

---

Subject: Re: there is no fontpusher class (just checking)

Posted by [nixnixnix](#) on Mon, 06 Aug 2007 19:02:37 GMT

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

---

Thanks Fudadmin, for stopping me barking up the wrong tree. I went back to overriding the Paint routine and it works fine.

Cheers,

Nick

```
#define LAYOUTFILE <OpenWind/FontPusher.lay>
```

```
#include <CtrlCore/lay.h>
```

```
class FontDlg : public WithFontPopupLayout<TopWindow>
```

```
{  
    typedef FontDlg CLASSNAME;  
public:
```

```
    FontDlg();  
    ~FontDlg();
```

```
    void Set(Font font,Color col);  
    Font GetFont();  
    Color GetColor(){return color.GetData();}
```

```
protected:
```

```
    Font m_font;
```

```
    void OnChange();  
    void Set();
```

```
};
```

```
class FontPusher : public Ctrl
```

```
{  
    typedef FontPusher CLASSNAME;  
public:
```

```
    FontPusher();  
    virtual ~FontPusher();
```

```
    virtual void Paint(Draw& w);  
    virtual void LeftDown(Point p, dword);
```

```
    void Set(Font font,Color col){m_font = font; m_color = col; Refresh();};  
    Font GetFont(){return m_font;}
```

```
Color GetColor(){return m_color;}
```

```
protected:
```

```
Color m_color;
```

```
Font m_font;
```

```
FontDlg fonts;
```

```
String text;
```

```
void Drop();
```

```
};
```

```
#include "FontPusher.h"
```

```
FontDlg::FontDlg()
```

```
{
```

```
CtrlLayoutOKCancel(*this, "Choose Font");
```

```
size.Add(4);
```

```
size.Add(5);
```

```
size.Add(6);
```

```
size.Add(7);
```

```
size.Add(8);
```

```
size.Add(9);
```

```
size.Add(10);
```

```
size.Add(11);
```

```
size.Add(12);
```

```
size.Add(14);
```

```
size.Add(16);
```

```
size.Add(18);
```

```
size.Add(20);
```

```
size.Add(22);
```

```
size.Add(24);
```

```
size.Add(26);
```

```
size.Add(28);
```

```
size.Add(36);
```

```
size.Add(42);
```

```
size.Add(48);
```

```
size.Add(60);
```

```
size.Add(72);
```

```
for(int i=0;i<Font::GetFaceCount();i++)
```

```
{
```

```
face.Add(Font::GetFaceName(i));
```

```
}
```

```

face.WhenAction = THISBACK(OnChange);
strikeout.WhenAction = THISBACK(OnChange);
bold.WhenAction = THISBACK(OnChange);
underline.WhenAction = THISBACK(OnChange);
italic.WhenAction = THISBACK(OnChange);
size.WhenAction = THISBACK(OnChange);
color.WhenAction = THISBACK(OnChange);

}

void FontDlg::OnChange()
{
    GetFont();
    Set();

    Refresh();
}

FontDlg::~FontDlg()
{
}

void FontDlg::Set(Font font,Color col)
{
    m_font = font;
    color.SetData(col);

    Set();
}

void FontDlg::Set()
{
    size.SetValue(m_font.GetHeight());
    face.SetIndex(m_font.GetFace());

    strikeout = m_font.IsStrikeout();
    italic = m_font.IsItalic();
    bold = m_font.IsBold();
    underline = m_font.IsUnderline();

    preview.SetText(m_font.GetFaceName());
    preview.SetFont(m_font);
    preview.SetInk(color.GetData());

    Refresh();
}

```

```

Font FontDlg::GetFont()
{
    m_font =
Font().Face(face.GetIndex()).Height(size.GetData()).Strikeout(strikeout).Italic(italic).Bold(bold).Underline(underline);

    return m_font;
}

FontPusher::FontPusher()
{
    text = "Choose Font";
}

FontPusher::~FontPusher()
{
}

void FontPusher::LeftDown(Point p, dword)
{
    Drop();
}

void FontPusher::Drop()
{
    fonts.Set(m_font,m_color);
    if(fonts.Execute()!=IDOK)return;

    m_color = fonts.GetColor();
    m_font = fonts.GetFont();

    Refresh();
}

void FontPusher::Paint(Draw& w)
{
    Size sz = GetSize();
    // String txt = m_font.GetFaceName();
    Size tsz = GetTextSize(text, m_font);

    w.DrawRect(1, 1, sz.cx - 2, sz.cy - 2, SWhite);
    DrawFrame(w, 0, 0, sz.cx, sz.cy, SBlack);
    w.DrawText((sz.cx - tsz.cx) / 2, (sz.cy - tsz.cy) / 2, text, m_font, m_color);
}

```

```
LAYOUT(FontPopupLayout, 332, 228)
ITEM(Label, dv___0, SetLabel(t_("Font")).LeftPosZ(16, 52).TopPosZ(16, 20))
ITEM(DropList, face, LeftPosZ(72, 172).TopPosZ(16, 19))
ITEM(Option, italic, SetLabel(t_("Italic")).LeftPosZ(16, 48).TopPosZ(44, 18))
ITEM(Option, bold, SetLabel(t_("Bold")).LeftPosZ(16, 44).TopPosZ(68, 18))
ITEM(Option, underline, SetLabel(t_("Underline")).LeftPosZ(72, 68).TopPosZ(44, 18))
ITEM(Option, strikethrough, SetLabel(t_("Strikethrough")).LeftPosZ(72, 64).TopPosZ(68, 18))
ITEM(DropList, size, LeftPosZ(256, 64).TopPosZ(16, 19))
ITEM(Button, ok, SetLabel(t_("OK")).RightPosZ(10, 68).BottomPosZ(12, 20))
ITEM(Button, cancel, SetLabel(t_("Cancel")).HCenterPosZ(68, 44).BottomPosZ(12, 20))
ITEM(LabelBox, dv___10, SetLabel(t_("Preview")).LeftPosZ(16, 304).TopPosZ(96, 92))
ITEM(ColorPusher, color, LeftPosZ(232, 76).TopPosZ(60, 24))
ITEM(LabelBox, dv___12, SetLabel(t_("Color")).HSizePosZ(220, 12).TopPosZ(44, 48))
ITEM(Label, preview, LeftPosZ(28, 276).TopPosZ(112, 68))
END_LAYOUT
```

So its not a gold standard class - it could use a SetText function as well as a nulltext option which makes it default back to the Face name but hey, its a start for anyone out there who wants the equivalent of the MSWord fontpicker.