

---

Subject: Dynamic theme changes

Posted by [mirek](#) on Fri, 06 Dec 2024 09:20:34 GMT

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

---

Before this change, U++ loaded host platform skin only once on start for simplicity and further changes of theme were ignored.

This is now fixed, but the problem is a bit more complicated, so you have to activate this in GUI\_APP\_MAIN, docs near the end:

[https://www.ultimatepp.org/dev/srcdoc\\$Draw\\$UhdAndDarkTheme\\_en-us.html](https://www.ultimatepp.org/dev/srcdoc$Draw$UhdAndDarkTheme_en-us.html)

---

---

Subject: Re: Dynamic theme changes

Posted by [Novo](#) on Sat, 07 Dec 2024 05:05:34 GMT

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

---

This change broke examples/BlueBar.

umk examples BlueBar CLANG -bus

UWord.cpp

/home/buildbot/worker/l-upp/build/uppsrc/art/BlueBar/BlueSkin.cpp:12:8: error: no member named 'Standard' in 'Upp::ToolBar::Style'

```
s = s.Standard();
```

```
~ ^
```

/home/buildbot/worker/l-upp/build/uppsrc/art/BlueBar/BlueSkin.cpp:32:8: error: no member named 'Standard' in 'Upp::MenuBar::Style'

```
s = s.Standard();
```

```
~ ^
```

2 errors generated.

---

---

Subject: Re: Dynamic theme changes

Posted by [mirek](#) on Sat, 07 Dec 2024 08:04:12 GMT

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

---

Novo wrote on Sat, 07 December 2024 06:05This change broke examples/BlueBar.

umk examples BlueBar CLANG -bus

UWord.cpp

/home/buildbot/worker/l-upp/build/uppsrc/art/BlueBar/BlueSkin.cpp:12:8: error: no member named 'Standard' in 'Upp::ToolBar::Style'

```
s = s.Standard();
```

```
~ ^
```

/home/buildbot/worker/l-upp/build/uppsrc/art/BlueBar/BlueSkin.cpp:32:8: error: no member named 'Standard' in 'Upp::MenuBar::Style'

```
s = s.Standard();
```

```
~ ^
```

2 errors generated.

Thanks, fixed.

---

---

Subject: Re: Dynamic theme changes  
Posted by [Tom1](#) on Sat, 07 Dec 2024 11:04:21 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi,

TheIDE Errors panel does not seem to follow dynamic skin changes yet. Maybe it needs a Skin()?

Best regards,

Tom

---

---

Subject: Re: Dynamic theme changes  
Posted by [mirek](#) on Sat, 07 Dec 2024 11:58:33 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Tom1 wrote on Sat, 07 December 2024 12:04Hi,

TheIDE Errors panel does not seem to follow dynamic skin changes yet. Maybe it needs a Skin()?

Best regards,

Tom

Thanks, hopefully fixed (also, in the process, I made SColor evaluation after skin change lazy)

---

---

Subject: Re: Dynamic theme changes  
Posted by [Tom1](#) on Sat, 07 Dec 2024 14:52:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi,

Thanks, it works fine now.

BR, Tom

---

---

Subject: Re: Dynamic theme changes

Posted by [Didier](#) on Fri, 13 Dec 2024 18:26:31 GMT

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

---

Hello Mirek,

I am trying to make my GraphCtrl re-style correctly: it almost worked out of the box :)  
Except for one thing : I don't draw everything on refresh, some of the drawing is cached for performance reasons. So i am trying to find the correct method to detect this FULL REPAINT event.

I Tried using IsFullRefresh() ==> didn't work

I tried overloading : void State(int reason)

```
virtual void State(int reason) {  
    if (reason == LAYOUTPOS) {  
        // do what I need  
    }  
}
```

==> this worked but is it the wright way ??

---

Subject: Re: Dynamic theme changes

Posted by [Tom1](#) on Fri, 13 Dec 2024 19:38:34 GMT

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

---

Hi Didier,

I guess you should implement Skin() for your control.

Best regards,

Tom

---

---

Subject: Re: Dynamic theme changes

Posted by [Didier](#) on Sat, 14 Dec 2024 08:31:55 GMT

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

---

Thankyou Tom,

I guess it was to obvious, so I missed it :)

---

---

Subject: Re: Dynamic theme changes

Posted by [Didier](#) on Wed, 18 Dec 2024 22:22:32 GMT

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

---

Just to share my little experience with dynamic theme changing.  
There are some traps (at least one in which I fell :) ):

If you use a Value to store a specific Style (to paint a rect for example): Value can store a Color, an Image, a function. Then it doesn't get modified when dynamic skin changes.

If you stored an Image inside the Value: things work out of the box :)

If you stored a Color inside the Value, the solution is simply to make an AColor instead of Color and it just works out of the box :)

Example:

```
struct MyCtrlStyle : ChStyle<MyCtrlStyle> {  
    Value    plotBckgndStyle;  
};
```

...

Replace:

```
plotBckgndStyle = Blend(SColorShadow(), SColorPaper(), 190);
```

with:

```
plotBckgndStyle = AColor( Blend(SColorShadow(), SColorPaper(), 190));
```