightText_Write(Color(...));
    SColorMenu_Write(Color(...));
}
else
    for(sysColor *s = sSysColor; s < sSysColor + __countof(sSysColor); s++) // this also resets all
        imls via SColorPaper_Write!!!
        (*s->set)(sAdjust(Color::FromCR(GetSysColor(s->syscolor))));
```

(do not bother about defining more colors than you know, those will adjust automatically)
