Subject: New dark theme thread
Posted by cbpporter on Fri, 19 Apr 2019 10:50:21 GMT
View Forum Message <> Reply to Message

So I tested out 2019.1 and for the first time I used dark mode support.

Aesthetics are subjective, so some may like it, some may not. I don't. Nor do I think that probably most graphics designers will. Or people with eyes. Unless you are somebody who needs high contrast themes.

Beyond that, there is eyestrain. It is well known that high contrast colors cause eye strain. There are several design docs upon design docs out there giving good practices. You can use formulas to compute contrast ratios.

The new dark mode support is not really a dark mode, but a high contrast theme.

I personally get a terrible eye pain when using this theme.

I don't want to highjack the official 2019.1 thread, so let's have a discussion here.

I spent the better part of the day trying to reduce eye strain and bring colors under check. It will be a long process, many controls to test, lots of trial and error.

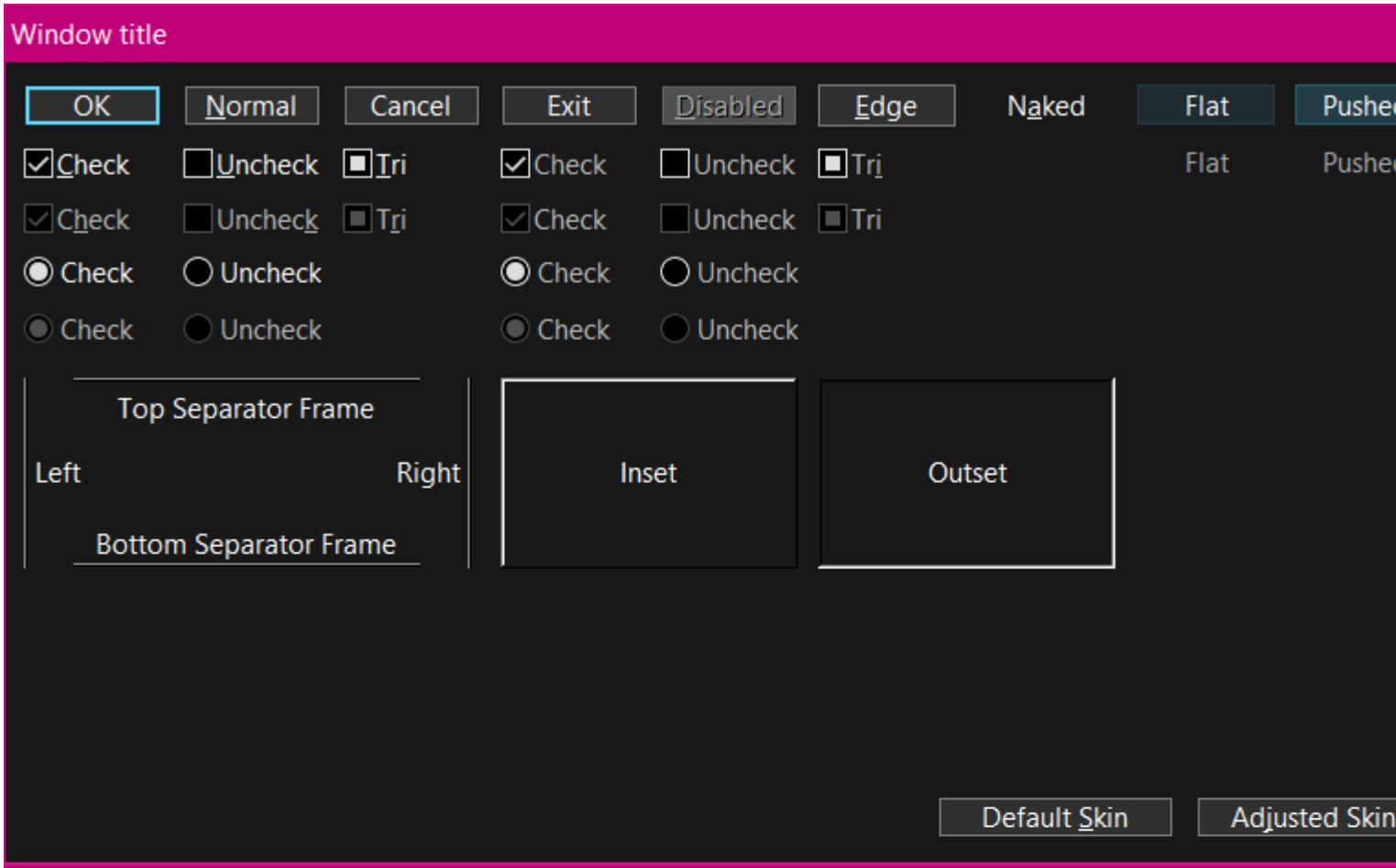
But here are the results of today's work, before above, after bellow:

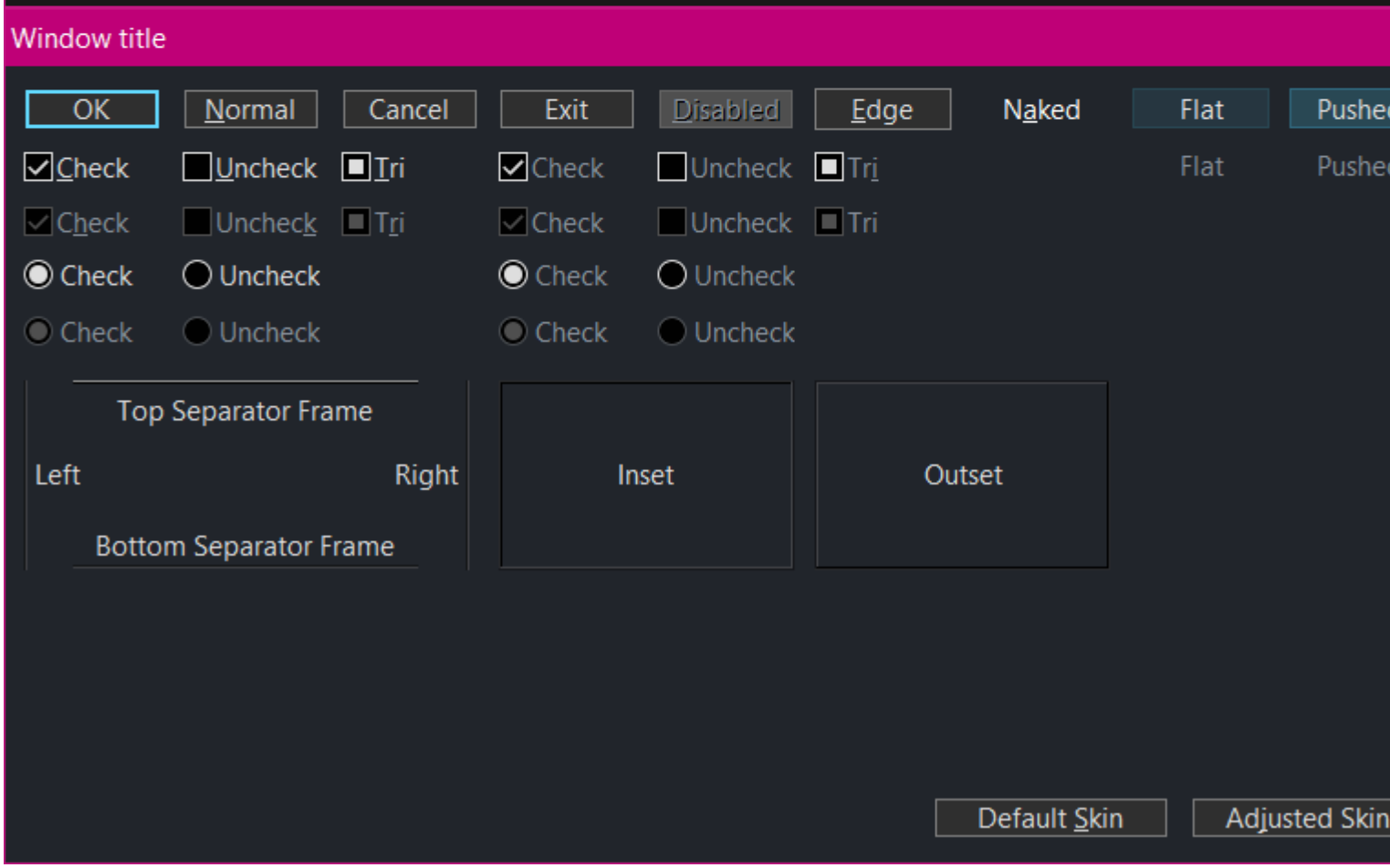
The toolbar is hot pink to see if this color affects any other GUI elements, in U++ or Win 10 apps. So far it doesn't.

PS: TheIDE needs to have an option to turn of dark mode, even if the system is in dark mode, until we fix this mode, or forever. It should default as false, i.e. use dark mode. Basically not call SetDarkThemeEnabled if dark mode is disabled.

File Attachments

1) comp1.png, downloaded 2348 times


Window title
OK
Normal
Cancel
Exit
Disabled
Edge
Naked
Flat
Check
Uncheck
Tri
Top Separator Frame
Left
Right
Bottom Separator Frame
Inset
Outset
Default Skin
Adjusted Skin


Window title
OK
Normal
Cancel
Exit
Disabled
Edge
Naked
Flat
Check
Uncheck
Tri
Top Separator Frame
Left
Right
Bottom Separator Frame
Inset
Outset
Default Skin
Adjusted Skin

Subject: Re: New dark theme thread
Posted by cbpporter on Mon, 22 Apr 2019 08:11:02 GMT
View Forum Message <> Reply to Message

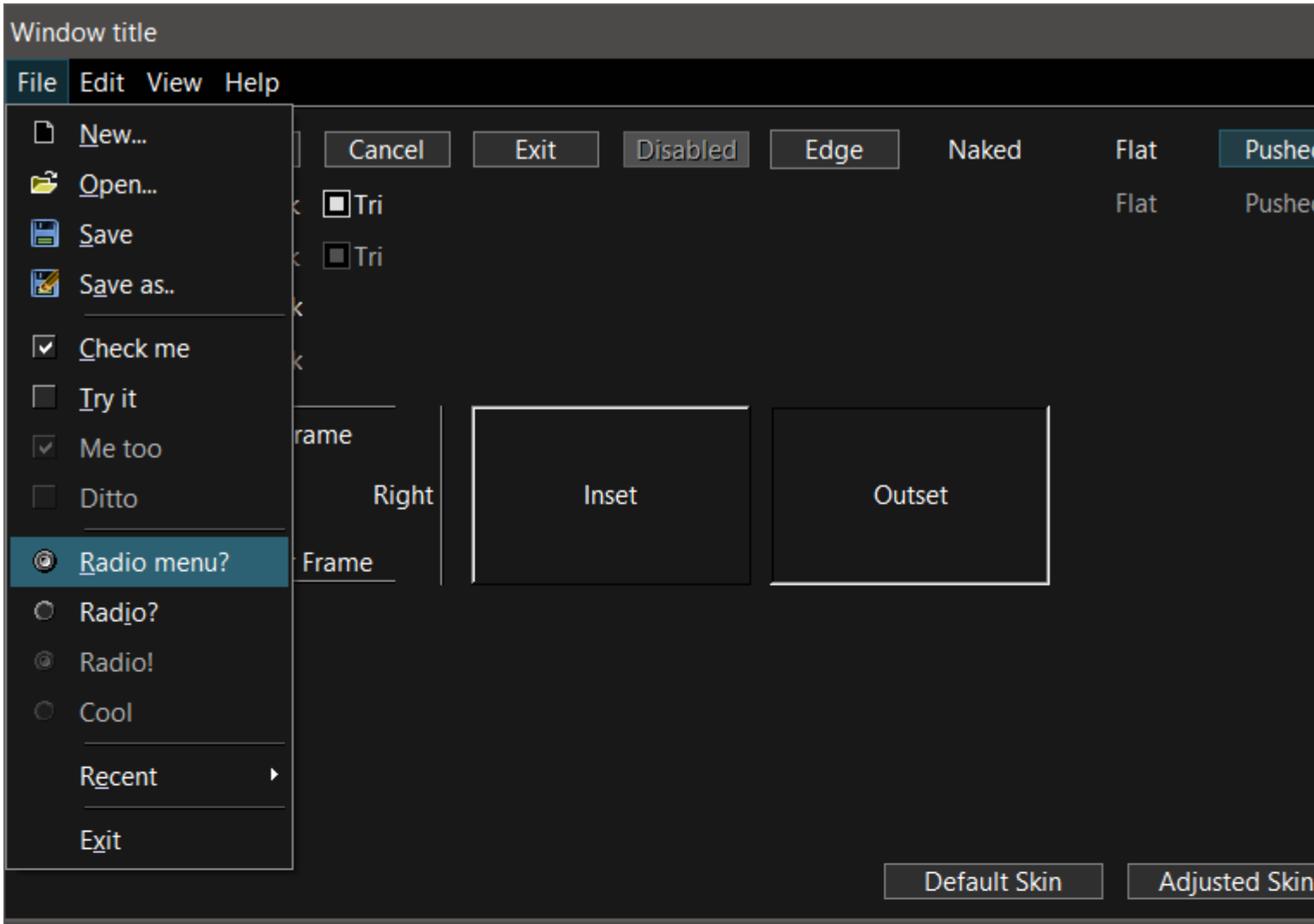
Another day of progress...

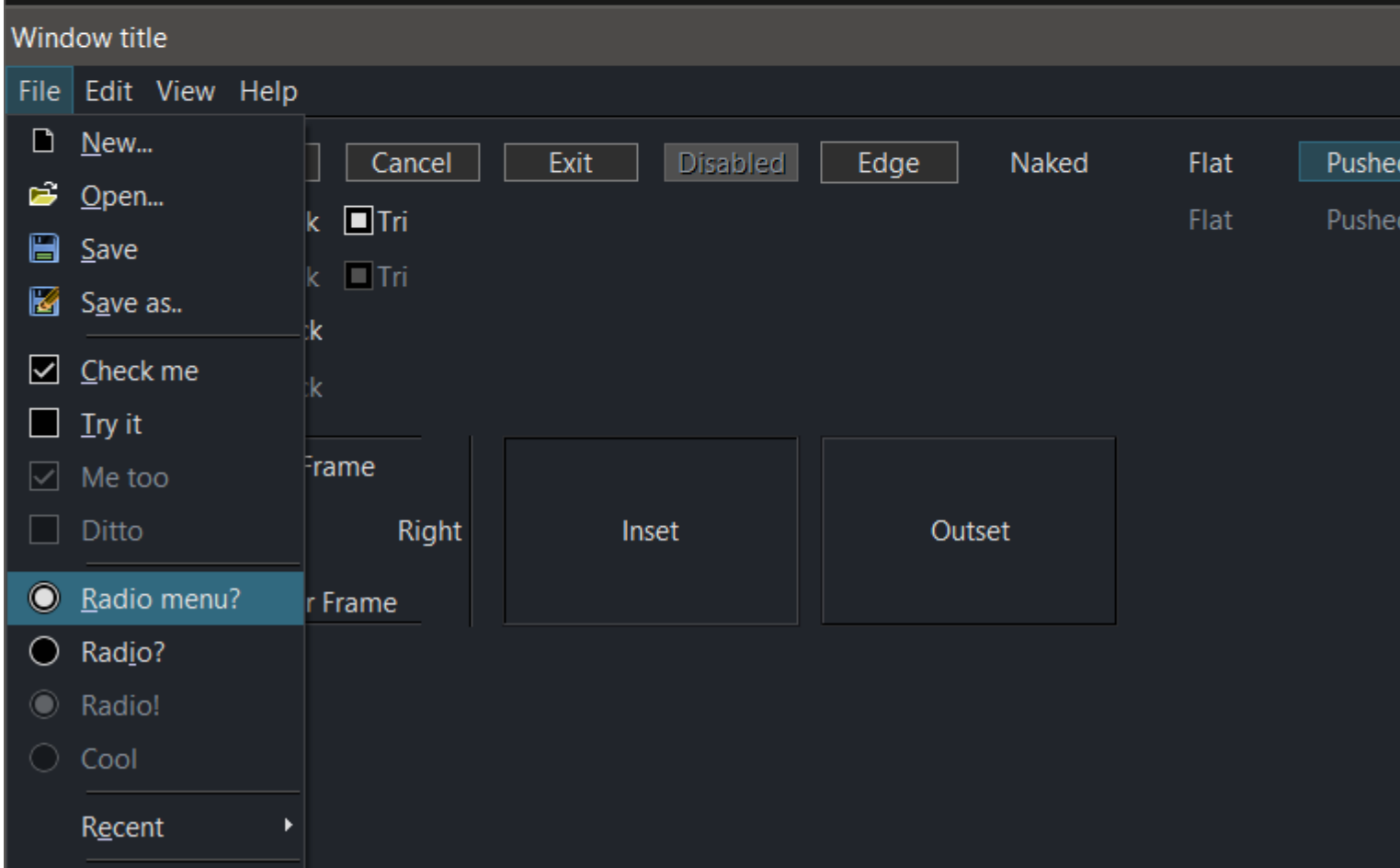
I'm having problems with adjusting the width of the separator in the menu and I'm not sure about the separator bezel as a whole and using native icons in menus, but the original radio menu option does not fit it with the overall look IMHO, so improvement?

What do you guys think?

File Attachments

1) comp2.png, downloaded 2300 times


Window title
File Edit View Help
New...
Open...
Save
Save as..
Check me
Try it
Me too
Ditto
Radio menu?
Radio?
Radio!
Cool
Recent
Exit
Cancel
Exit
Disabled
Edge
Naked
Flat
Tri
Tri
Right
Frame
Inset
Outset
Default Skin
Adjusted Skin


Window title
File Edit View Help
New...
Open...
Save
Save as..
Check me
Try it
Me too
Ditto
Radio menu?
Radio?
Radio!
Cool
Recent
Cancel
Exit
Disabled
Edge
Naked
Flat
Tri
Tri
Right
Frame
Inset
Outset

Subject: Re: New dark theme thread
Posted by cbpporter on Mon, 22 Apr 2019 08:32:08 GMT
View Forum Message <> Reply to Message

Oh, and incredibly weird bug.

If you compare the screenshot from the first post and the screenshot from the second, several option boxes are missing.

Adding the menu causes them to not be visible.

I'll try and produce a compilable minimal testcase.

Subject: Re: New dark theme thread
Posted by cbpporter on Mon, 22 Apr 2019 10:36:48 GMT
View Forum Message <> Reply to Message

I also did one of the most used controls in U++: clock!

A few observations:
1. I could not figure out how to control the color for those two bars on the left and right. Looks like some frame/backpaint problems.
2. This is not that important, but maybe we should update it with Painter for smooth AA lines?

File Attachments

1) comp3.png, downloaded 2311 times



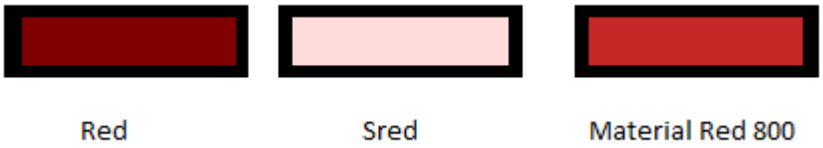
Subject: Re: New dark theme thread
Posted by cbpporter on Mon, 22 Apr 2019 10:39:42 GMT
View Forum Message <> Reply to Message

Continuing here because of 1 image/reply.

3. I used Material design color for red blue and green there. There are tables that allow you to make a color good for dark mode. We have SRed, but here is comparison:

File Attachments

1) red.png, downloaded 2216 times



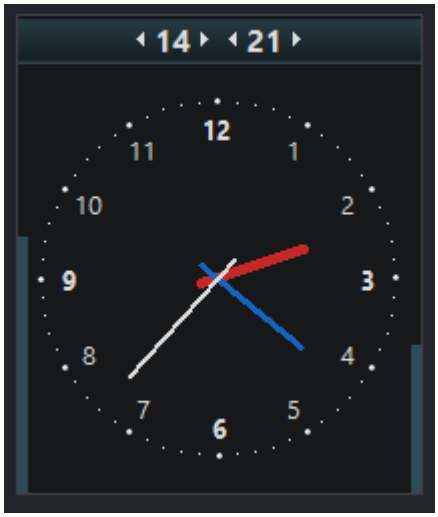
Subject: Re: New dark theme thread
Posted by cbpporter on Mon, 22 Apr 2019 11:22:53 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Mon, 22 April 2019 13:36
1. I could not figure out how to control the color for those two bars on the left and right. Looks like some frame/backpaint problems.

Yeah, figured it out: those are two LineCtrl!

File Attachments

1) comp4.png, downloaded 2223 times



Subject: Re: New dark theme thread
Posted by cbpporter on Mon, 22 Apr 2019 15:50:17 GMT
View Forum Message <> Reply to Message

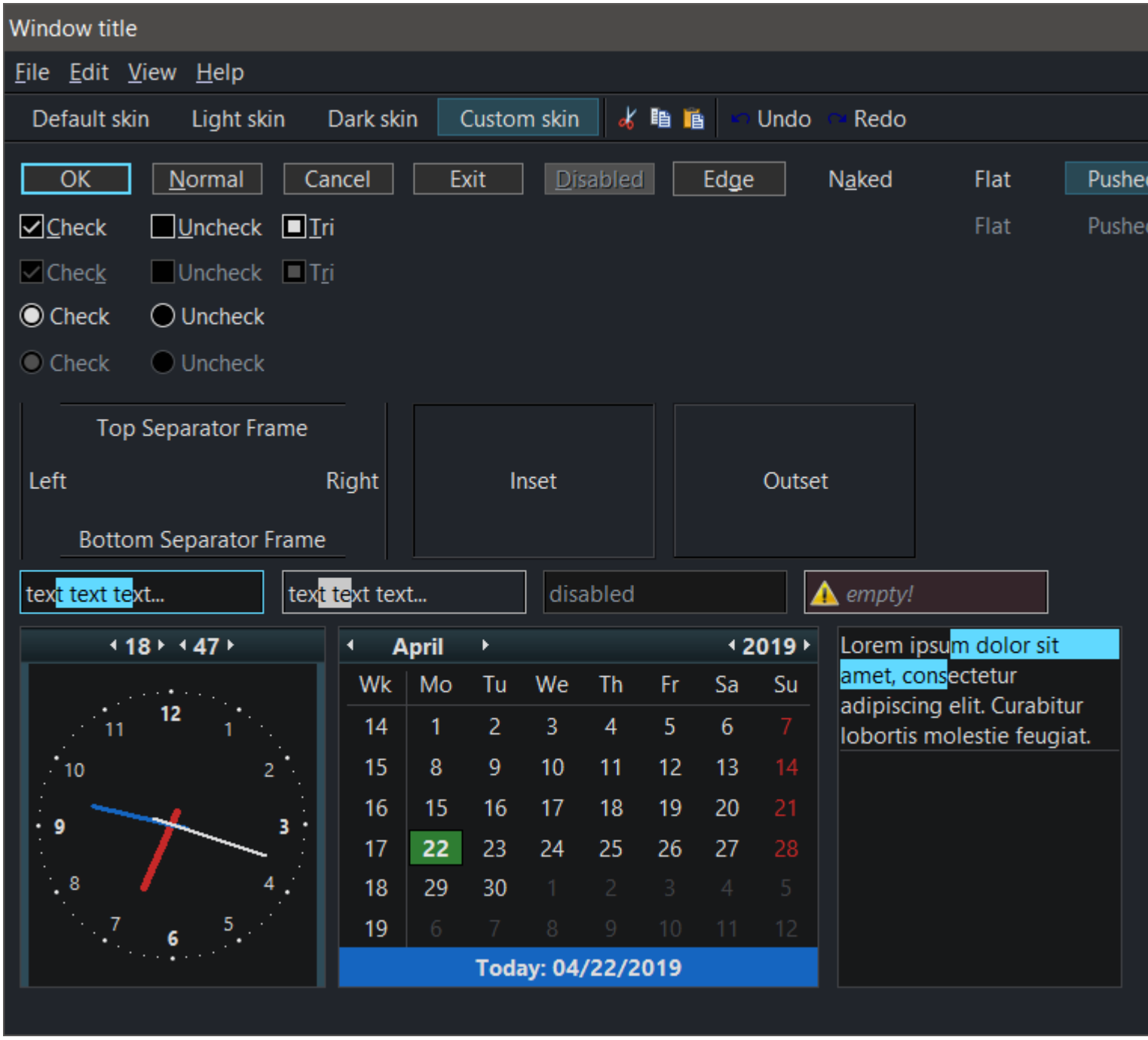
I'm relatively close to success...

Default autodetects dark and light, but you can also force dark, force light our use my custom skin:

Calendar needs a bit more work...

File Attachments

1) comp5.png, downloaded 2231 times



Subject: Re: New dark theme thread
Posted by mr_ped on Mon, 22 Apr 2019 20:22:14 GMT
View Forum Message <> Reply to Message

Seems like good progress to me.

Still looks kinda amateur compared to KDE5 dark breeze theme (default dark in KDE5), but then again, that thing took decade+ to end in its current state, so I don't expect anyone to catch up after few hours of work, for the amount of time you are working on this, it looks very very good.

Subject: Re: New dark theme thread
Posted by cbpporter on Mon, 22 Apr 2019 21:08:37 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Mon, 22 April 2019 23:22Seems like good progress to me.

Still looks kinda amateur compared to KDE5 dark breeze theme (default dark in KDE5), but then again, that thing took decade+ to end in its current state, so I don't expect anyone to catch up after few hours of work, for the amount of time you are working on this, it looks very very good. Thanks!

I'm between two worlds: half of my effort is fixing bugs and parametrizing the skin for a future custom skin (make the current U++ implementation look good, but in line with what exists today), half trying to implement a good skin (I like this one https://midnightsketch.com/)

I wouldn't use exactly those separator frames, but for now they need to work correctly with dark mode and they don't really, so for now it will look like this.

The left separator frame is still wrong I think.

Once everything is fully skinable to my liking and I ran tests even with RNG colors, we can work on making a custom skin that doesn't care about Win10 and looks hopefully less amateurish.

Subject: Re: New dark theme thread
Posted by cbpporter on Tue, 23 Apr 2019 10:33:22 GMT
View Forum Message <> Reply to Message

Today I familiarized myself with the various TabCtrls.

In the screenshot bellow you can see that booting up in light mode then turning on dark mode creates some black on black tabs. This is fixed with my dynamic theme change support and at the bottom you can see the new skin for the tabs.

I couldn't skin the body of the tab and the tabs themselves no mater how much I looked over ChHostSkin. For now it is a mystery how that works and where does that black come from... Hopefully one day I can skin them with SColorFace().

Oh, and there are more shenanigans with FileTabs icons. First of all, there is like a 10% chance they show up. Most of the time they are not visible at all. Then they are either light or dark per execution of the application. Weird... I'm investigating NativePathIcon, maybe that is the problem...

File Attachments

1) comp6.png, downloaded 2145 times


aaa
bbb
tab1
tab2
tab3
tab4
tab5
file2
file3
file4
file5
tab1
tab2
tab3
tab4
tab5
tab6
tab7
tab8
tab9
tab10
tab11
file2
file3
file4
file5
file6
file7
file8
file9
file10
file11


aaa
bbb
tab1
tab2
tab3
tab4
tab5
file1.txt
file2.txt
file3.txt
file4.txt
file5.txt
tab1
tab2
tab3
tab4
tab5
tab6
tab7
tab8
tab9
tab10
tab11
file1.txt
file2.txt
file3.txt
file4.txt
file5.txt
file6.txt
file7.txt

Subject: Re: New dark theme thread
Posted by cbpporter on Tue, 23 Apr 2019 12:48:11 GMT
View Forum Message <> Reply to Message

This is quite the (unplanned) project!

I am restructuring my effort into 3 subprojects...

Project A
This will be a fixed vanilla dark mode. It will look like today's U++ dark mode, trying to emulate system colors. The idea is that you set dark theme, drop any control in any window, do nothing else special and it will look alright, no bugs, no contrast issues.

This means that the dark mode will continue to be "high contrast".

A few minor fixes and changes are needed to U++ and I will want these changes committed to the official repo. Basically a few things can't be skined and a few bugs.

Part of this project will be to provide hand edited darkmode icons where needed for the default ones, like file open and copy paste, but this part I'm probably not going to do myself.

Project B
This will take fixes from Project A and add on top of them a new low contrast dark skin, which mean black will be replaced by dark gray, super light gray/white will be replaced by darker gray, including for things like FieldFrame.

Pro: this skin will look nice and modern and will not be high contrast and eye straining.

Cons: some of the system looks, like buttons and edit fields and tabs will no longer be system but will be fully custom and theme specific.

I will put this into bazaar or something and use it for my apps, never using the U++ default. And since it is heavily skinned, you can use it on Linux too if you want a consistent experience.

Project C
This will take the skin from Project B and make some very specific non-universal changes to allow all my apps to seamlessly change skin at runtime.

This will offer the following guarantee: an app launching with dark mode will be pixel perfect equal to the same app launched in light mode, but the user has switched to dark mode.

Subject: Re: New dark theme thread
Posted by cbpporter on Wed, 24 Apr 2019 09:18:48 GMT
View Forum Message <> Reply to Message

So Project A is in full force!

To reiterate, I have two new skins and on the fly skin changing at runtime, but the goal of Project A is to make more U++ compatible with my solution with minimal changes while at the same time fixing bugs.

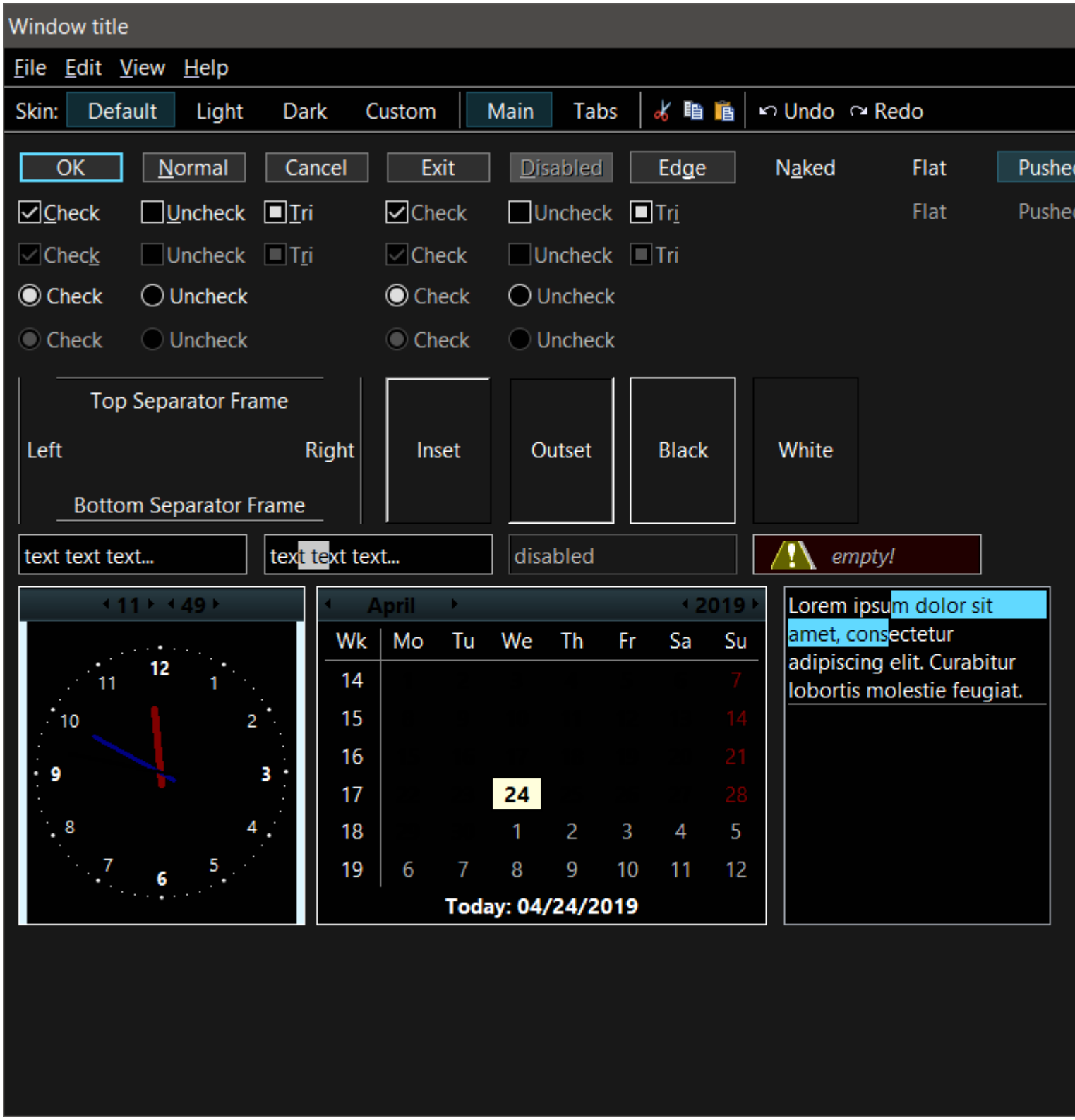
So standard U++ Dark mode attached.

I'll submit the fixes in the main thread.

Before:

File Attachments

1) before1.png, downloaded 2209 times

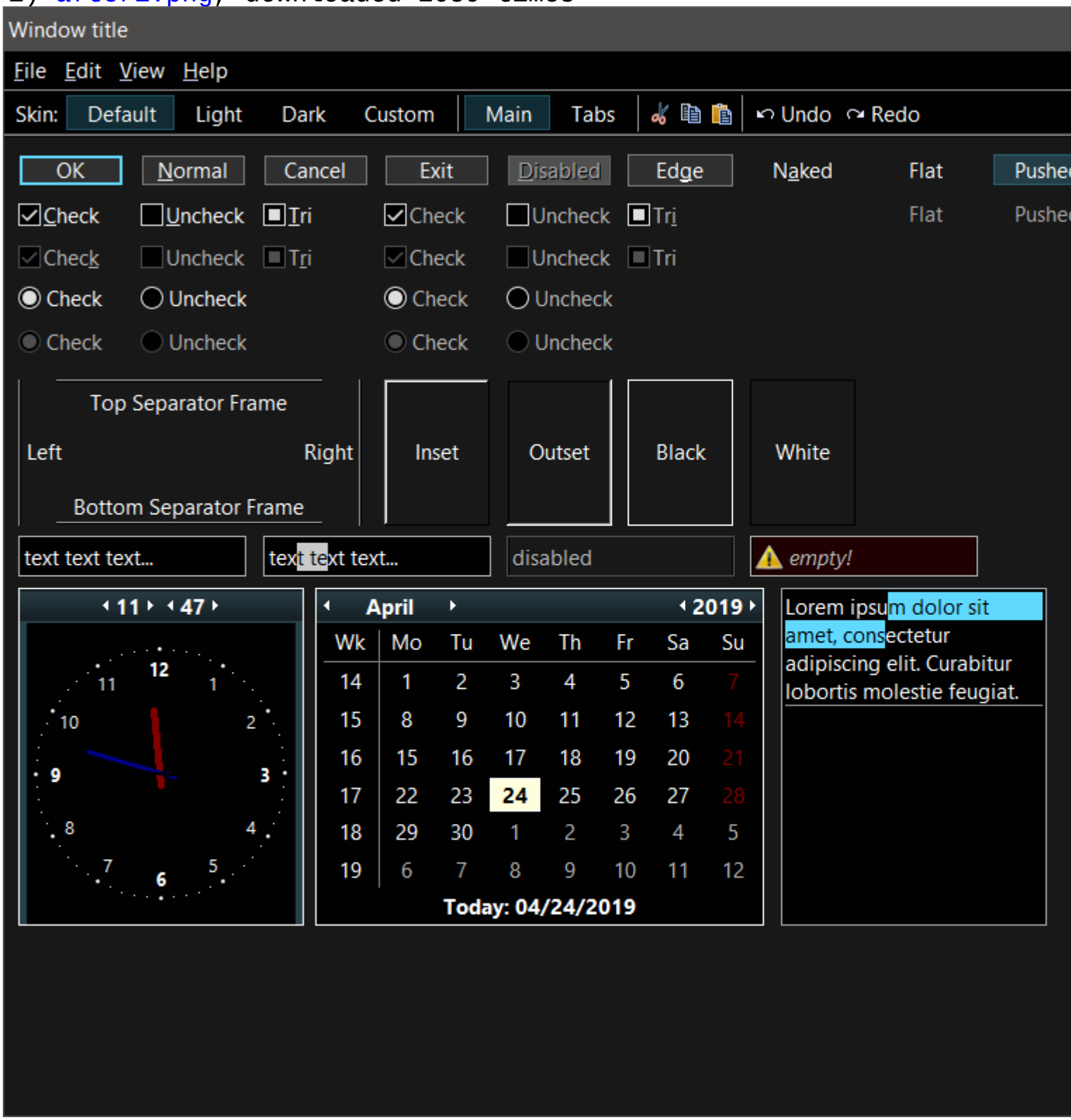


Subject: Re: New dark theme thread
Posted by cbpporter on Wed, 24 Apr 2019 09:20:00 GMT
View Forum Message <> Reply to Message

After:

File Attachments

1) after1.png, downloaded 2059 times



Subject: Re: New dark theme thread
Posted by cbpporter on Thu, 25 Apr 2019 10:15:23 GMT
View Forum Message <> Reply to Message

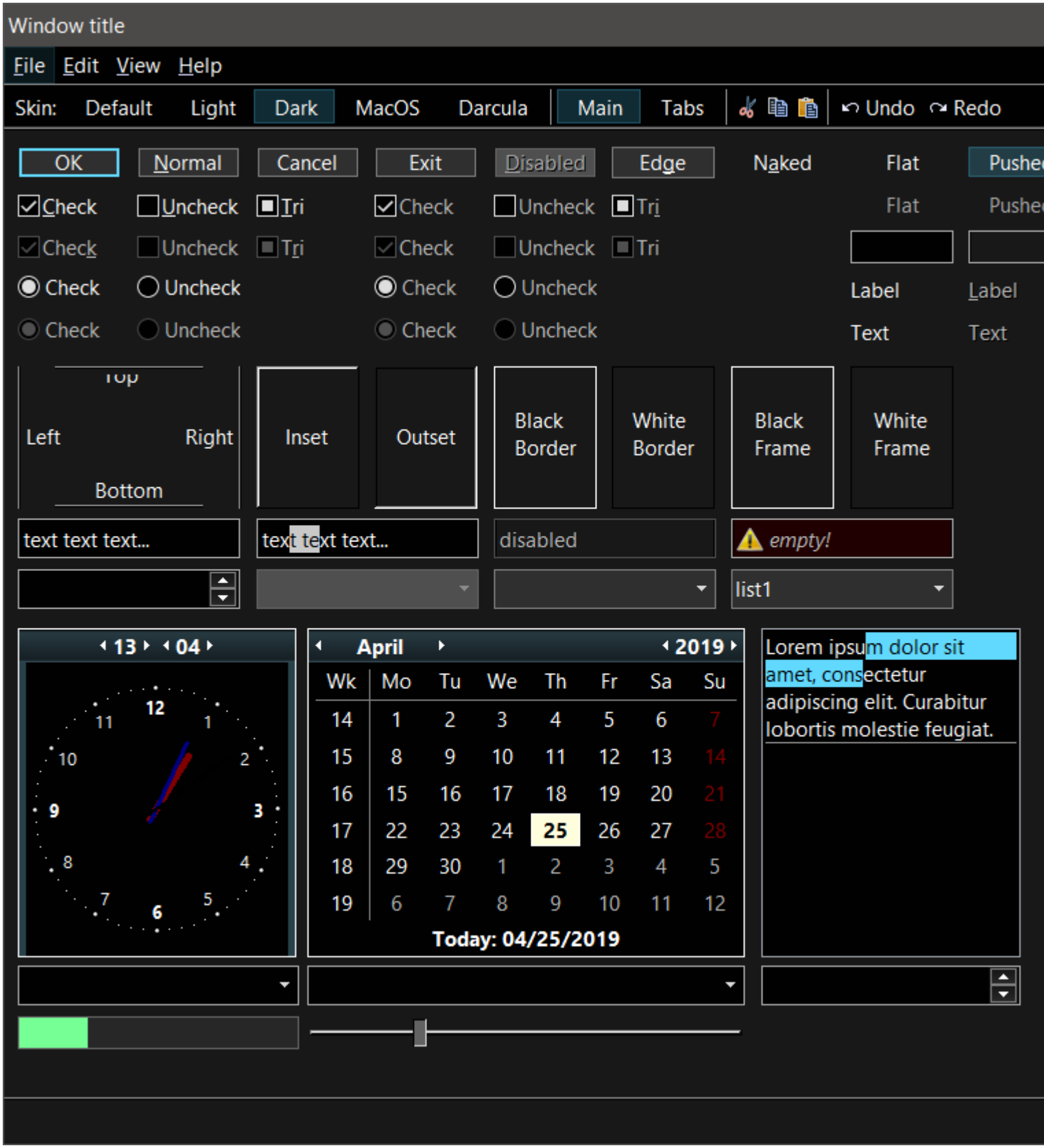
Well, Easter is almost here, so I'll be out for a week.

So here is a small update.

comp7a is U++ darkmode.

File Attachments

1) comp7a.png, downloaded 1197 times


Window title
File Edit View Help
Skin: Default Light Dark MacOS Darcula Main Tabs Undo Redo
OK Normal Cancel Exit Disabled Edge Naked Flat
Check Uncheck Tri
Label
Text
Left Right Bottom
Inset Outset
Black Border White Border Black Frame White Frame
text text text...
disabled
empty!
list1
April 2019
Wk Mo Tu We Th Fr Sa Su
Today: 04/25/2019
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur lobortis molestie feugiat.

Subject: Re: New dark theme thread
Posted by cbpporter on Thu, 25 Apr 2019 10:17:06 GMT
View Forum Message <> Reply to Message

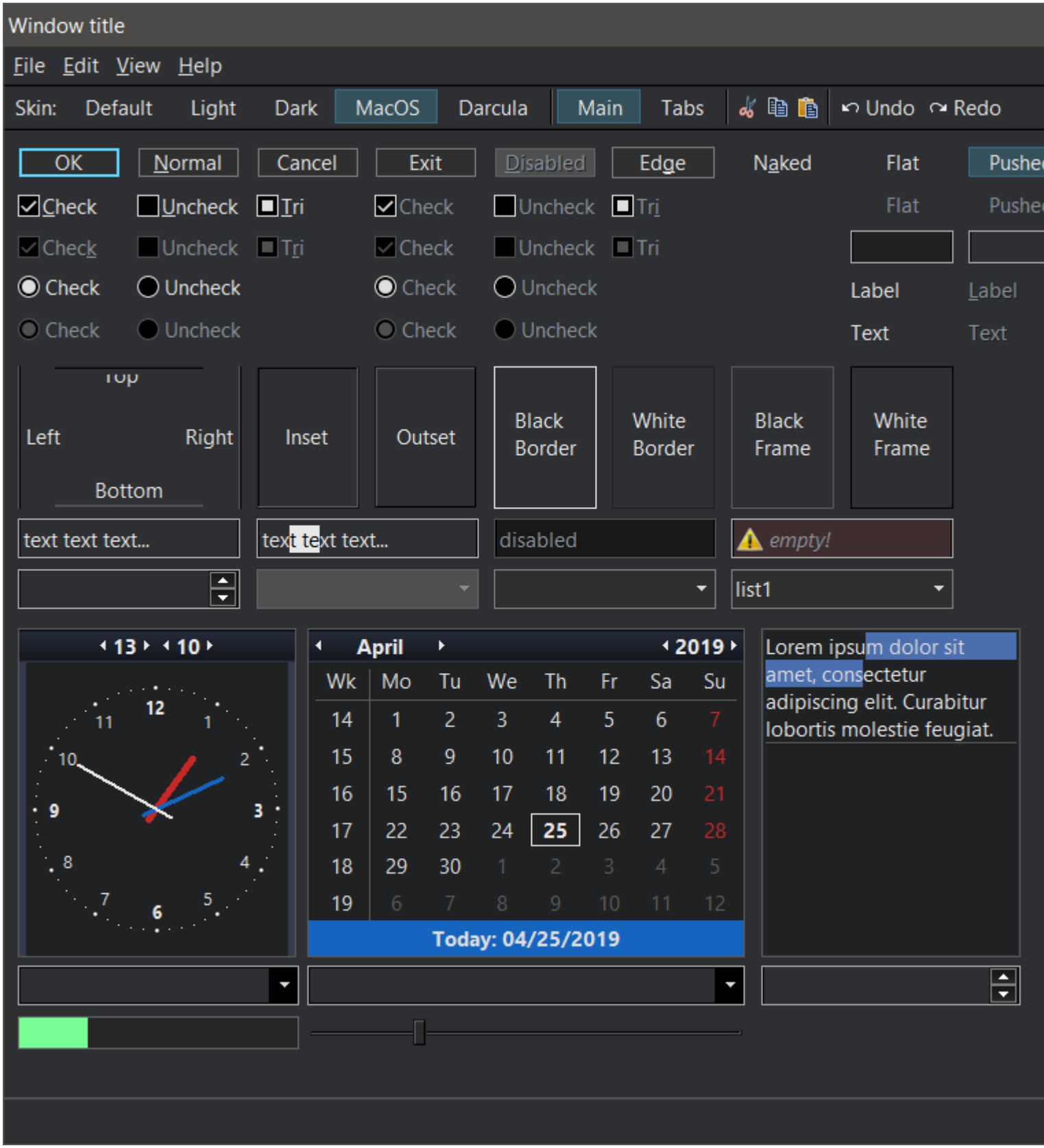
comp7b is recoloring U++ dark mode, inspired by macos.

I added dynamic blending to borders and don't really like it. Needs more tuning.

Something like this is close to the limits of what you can do with just recolors.

File Attachments

1) comp7b.png, downloaded 1203 times

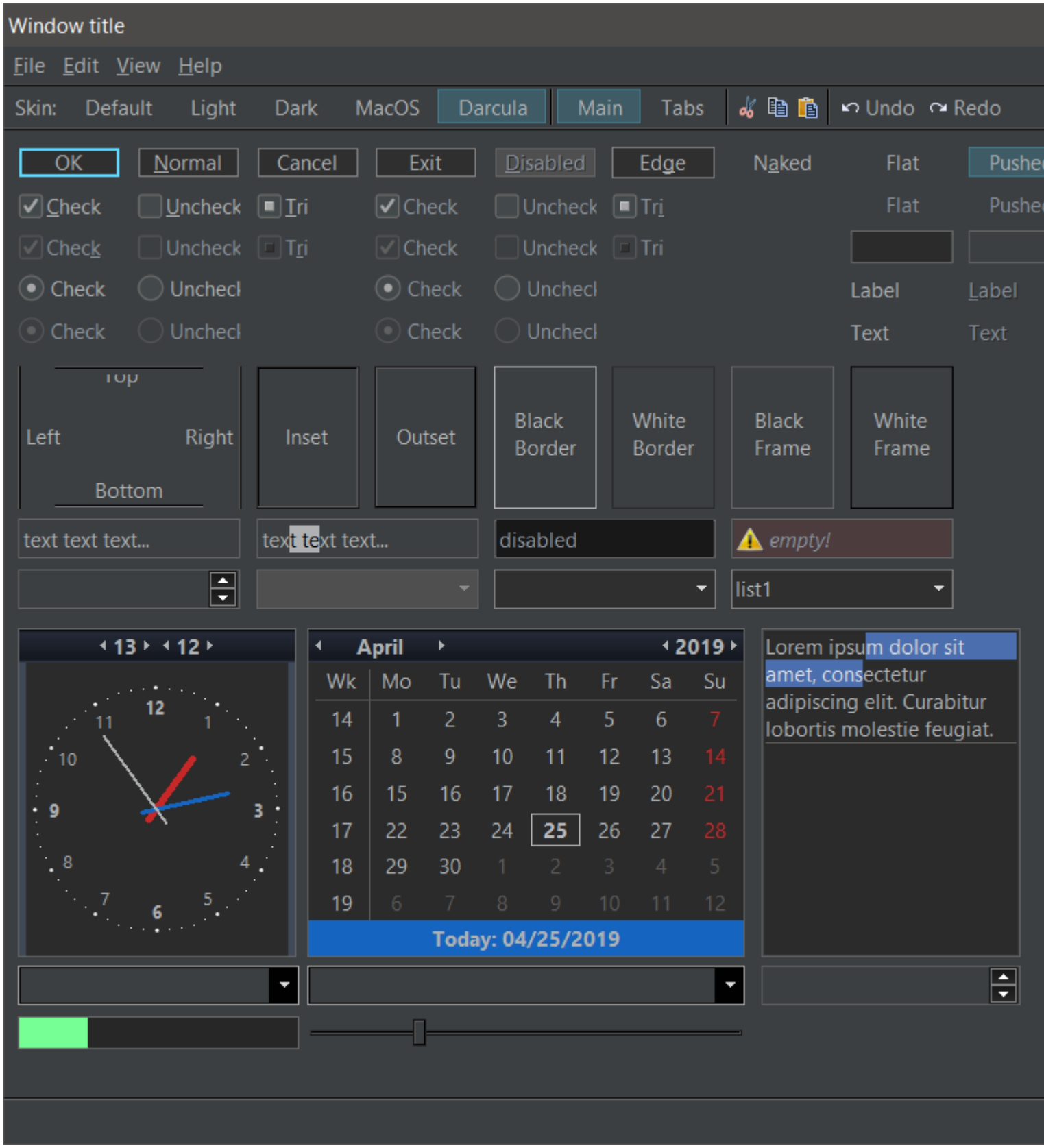

Window title
File Edit View Help
Skin: Default Light Dark MacOS Darcula Main Tabs Undo Redo
OK Normal Cancel Exit Disabled Edge Naked Flat Pushe
Check Uncheck Tri
Label Text
Left Right Bottom
Inset Outset
Black Border White Border
Black Frame White Frame
text text text...
disabled
empty!
list1
April 2019
Wk Mo Tu We Th Fr Sa Su
Today: 04/25/2019
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur lobortis molestie feugiat.

Subject: Re: New dark theme thread
Posted by cbpporter on Thu, 25 Apr 2019 10:18:33 GMT
View Forum Message <> Reply to Message

And finally, comp7c is a little prototype I've been working on that not only recolors, but reskins.

File Attachments

1) comp7c.png, downloaded 1243 times

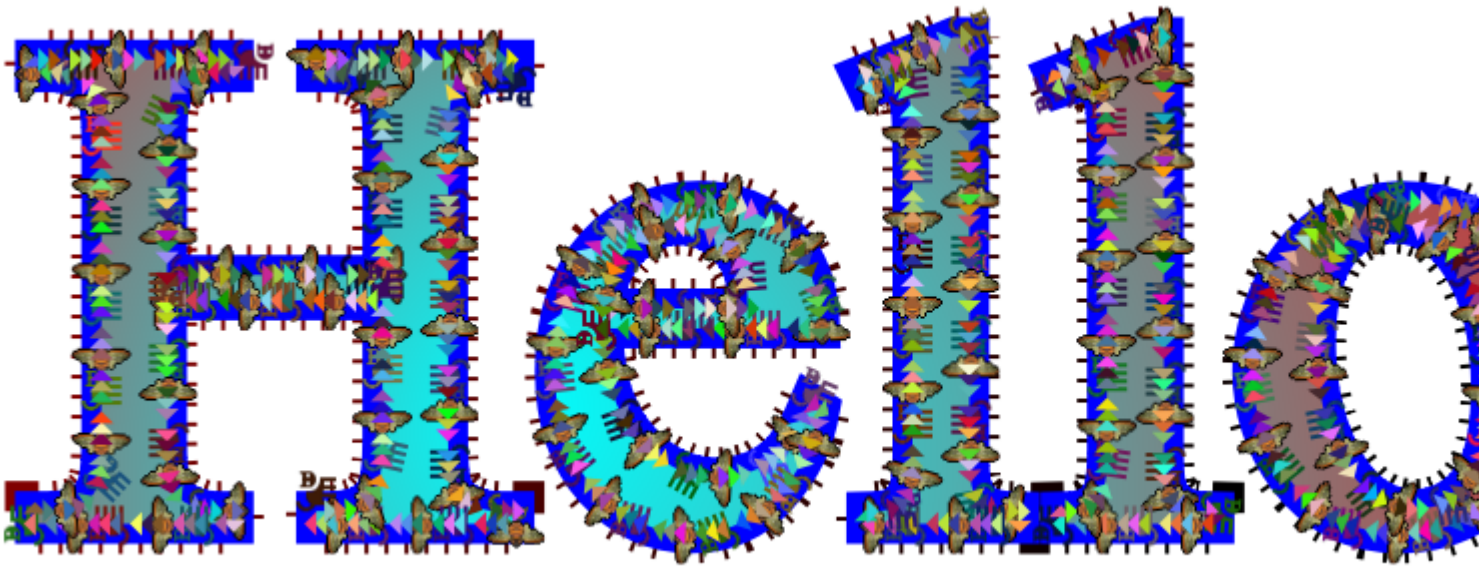

Window title
File Edit View Help
Skin: Default Light Dark MacOS Darcula Main Tabs Undo Redo
OK Normal Cancel Exit Disabled Edge Naked Flat Pushe
Check Uncheck Tri Check Uncheck Tri Flat Pushe
Check Uncheck Tri Check Uncheck Tri
Check Uncheck Check Uncheck Label Label
Check Uncheck Check Uncheck Text Text
Top
Left Right
Bottom
Inset
Outset
Black Border
White Border
Black Frame
White Frame
text text text...
text text text...
disabled
empty!
list1
13 12
April 2019
Wk Mo Tu We Th Fr Sa Su
14 1 2 3 4 5 6 7
15 8 9 10 11 12 13 14
16 15 16 17 18 19 20 21
17 22 23 24 25 26 27 28
18 29 30 1 2 3 4 5
19 6 7 8 9 10 11 12
Today: 04/25/2019
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur lobortis molestie feugiat.

Subject: Re: New dark theme thread
Posted by cbpporter on Thu, 25 Apr 2019 13:59:17 GMT
View Forum Message <> Reply to Message

Well, I just discovered the power of Painter!

Might create my skinned GUI with Painter plus some scripting. Maybe Esc?

File Attachments
1) 2019-04-25.png, downloaded 2130 times



Subject: Re: New dark theme thread
Posted by aquilarubra on Tue, 22 Sep 2020 16:53:32 GMT
View Forum Message <> Reply to Message

I have this:

```
GUI_APP_MAIN
{
 Ctrl::SetDarkThemeEnabled();
```

But nothing happens. What am I missing?

Subject: Re: New dark theme thread
Posted by Klugier on Tue, 22 Sep 2020 22:48:40 GMT
View Forum Message <> Reply to Message

Hello Aquilarubra,

As far as I know dark theme is inherit from operating system. So, if you set in your OS theme that is dark then upp app should use it. Please noticed that it only happen on app loading. If you modify it on runtime then it doesn't work. It is doable, but require some changes on uppsrc side.

So, what "Ctrl::SetDarkThemeEnabled();" do. From the documentation we can read:
Quote:
Hints coloring and skinning algorithms that the application is ready to work in dark theme mode. Currently this enables dark theme in Windows (with other platforms dark theme is enabled by default). Default value is true. Disabling dark theme has to be done before GUI_APP_MAIN, e.g. in INITBLOCK.

So, it will do not transform your app into dark especially on Linux (I do not test on Windows). This should be enable on OS level and then app should be launch and inherit this settings from OS.

Klugier

Subject: Re: New dark theme thread
Posted by mirek on Wed, 23 Sep 2020 15:30:11 GMT
View Forum Message <> Reply to Message

In addition to what Klugier wrote:

You can however set "internal" dark skin

```
#include <CtrlLib/CtrlLib.h>

using namespace Upp;

GUI_APP_MAIN
{
 Ctrl::SetSkin(ChDarkSkin);
 PromptOK("Hello!");
}
```

Subject: Re: New dark theme thread

Posted by Tom1 on Tue, 21 Dec 2021 15:32:02 GMT
View Forum Message <> Reply to Message

Hi,

I may be a year or two late here, but since then I have experimented with the dark mode in Windows 10 and 11, and just find the U++ internal ChDarkSkin a little too black and high contrast. Also, ScrollBar in default Dark mode does not show properly at all on Windows 11. Additionally, it seems that Microsoft has more or less abandoned the 'system colors' and even their own apps appear to use their own internally defined colors instead. Please compare e.g. Edge and Windows Explorer and Notepad in dark mode...

I thought I could suggest some coloring changes to ChDarkSkin / ChFlatDarkSkin or maybe even the default dark theme in Windows. However, after reading this thread, I realized that cbpporter was way ahead with this already and went far beyond what I could dream of. So, could this work or something similar be included in U++ to give a visually appealing look with Windows Dark mode? Maybe something that resembles the average darkness and contrast of other Windows applications in Dark mode?

Best regards,

Tom

Subject: Re: New dark theme thread
Posted by Tom1 on Mon, 03 Jan 2022 14:28:01 GMT
View Forum Message <> Reply to Message

Hi,

I found out that sometimes it's good to re-read some docs, as they get updated:

https://docs.microsoft.com/en-us/windows/win32/api/winuser/n f-winuser-getsyscolor
https://docs.microsoft.com/en-us/windows/apps/design/style/c olor

From the GetSysColor() documentation, I have picked the following:

Colors no longer available in Windows 10/11:
COLOR_3DDKSHADOW
COLOR_3DHIGHLIGHT
COLOR_3DHILIGHT
COLOR_3DLIGHT
COLOR_3DSHADOW
COLOR_ACTIVEBORDER
COLOR_ACTIVECAPTION
COLOR_APPWORKSPACE
COLOR_BACKGROUND
COLOR_BTNFACE

COLOR_BTNHIGHLIGHT
COLOR_BTNHILIGHT
COLOR_BTNSHADOW
COLOR_CAPTIONTEXT
COLOR_DESKTOP
COLOR_GRADIENTACTIVECAPTION
COLOR_GRADIENTINACTIVECAPTION
COLOR_INACTIVEBORDER
COLOR_INACTIVECAPTION
COLOR_INACTIVECAPTIONTEXT
COLOR_INFOBK
COLOR_INFOTEXT
COLOR_MENU
COLOR_MENUHILIGHT
COLOR_MENUBAR
COLOR_MENUTEXT
COLOR_SCROLLBAR
COLOR_WINDOWFRAME

Colors available in Windows 10/11:

COLOR_WINDOW - Window background
COLOR_WINDOWTEXT - Headings, body, lists, app and window borders...
COLOR_HOTLIGHT - Hyperlinks
COLOR_GRAYTEXT - Inactive (disabled) UI
COLOR_HIGHLIGHTTEXT - Foreground color for text or UI that is selected/interacted with (hover, pressed) or in progress
COLOR_HIGHLIGHT - Background or accent for COLOR_HIGHLIGHTTEXT
COLOR_BTNTEXT - Foreground color for buttons and UI with interaction (use with COLOR_3DFACE)
COLOR_3DFACE - Background for COLOR_BTNTEXT

Maybe, the system color list in ChWin32.cpp should be updated to use only colors that are still available on Windows 10/11:

```
...
static sysColor sSysColor[] = {
 { SColorPaper_Write, COLOR_WINDOW },
...
```

In any case, the dark mode is too high contrast to look at for a full day. E.g. TheIDE has a very nice contrast on Linux Mint following some basic dark theme I picked from the basic list. However, when in Dark mode on Windows, the GetSysColor(COLOR_WINDOW) returns "Color(1, 1, 1)" and this is simply too dark. Something between Color(32, 32, 32) and Color(48, 48, 48), would work better as the dark background. The invisibility issue with ScrollBar button arrows also make it difficult to use.

Best regards,

Tom

Subject: Re: New dark theme thread
Posted by mirek on Wed, 05 Jan 2022 08:13:42 GMT
View Forum Message <> Reply to Message

OK, 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

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.

Subject: Re: New dark theme thread
Posted by Tom1 on Wed, 05 Jan 2022 15:15:25 GMT
View Forum Message <> Reply to Message

Mirek,

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,&Buff
erSize);

          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 Ctrls in e.g. Style manager.

Best regards,

Tom

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,&Buff
erSize);

          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 Ctrls 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)

Subject: Re: New dark theme thread
Posted by mirek on Thu, 06 Jan 2022 07:46:01 GMT
View Forum Message <> Reply to Message

P.S.: Alternatively, you can check Firefox codebase as they seem to get windows system colors right (but I guess the method might be the same in the end

Mirek

Subject: Re: New dark theme thread
Posted by Tom1 on Thu, 06 Jan 2022 11:34:42 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 06 January 2022 09:46P.S.: Alternatively, you can check Firefox codebase as they seem to get windows system colors right (but I guess the method might be the same in the end

Mirek

Probably yes, the method in FF must be similar: In fact, their colors are close but not exact match to those in Windows dark theme. FF does not seem to pick Windows accent color, but uses the default blue for the related items instead. (Tried Orange bright accent color to easily see where it is used.)

I'm now on Windows 10 and the default dark theme behavior in UWord is here much better than it is on Windows 11.

BTW, in U++: Which SColor(s) are used for:
- main menu bar background
- tool bar background
- status bar background

- scroll bar background?

I cannot seem to find which one to change...

Best regards,

Tom

Subject: Re: New dark theme thread
Posted by mirek on Thu, 06 Jan 2022 18:48:17 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Thu, 06 January 2022 12:34mirek wrote on Thu, 06 January 2022 09:46P.S.: Alternatively, you can check Firefox codebase as they seem to get windows system colors right (but I guess the method might be the same in the end

Mirek

Probably yes, the method in FF must be similar: In fact, their colors are close but not exact match to those in Windows dark theme. FF does not seem to pick Windows accent color, but uses the default blue for the related items instead. (Tried Orange bright accent color to easily see where it is used.)

I'm now on Windows 10 and the default dark theme behavior in UWord is here much better than it is on Windows 11.

BTW, in U++: Which SColor(s) are used for:
- main menu bar background
- tool bar background
- status bar background
- scroll bar background?

I cannot seem to find which one to change...

Best regards,

Tom

A lot of them is image inversion unfortunately... (theming API gives me only light appearance, so I invert it to dark).

Subject: Re: New dark theme thread
Posted by Tom1 on Fri, 07 Jan 2022 15:51:32 GMT
View Forum Message <> Reply to Message

Hi Mirek,

There sure are better color artists than myself. Anyway, here's approximately what I have in mind. I've tried to follow Windows 11 default dark theme, but I just cannot get everything right.

Best regards,

Tom

EDIT: As there does not seem to be official dark theme colors available in Windows, anything that looks pretty much the same as dark built-in apps in Windows, is good enough for me. I have looked at Windows Explorer and Windows Settings primarily for the coloring.

EDIT2: Deleted the attachment. Please use the latest instead.

Subject: Re: New dark theme thread
Posted by mirek on Sat, 08 Jan 2022 13:23:19 GMT
View Forum Message <> Reply to Message

I am mostly ok with that (and accepted it to codebase), however I would like those blue frames of buttons (with defualt button having thicker frame) retained.

Also, it would be good to make DropList popup table frame somewhat more gray.

Mirek

Subject: Re: New dark theme thread
Posted by Tom1 on Sat, 08 Jan 2022 16:16:18 GMT
View Forum Message <> Reply to Message

mirek wrote on Sat, 08 January 2022 15:23I am mostly ok with that (and accepted it to codebase), however I would like those blue frames of buttons (with defualt button having thicker frame) retained.

Also, it would be good to make DropList popup table frame somewhat more gray.

Mirek

Mirek,

Changed the above mentioned items. Also made the TabCtrl tabs look more in line with the rest of the Dark GUI. Please find attached the revised ChWin32.cpp zipped.

TODO: I would still like to have:

1. changed the white/blue/white stripe below TheIDE file tabs to something grayish. Maybe the same color as the active tab.
2. changed the background of unselected Switch and Option to transparent to nicely fit on any surface.

...but I have no idea how to do either of those.

Best regards,

Tom

EDIT: Deleted the attachment. Please use the latest instead.

Subject: Re: New dark theme thread
Posted by Tom1 on Wed, 12 Jan 2022 08:56:42 GMT
View Forum Message <> Reply to Message

Hi,

In the above attempts I made a bug in scrollbar button visualization. The attached code fixes this issue and cleans up the color definition a little.

Best regards,

Tom

EDIT: Further changed SColorHighlight and SColorHighlightText to Windows system colors as similar colors seem to be used in Dark color mode too in the sample images in Windows UI Color document. (Windows Explorer does not use the same blue, but gray for file selection though...)

File Attachments
1) ChWin32.zip, downloaded 391 times

Subject: Re: New dark theme thread
Posted by Tom1 on Thu, 13 Jan 2022 12:22:04 GMT
View Forum Message <> Reply to Message

Hi,

I found this about colors in W11:

https://docs.microsoft.com/en-us/windows/apps/design/signatu re-experiences/color#accent-color

-

Generally, I think MS is expecting apps to use 'Windows UI library' where colors are defined instead of Windows system colors. The accent color (as per the above link) is supposed to be used sparingly for highlighting specific interactive spots on the GUI.

In real life, when running different apps, I find each of them looking a bit different (in color and style). Also, most apps do not use the accent color for anything, or just use the default accent color instead of the user selected accent color. The overall GUI experience even in Windows 11 itself is currently a mix of design styles and colors from many preceding Windows versions in addition to the current one. Maybe 'Windows Settings' and 'Calculator' are one of the few apps in Windows that actually follow the current MS principles in UI design.

Where should U++ stand here? Should it attempt to provide a 'Windows UI library' -like look on Windows 10 and 11? I'm not an expert on CtrlLib (either), but I'm afraid that might require quite a lot of work on small details.

Best regards,

Tom

Subject: Re: New dark theme thread
Posted by mirek on Sun, 16 Jan 2022 18:15:31 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Thu, 13 January 2022 13:22Hi,

I found this about colors in W11:

https://docs.microsoft.com/en-us/windows/apps/design/signatu
re-experiences/color#accent-color

-

Generally, I think MS is expecting apps to use 'Windows UI library' where colors are defined instead of Windows system colors. The accent color (as per the above link) is supposed to be used sparingly for highlighting specific interactive spots on the GUI.

In real life, when running different apps, I find each of them looking a bit different (in color and style). Also, most apps do not use the accent color for anything, or just use the default accent color instead of the user selected accent color. The overall GUI experience even in Windows 11 itself is currently a mix of design styles and colors from many preceding Windows versions in addition to the current one. Maybe 'Windows Settings' and 'Calculator' are one of the few apps in Windows that actually follow the current MS principles in UI design.

Where should U++ stand here? Should it attempt to provide a 'Windows UI library' -like look on Windows 10 and 11? I'm not an expert on CtrlLib (either), but I'm afraid that might require quite a

lot of work on small details.

Best regards,

Tom

I thought we have already established that for dark mode we just choose our own colors?

Mirek

Subject: Re: New dark theme thread
Posted by Tom1 on Mon, 17 Jan 2022 07:50:07 GMT
View Forum Message <> Reply to Message

mirek wrote on Sun, 16 January 2022 20:15

I thought we have already established that for dark mode we just choose our own colors?

Mirek

Hi Mirek,

As I found relevant information about the subject, I felt it as my responsibility to make it available for evaluation here. If you're pleased with the current plan, despite the new found information, then that's it. You're the boss here.

But please merge my latest ChWin32.cpp (somewhere above), as it fixes my stupid bug with the ScrollBar button look, among with other small improvements.

Best regards,

Tom

Subject: Re: New dark theme thread
Posted by mirek on Mon, 17 Jan 2022 16:37:31 GMT
View Forum Message <> Reply to Message

Merged.

Subject: Re: New dark theme thread
Posted by Tom1 on Tue, 18 Jan 2022 14:32:46 GMT

View Forum Message <> Reply to Message

Quote:Merged.
Thanks!

Best regards,

Tom

Subject: Re: New dark theme thread
Posted by Alboni on Tue, 17 May 2022 13:16:24 GMT
View Forum Message <> Reply to Message

Cool!
Can that theme be adjusted? The client says it's ugly.

Subject: Re: New dark theme thread
Posted by Tom1 on Tue, 17 May 2022 15:54:46 GMT
View Forum Message <> Reply to Message

Hi Alboni,

A lot can be changed for a specific application. I'm not an expert here, but if you put this into the beginning of your GUI_APP_MAIN you can have a different set of colors and more:

```
if(IsDarkTheme()){
 ChReset();
 SColorPaper_Write(Color(32, 32, 32));
 SColorFace_Write(Color(40, 40, 40));
 SColorText_Write(Color(255, 255, 255));
 SColorHighlight_Write(Color(64, 64, 64));
 SColorHighlightText_Write(Color(255, 255, 255));
 SColorMenu_Write(Color(48, 48, 48));
 SColorMenuText_Write(Color(255, 255, 255));
 SColorInfo_Write(Color(48, 48, 48));
 SColorInfoText_Write(Color(255, 255, 255));
 SColorLight_Write(Color(32, 32, 32));
 SColorShadow_Write(Color(64, 64, 64));
 SColorDisabled_Write(Color(128, 128, 128));
 static int adj[] = { 10, 80, -5, -10 };

 ChMakeSkin(3, Color(64, 64, 64), Color(64, 64, 64), adj);

 // ... and a lot more ...
}
```

Please look at uppsrc to find out how different items are changed. (The above merely touches the surface.) You may find the implementation of these interesting:
ChFlatDarkSkin();
ChFlatGraySkin();
ChDarkSkin();
ChFlatSkin();
ChGraySkin();
ChHostSkin();
ChClassicSkin();
ChStdSkin();
ChBaseSkin();
Best regards,

Tom

EDIT: PS. Making that particular kind of client happy, takes a lot of work, so please remember to charge by the hour.

Subject: Re: New dark theme thread
Posted by Alboni on Wed, 18 May 2022 13:51:43 GMT
View Forum Message <> Reply to Message

Thanks, Tom.
Charging extra isn't really possible, as we don't charge for the software at all, but rather by charging our client's clients thru third parties, but I think I'll make a quick utility that will save some kde color profiles.

Ruben

Subject: Re: New dark theme thread
Posted by superdev on Fri, 26 Aug 2022 17:35:19 GMT
View Forum Message <> Reply to Message

SetSkin(ChDarkSkin) doesn't make TextCtrl dark
