
Subject: State of "button"

Posted by [cbporter](#) on Wed, 02 Apr 2008 15:32:10 GMT

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

OK, I've done it again! I had to write custom code to be able to do some very basic and common things with a button.

I'm a little unhappy with the state of button classes in U++. Button is by far the most advanced of them, but it does not feature a "down" state. A lot of my buttons need a down state. You can use ButtonOption, but that class is semi useless. Or you can use ToolButton, but that need a lot of visual tweaks to get it to behave nicely. Worst of all, if you change class, you have to modify a lot of code, since interfaces are different(one has Label, other has SetLabel).

So I propose a number of changes:

1. Uniform interface. Pretty self-explanatory.
2. Get rid of ButtonOption and add down state to Button (add typedef Button ButtonOption) or make ButtonOption inherit Button, add a label property, StyleDefault, etc.
3. Either add a StyleFlat to Button, or tweak ToolButton so that it can be used as a generic button.
4. Make styles compatible between button classes.

These are just some suggestions and the list is open, but before stable 2008 I think that it would be great if buttons wouldn't cause such headaches. I implemented partially on a by need bases a lot of these ideas, and would be relatively easy to gather them and implement this, but I need to know what you think.

Subject: Re: State of "button"

Posted by [cbporter](#) on Mon, 14 Apr 2008 22:25:18 GMT

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

Well, since no reply to my proposition, I think that it is safe to assume that people are happy with the state of "button", so can I go forward and implement a couple of custom button classes and start using them exclusively for my own needs?

Subject: Re: State of "button"

Posted by [mrjt](#) on Tue, 15 Apr 2008 08:48:40 GMT

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

Personally I agree with you, especially with regard to ButtonOption. But then it isn't my decision to make, and I'd be bit worried about backward compatibility.

Subject: Re: State of "button"

Posted by [mirek](#) on Wed, 16 Apr 2008 08:17:01 GMT

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

The trouble of ButtonOption visuals (which I do not really like too) is that needed bits are missing in theming APIs and I was not able to find a good heurestics to create them...

Other than that, I am not a big friend of "one fits all" button class. What is wrong with using specific classes for specific purposes?

Mirek

Subject: Re: State of "button"

Posted by [cbporter](#) on Wed, 16 Apr 2008 19:46:36 GMT

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

luzr wrote on Wed, 16 April 2008 11:17The trouble of ButtonOption visuals (which I do not really like too) is that needed bits are missing in theming APIs and I was not able to find a good heurestics to create them...

Other than that, I am not a big friend of "one fits all" button class. What is wrong with using specific classes for specific purposes?

Mirek

The problem with ButtonOption is not related to visual style, which on both XP and Vista look OK, but the lack of functionality.

I was not suggesting to make a "one fits all", but rather make other classes except Button viable. But anyway, Button still is a special class and could very well do with a "one fits all" solution, mainly because it's a button. It's not supposed to do anything fancy, just be able to respond to click, display text and optionally an Image.

So to sum of what I consider insufficient: Button has no down state. That means I have to use ButtonOption or ToolButton. ButtonOption has no label and also the interface could be much more similar to Button. IMO, It should be 100% the same, except the part related to checking and unchecking. ToolButton on the other hand has two problems. First, the visual are not good at custom sizes. Easy to fix. I proposed a fix a while ago and it was applied, and it works well, except when messing with StyleDefault. I have better fix since then. Second, interface, and especially style is not compatible, so there is a lot of extra code If you want it to not be flat. The same applies to other button classes if you want them to be flat.

Subject: Re: State of "button"

Posted by [cbporter](#) 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(ToolButton::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;
}
```

Subject: Re: State of "button"

Posted by [mirek](#) on Sat, 26 Apr 2008 07:59:08 GMT

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

OK, both patches applied. Thanks.

Mirek

Subject: Re: State of "button"

Posted by [cbporter](#) on Sat, 26 Apr 2008 15:00:01 GMT

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

Great! Thanks! This solves about 70% of my problems with buttons. All there is left is to improve the position of labels and image in ToolButton and I'm happy. I'll get to it after Easter. Then I'll use Button as a standard button and ToolButton as a flat button or an option button and can forget about OptionButton. But still cleaning OptionButton up is not a bad idea and maybe I can do that too if you wish.

Subject: Re: State of "button"

Posted by [cbporter](#) on Tue, 29 Apr 2008 08:17:40 GMT

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

I centered the image in text in ToolButton. I've done this as default for both modes that have label. I hope I did not upset some pixel perfect layout of these elements that you had in mind, but I think that they are set up the same in toolbars. Tested in UWord too and looks the same. I don't think that default center is a problem since no one would like a button like this:

Here is the code:

```
void ToolButton::Paint(Draw& w)
{
    LTIMING("ToolButton::Paint");
    paint_checked = checked;
```

```

Size sz = GetSize();
UPP::Image image = GetImage();
Size isz = image.GetSize();
// Ctrl *q = GetParent()->GetParent();
// if(!q || !q->IsTransparent())
// w.DrawRect(sz, checked && !HasMouse() ? Blend(SColourFace, SColourLight) : SColourFace);
int li = IsEnabled() ? HasMouse() ? GetMouseLeft() ? CTRL_PRESSED
                      : checked ? CTRL_HOTCHECKED : CTRL_HOT
                      : checked ? CTRL_CHECKED : CTRL_NORMAL
                      : CTRL_DISABLED;
ChPaint(w, sz, style->look[li]);
Point off = style->offset[li];
Point ip = (sz - isz) / 2 + off;
Size tsz;
if(kind != NOLABEL)
  tsz = GetTextSize(text, style->font);
if(kind == BOTTOMLABEL) {
  ip.y -= tsz.cy / 2 + 1;
  w.DrawText((sz.cx - tsz.cx) / 2 + off.x, ip.y + isz.cy + 2 + off.y, text, style->font,
  style->textcolor[li]);
}
if(kind == RIGHTLABEL) {
  ip.x -= tsz.cx / 2 + 2;
  w.DrawText(ip.x + isz.cx + 3 + off.x, (sz.cy - tsz.cy) / 2 + off.y, text, style->font,
  style->textcolor[li]);
}
UPP::Image img = CachedContrast(image, style->contrast[li]);
if(!IsEnabled())
  img = DisabledImage(img);
if(IsEnabled() && style->light[li])
  DrawHighlightImage(w, ip.x, ip.y, img, true);
else
  w.DrawImage(ip.x, ip.y, img);
}

```

Also, some extra stuff for checking:

```

Bar::Item& ToolButton::Check(bool check)
{
  checked = check;
  Refresh();
  return *this;
}

```

ToolButton& Label(const char *text, int kind = ToolButton::RIGHTLABEL);

bool IsChecked() { return checked; } // in header

I didn't use bool Get() since there is Radio too, and it would be ambiguous: does get return checked state or radio state? Right now they are the same, but this could change.

This about covers all there is to make ToolButton a viable ButtonOption replacement. It does not

look that great checked with solid skin on Vista, but neither does ButtonOption, so I guess it's ok.

File Attachments

1) [Button.png](#), downloaded 989 times

Subject: Re: State of "button"

Posted by [mirek](#) on Tue, 29 Apr 2008 17:30:21 GMT

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

Thank you, applied.

Mirek

Subject: Re: State of "button"

Posted by [cbporter](#) on Sat, 14 Mar 2009 09:33:41 GMT

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

While doing the my documentation rounds of CtrlLib I have intentionally skipped ButtonOption. It is still as useless as ever. I use normal Buttons and ToolButtons, but there are still two things that ButtonOption can do which the others can't (automatic state switch on click and down state image).

So I propose that I add a Label property to ButtonOption, and a textcolor to the style.

Or a second solution would be to make it inherit from Button. That is a little complicated for such a niche class.

Which would you prefer?

Subject: Re: State of "button"

Posted by [mirek](#) on Mon, 16 Mar 2009 09:31:40 GMT

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

Label property and textcolor is OK with me.

Mirek

Subject: Re: State of "button"

Posted by [mrjt](#) on Mon, 16 Mar 2009 16:26:12 GMT

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

Yes please!

Subject: Re: State of "button"

Posted by [cbporter](#) on Tue, 17 Mar 2009 17:37:56 GMT

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

Here are the patch files.

I tested under XP and Vista, and also did a branch of an app and replaced all ToolButtons with ButtonOptions so I think it is OK now.

One thing that I would like to change is the way the control behaves as if in state CTRL_HOT when parent control automatically focuses it or when navigating with tab key. I don't know how to achieve this. It doesn't have to be part of the control, but it would really help me.

File Attachments

1) [Button.patch](#), downloaded 359 times

2) [PushCtrl.patch](#), downloaded 474 times

Subject: Re: State of "button"

Posted by [mirek](#) on Thu, 19 Mar 2009 17:29:29 GMT

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

Can you post full files, please?

Alternatively, I would like to add .patch and .diff support to theide (it is quite ready for such thing), but I am now too lazy to implement the algorithm...

Could anybody contribute the code that combines (or tries to combine) original file and patch/diff and produces altered file?

Mirek

Subject: Re: State of "button"

Posted by [cbporter](#) on Thu, 19 Mar 2009 18:03:21 GMT

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

Sure!

File Attachments

1) [PushCtrl.h](#), downloaded 297 times

2) [Button.cpp](#), downloaded 296 times

Subject: Re: State of "button"

Posted by [mirek](#) on Thu, 19 Mar 2009 18:28:54 GMT

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

luzr wrote on Thu, 19 March 2009 13:29

Alternatively, I would like to add .patch and .diff support to theide (it is quite ready for such thing), but I am now too lazy to implement the algorithm...

Could anybody contribute the code that combines (or tries to combine) original file and patch/diff and produces altered file?

Mirek

OK, take .diff out of list, TheIDE now can apply .diff (at least I hope).

Mirek

Subject: Re: State of "button"

Posted by [mirek](#) on Thu, 19 Mar 2009 18:32:59 GMT

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

cbporter wrote on Thu, 19 March 2009 14:03Sure!

BTW, what have you used to create these .patch files?

Somehow I cannot find a correct specs, although it seems pretty simple format...

Mirek

Subject: Re: State of "button"

Posted by [cbporter](#) on Thu, 19 Mar 2009 18:36:32 GMT

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

luzr wrote on Thu, 19 March 2009 20:32cbporter wrote on Thu, 19 March 2009 14:03Sure!

BTW, what have you used to create these .patch files?

Somehow I cannot find a correct specs, although it seems pretty simple format...

Mirek

I used WinMerge integrated into TortoiseSVN. Generate Patch has 3 modes: Normal, Context and

Unified. I used normal.

Subject: Re: State of "button"

Posted by [cbporter](#) on Fri, 20 Mar 2009 15:42:26 GMT

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

I see you applied the patches. Well if it's final, I'm quite happy with both ToolButton and ButtonOption. Future additions only on a by need basis. Still need to investigate that tab key behavior, but I'll ignore it for now.

cbporter Issue #004: ButtonOption is useless

Fixed

Subject: Re: State of "button"

Posted by [mirek](#) on Fri, 20 Mar 2009 20:02:04 GMT

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

cbporter wrote on Fri, 20 March 2009 11:42 I see you applied the patches. Well if it's final, I'm quite happy with both ToolButton and ButtonOption. Future additions only on a by need basis. Still need to investigate that tab key behavior, but I'll ignore it for now.

cbporter Issue #004: ButtonOption is useless

Fixed

Well, I have also used it as excuse to develop patching for theide, and then I could check the result with full files too

Mirek
