
Subject: Re: U++ 2019.1.rc4 released
Posted by [cbpporter](#) on Fri, 19 Apr 2019 12:55:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

I'm close to getting a workable system for themeing that does not need U++ changes (maybe something minor) but lives peacefully together.

What I want to do is have choice in what theme to use.

```
if(Ctrl::IsDarkThemeEnabled() && IsSystemThemeDark() &&  
!IsDark(Color::FromCR(GetSysColor(COLOR_WINDOW))))  
    sEmulateDarkTheme = true;
```

So basically if IsDarkThemeEnabled and the system is dark themed, we use dark theme.

Here is where I want more control. I want all valid combinations of system dark theme and app dark theme, so 4 combinations.

We need to adjust this condition a bit.

I propose we rename IsDarkThemeEnabled, in something like IsDarkModePermitted and add a new IsDarkThemeEnabled, geter and seter, where the user can choose if the skin is dark. SetDarkThemeEnabled would be called set as is today, with IsSystemThemeDark() && !IsDark(Color::FromCR(GetSysColor(COLOR_WINDOW))), but NOT in ChSysInit. Before, on system startup, so auto-detection is correct, but I can override it later.

Of course, this is just an option. Here is what I want:

```
// autodetect light or dark  
void SetHostSkin() {  
    Ctrl::SetDarkThemeEnabled();  
    Ctrl::SetSkin(ChHostSkin);  
}  
  
// always light  
void SetLightHostSkin() {  
    Ctrl::SetDarkThemeEnabled(false);  
    Ctrl::SetSkin(ChHostSkin);  
}  
  
// always dark  
void SetDarkHostSkin() {  
    Ctrl::SetDarkThemeEnabled(true);  
    Ctrl::ForceDarkSkin(true);  
    Ctrl::SetSkin(ChHostSkin);
```

```

}

// old, unchanged
void SetStdSkin() {
    Ctrl::SetDarkThemeEnabled(false);
    Ctrl::SetSkin(ChStdSkin);
}

// old, unchanged
void SetClassicSkin() {
    Ctrl::SetDarkThemeEnabled(false);
    Ctrl::SetSkin(ChClassicSkin);
}

// my fixed version
void SetNewDarkHostSkin() {
    Ctrl::SetDarkThemeEnabled(false);
    Ctrl::SetSkin(ChHostSkin);

    // do fixes

    Ctrl::ReSkin();
}

```

I managed to do this, except for forcing dark skin on light skin windows. Hence the Ctrl::ForceDarkSkin(true); or the proposed rename mentioned above.