## Subject: Global style changes using Chameleon...
Posted by tvanriper on Fri, 11 Apr 2008 14:48:31 GMT

View Forum Message <> Reply to Message

Introduction

I wanted my application to use custom graphics and colors for a lot of things, to include my buttons, text boxes, and tab headers.

I found it very tedious to do this for every single control in every single dialog in my application, though.  This gets extremely rough if your application has quite a few dialogs in it, and you want everything to look uniform.

So, I figured the whole point of Chameleon was to make this kind of thing easier, and I set about figuring out how to use it this way.  Documentation is scarce, since it's pretty new... but I can read source code well enough, so I thought I'd give it a try.

Hopefully, I can describe what I learned in some fashion, so others can more easily use this bit of technology.  If I have more time, and if nobody else beats me to it, I'll try my hand at writing a decent guide for the help topics later.

Using Chameleon for global style changes

If you're looking to use Chameleon for this sort of purpose, you need to remember the word 'Write'.  All of the Chameleon functions intended to help you modify the initial values tend to have the word 'Write' in it somewhere.

So, for example, if you want to change the color of the 'paper' attribute, so anything using SColorPaper will use your color instead of the default, you have to call SColorPaper_Write( [your color here] ).

You can even use LabelBoxTextColor_Write to help change the default color of your label box texts.

Here's a quick, but non-exhaustive, list of attributes whose colors you can change:


```
SColorPaper_Write
SColorText_Write
SColorHighlight_Write
SColorHighlightText_Write
SColorMenu_Write
SColorMenuText_Write
SColorDisabled_Write
SColorFace_Write
SColorShadow_Write
LabelBoxTextColor_Write
```

You may want to hunt around for others, as needed.

This said, sometimes you might want to modify the default graphics for a particular item... maybe you want a different look to your progress bar, or your buttons.

The best approach for this, I think, involves calling the Write() function for your particular style, and modifying the object directly.

Here's an example using the Write() function:

```
Button::Style& button_style = Button::StyleNormal().Write();
button_style.look[0] = MyButton::Get(0);
button_style.look[1] = MyButton::Get(1);
button_style.look[2] = MyButton::Get(2);
button_style.look[3] = MyButton::Get(3);
button_style.textcolor[0] = White;
button_style.textcolor[1] = White;
button_style.textcolor[2] = White;
button_style.textcolor[3] = Gray;
```

Edit:

Edited to reflect some of the information mentioned in this thread.