
Subject: Style for FileSel / FileList
Posted by [kohait00](#) on Wed, 07 Jul 2010 11:15:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

hi folks

FileSel / FileList is not Style compliant, so when changing the global params SColorFace(), SColorText() etc. at runtime, the first settings remain..

normally, after a SColor* change, i do

```
//for all relevant UI parts that depend on global GUI style i do, (here only TabBar shown)
{
  //TABBAR STYLE
  TabCtrl::Style& st = TabBar::StyleDefault().Write();
  st = st.Standard();
}
//after recalculating default style infos
RefreshLayoutDeep();
```

this works pretty fine for most controls,
well except for FileSel and FileList for now.

any idea how to change the visual formatting of FileList / FileSel on the fly?

Subject: Re: Style for FileSel / FileList
Posted by [mirek](#) on Wed, 07 Jul 2010 20:20:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

kohait00 wrote on Wed, 07 July 2010 07:15hi folks

FileSel / FileList is not Style compliant, so when changing the global params SColorFace(), SColorText() etc. at runtime, the first settings remain..

Actually, I have to admit the we gave up at that point. "Live" style changes would require too much work for too little practical benefit.

Subject: Re: Style for FileSel / FileList
Posted by [kohait00](#) on Wed, 07 Jul 2010 20:54:21 GMT

Quote:

"Live" style changes would require too much work for too little _practical_ benefit.

my app unfortunately needs to change the appearance at runtime. (day / night view, it's a audio control app, working both backstage with dark light and also outdoors..)

but the problem was solved when i inspected the sources and properly adjusted the

Quote:

```
SColorHighlightText_Write(White());
```

because FileList recalculates stuff based on a bool dark flag, then using highlight stuff..

nevertheless, the described method above works for me pretty fine.

together with setting global colors prior..i might post a topic on that later to indicate it complete.

the results are pretty nice though thanks to 'porting and overriding' the Ctrls.iml as well.
