
Subject: Re: State of "button"

Posted by [cbpporter](#) on Thu, 24 Apr 2008 11:08:33 GMT

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

Here is a little annoyance: doing something like this switches from the OS skin to some other skin in ToolButton:

```
static ToolButton::Style s = ToolButton::StyleDefault();  
but.SetStyle(s);
```

This is actually due to my fix for skins outside a ToolBar, by adding this line to the constructor:

```
ToolButton::ToolButton()  
{  
    Reset();  
    checked = false;  
    paint_checked = false;  
    // next line  
    SetStyle(ToolBar::StyleDefault().buttonstyle);  
    Transparent();  
}
```

This was not the best fix, because it does not update the default style. My understanding of Chameleon is better now, and I think it would be better to update the style directly:

```
void ChHostSkin()  
{  
    ChSysInit();  
    .....  
  
    {  
        ToolBar::Style& s = ToolBar::StyleDefault().Write();  
        Win32Look(s.buttonstyle.look, 6, XP_TOOLBAR, 1, 1);  
        Win32Look(ToolBar::StyleDefault().Write().look, 6, XP_TOOLBAR, 1, 1);  
        Win32Look(s.arealook, XP_REBAR, 0, 1);  
        s.breaksep.l2 = SColorLight();  
    }  
}
```

This is also IMO better, since ToolButton has a style default, and thus should own it's style. buttonstyle from TollBar is still updated to maintain backward compatibility.

Also, would it be possible to add this style to ToolButton?

```
CH_STYLE(ToolButton, Style, StyleSolid)  
{  
    const Button::Style& bs = Button::StyleNormal();  
    look[0] = bs.look[0];  
    look[1] = bs.look[1];  
}
```

```
look[2] = bs.look[2];
look[3] = bs.look[3];
look[4] = bs.look[2];
look[5] = bs.look[1];
font = StdFont();
for(int i = 0; i < 4; i++)
    textcolor[i] = Button::StyleNormal().textcolor[i];
textcolor[CTRL_CHECKED] = textcolor[CTRL_NORMAL];
textcolor[CTRL_HOTCHECKED] = textcolor[CTRL_HOT];
for(int i = 0; i < 6; i++) {
    light[i] = false;
    contrast[i] = 0;
}
light[CTRL_PRESSED] = light[CTRL_HOT] = light[CTRL_HOTCHECKED] = true;
}
```
