Subject: FIX: proper use of Style in Ctrls from CtrlLib
Posted by kohait00 on Thu, 24 Jun 2010 10:38:34 GMT
View Forum Message <> Reply to Message

hi folks,

i stumbled over inconsistant use of style stuff inside some upp Ctrls, names EditCtrl, Calender, Clock, ButtonOption..

they should not have stuff like

Paint:
const Style &st = style ? *style : StyleDefault();

but should instead properly do what other Ctrl's do

ctor:
style = &StyleDefault();

//Paint and elsewhere
style->line...

this prevents things from breaking up when editing global style information (over StyleDefault)..

i attached a rar file with .svn inside to easy compare what i mean.

PS: neither should Font be part of Style or should it.. i havent found any other that Calender Ctrl to be the case... because SetFont should be used instead, right?

RFC

File Attachments

1) CtrlLib.rar, downloaded 398 times

Subject: Re: FIX: proper use of Style in Ctrls from CtrlLib
Posted by mirek on Fri, 25 Jun 2010 13:01:25 GMT
View Forum Message <> Reply to Message

Yes.

Mirek

Subject: Re: FIX: proper use of Style in Ctrls from CtrlLib
Posted by kohait00 on Fri, 25 Jun 2010 13:34:58 GMT
View Forum Message <> Reply to Message

hey mirek,

something went wrong i thing, in the last cmmit 2508
i even missed it myself...

```
EditField::EditField()
{
 dropcaret = Rect(0, 0, 0, 0);
 style = &StyleDefault(); // <---
 Unicode();
 Reset();
 WhenBar = THISBACK(StdBar);
}
```

and:

```
void EditField::Paint(Draw& w)
{
 Size sz = GetSize();
 const EditField::Style& st = StyleDefault(); //<--- this one prevents overriding style completely,
should be '*style;'
 bool enabled = IsShowEnabled();
```

and:
what about ButtonOption

Subject: Re: FIX: proper use of Style in Ctrls from CtrlLib
Posted by mirek on Sat, 26 Jun 2010 06:38:57 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Fri, 25 June 2010 09:34hey mirek,

something went wrong i thing, in the last cmmit 2508
i even missed it myself...

```
EditField::EditField()
{
 dropcaret = Rect(0, 0, 0, 0);
 style = &StyleDefault(); // <---
```

```
 Unicode();
 Reset();
 WhenBar = THISBACK(StdBar);
}
```

This one is in Reset... In fact, EditField was working properly, the only problem was redundant code in Paint.

```
void EditField::Paint(Draw& w)
{
 Size sz = GetSize();
 const EditField::Style& st = StyleDefault(); //<--- this one prevents overriding style completely, should be '*style;'
 bool enabled = IsShowEnabled();
```

Ops

Mirek