
Subject: Re: Manual for custom Chameleonized CtrlS

Posted by [mrjt](#) on Fri, 25 Jun 2010 10:31:10 GMT

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

In case it may be of use to other people here is example of how to modify a Chameleon style while retaining theming.

Before Chameleon ButtonOptions had a none themed look that I preferred, so when Chameleon was added I reproduced it:

(both buttons are pressed, I keep the OS look when not pressed)

This is the code that does it:

```
#include <CtrlLib/CtrlLib.h>
```

```
using namespace Upp;
```

```
#define IMAGECLASS Img
```

```
#define IMAGEFILE <LayoutTest/LayoutTest.iml>
```

```
#include <Draw/iml.h>
```

```
Image ApplyStyle(const Value &base, const Image &style)
```

```
{  
    Size sz(30, 30);  
    ImageDraw w(30, 30);  
    w.DrawRect(sz, SColorFace());  
    ChPaint(w, sz, base);  
    ChPaint(w, sz, style);
```

```
  
    Image img = w;  
    ImageBuffer ib(img);  
    ib.SetHotSpot(style.GetHotSpot());  
    ib.Set2ndSpot(style.Get2ndSpot());  
    return (Image)ib;  
}
```

```
void ButtonOption_HotPress(ButtonOption &bo)
```

```
{  
    static ButtonOption::Style style;  
    style = ButtonOption::StyleDefault();  
    style.look[2] = ApplyStyle(style.look[2], Img::HotPress());  
    bo.SetStyle(style);  
}
```

```
GUI_APP_MAIN
```

```
{  
    TopWindow wnd;  
    ButtonOption normal, hotpress;
```

```
wnd.SetRect(Size(140, 32));  
wnd.CenterScreen();  
  
ButtonOption_HotPress(hotpress);  
  
normal.SetLabel("Normal").LeftPos(4, 64).TopPos(4, 24);  
hotpress.SetLabel("HotPress").LeftPos(72, 64).TopPos(4, 24);  
wnd.Add(normal);  
wnd.Add(hotpress);  
  
wnd.Run();  
}
```

What it does is render the existing style value to a buffer (since it's not an Image - it's an os theme) and then draw my own image over the top.

File Attachments

1) [HotPressWnd.png](#), downloaded 1471 times
