
Subject: Chameleon-ized EditField
Posted by [mrjt](#) on Fri, 20 Apr 2007 11:30:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

I would like EditFields to change color when they have focus (as they do when they have invalid data). However, because so many controls are derived from EditField it is unpleasant to modify the Paint method.

Instead I have given the control a simple Chameleon style that functions identically to the original but exposes a Color that can be set when it has focus. My version is attached.

Would it be possible to have something similar incorporated into the Upp::CtrlLib package so that I don't have to update the source myself?

File Attachments

1) [EditField.cpp](#), downloaded 702 times

Subject: Re: Chameleon-ized EditField
Posted by [mrjt](#) on Fri, 20 Apr 2007 11:30:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Header file.

File Attachments

1) [EditCtrl.h](#), downloaded 666 times

Subject: Re: Chameleon-ized EditField
Posted by [mirek](#) on Fri, 20 Apr 2007 13:19:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

OK. I did two changes:

Invalid color - moved that Blend to default style, so that in Paint no blending, just invalid is used.

Renamed hligght to select.

Mirek

Subject: Re: Chameleon-ized EditField
Posted by [mrjt](#) on Tue, 03 Jul 2007 16:41:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Currently disabled EditFields show pink if they have invalid data, which doesn't make sense to me, so I've added an extra check (and it now caches the result of `IsShowEnabled` since it's called 5 times):

```
void EditField::Paint(Draw& w)
{
    Size sz = GetSize();
    bool f = HasBorder();
    bool enabled = IsShowEnabled();
    const EditField::Style *st = style ? style : &StyleDefault();
    Color paper = enabled && !IsReadOnly() ? (HasFocus() ? st->focus : st->paper) : st->disabled;
    if(nobg)
        paper = Null;
    Color ink = enabled ? st->text : st->textdisabled;
    if(convert && enabled && convert->Scan(text).IsError())
        paper = st->invalid;
    int fcy = font.Info().GetHeight();
    int yy = GetTy();
    w.DrawRect(0, 0, 2, sz.cy, paper);
    w.DrawRect(0, 0, sz.cx, yy, paper);
    w.DrawRect(0, yy + fcy, sz.cx, sz.cy - yy - fcy, paper);
    w.DrawRect(sz.cx - 2, 0, 2, sz.cy, paper);
    w.Clipoff(2, yy, sz.cx - 4, fcy);
    int x = -sc;
    if(IsNull(text) && !IsNull(nulltext)) {
        const wchar *txt = nulltext;
        Paints(w, x, fcy, txt, nullink, paper, nulltext.GetLength(), false);
    }
    else {
        const wchar *txt = text;
        int l, h;
        if(GetSelection(l, h)) {
            Paints(w, x, fcy, txt, ink, paper, l, password);
            Paints(w, x, fcy, txt, enabled ? st->selectedtext : paper,
                    enabled ? st->selected : ink, h - l, password);
            Paints(w, x, fcy, txt, ink, paper, text.GetLength() - h, password);
        }
        else
            Paints(w, x, fcy, txt, ink, paper, text.GetLength(), password);
    }
    w.DrawRect(x, 0, 9999, fcy, paper);
    w.End();
    // w.DrawRect(dropcaret, LtBlue());
    DrawTiles(w, dropcaret, CtrlImg::checkers());
}
```

Could this be added to Uppsrc please?

James

Subject: Re: Chameleon-ized EditField
Posted by [mirek](#) on Tue, 03 Jul 2007 17:45:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Yes. The base code has changed meanwhile, but I hope I have preserved the spirit of your change;)

Mirek
