

---

Subject: Dynamic Ctrl frame

Posted by [koldo](#) on Sun, 26 Apr 2020 21:56:08 GMT

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

---

I would like a coloured frame to appear around a control when the mouse passes over it, like the image:

I have tried it in two ways:

- Playing with CtrlFrame. Unfortunately the colors are set as const, so they cannot be changed using MouseEnter()
- Working with another frame control. Unfortunately I do not know how a child control can adapt to parent, but staying around it, not inside it (as with SizePos()). I would prefer that this child frame control would be created automatically in the parent field construction.

Thank you.

---

### File Attachments

1) [Sin título.png](#) , downloaded 533 times

---

---

Subject: Re: Dynamic Ctrl frame

Posted by [Didier](#) on Mon, 27 Apr 2020 11:48:29 GMT

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

---

Hello Koldo,

I've been playing around with Chameleon lately and  
I think you need to look at the EditField::Style.edge[]

```
struct Style : ChStyle<Style> {  
    Color paper;  
    Color disabled;  
    Color focus;  
    Color invalid;  
    Color text, textdisabled;  
    Color selected, selectedtext;  
    Color selected0, selectedtext0; // If does not have focus...  
    [b]Value edge[4];[/b] // border (for various active edge states)  
    Value coloredge; // border mask for adding color, e.g. round borders with red Error  
    bool activeedge;  
    int vfm;  
};
```

and using:

```
enum {  
    CTRL_NORMAL, CTRL_HOT, CTRL_PRESSED, CTRL_DISABLED,  
    CTRL_CHECKED, CTRL_HOTCHECKED  
};
```

So if you set `edge[CTRL_HOT]` to the right style image, you will probably get what you want

If you want to apply it to all `EditField` children, add at application init:  
`EditField::StyleDefault.Write().edge[CTRL_HOT] = YoureStyle;`

Note: this will only work for all `EditField` children ... not all `Ctrls`

If you want to apply it to a specific `Ctrl` (not all), this should do the trick :

Quote:

```
static EditField::Style myCtrlStyle = EditField::StyleDefault();
myCtrlStyle.edge[CTRL_HOT] = YoureStyle;
...
myCtrl.SetStyle(myCtrlStyle);
```

You may need to replace '`EditField::Style`' by '`<EditFieldChildType>::Style`' (depending on you're `EditXxxx Ctrl`)

---

---

Subject: Re: Dynamic Ctrl frame  
Posted by [koldo](#) on Mon, 27 Apr 2020 12:31:33 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Thank you Didier

The idea was to apply it to a set of `Ctrl`, and that when the mouse is out of the `Ctrl`, it would remain selected until other `Ctrl` of the set is chosen.

It would be like a Switch (from a set of `Ctrl`, you choose just one), but with a visual signal around any `Ctrl` of the set, and that is activated just by hovering on the `Ctrl` instead of by clicking it.

---

---

Subject: Re: Dynamic Ctrl frame  
Posted by [koldo](#) on Tue, 28 Apr 2020 13:51:54 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Problem solved.

Quote:- Playing with `CtrlFrame`. Unfortunately the colours are set as `const`, so they cannot be changed using `MouseEnter()`. Just creating a class derived from `CtrlFrame` but using dynamic coloring, instead of `const`, solved the issue. :)

Thank you!

---

---

Subject: Re: Dynamic Ctrl frame  
Posted by [Didier](#) on Tue, 28 Apr 2020 18:24:09 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hello Koldo,

I completely mis-understood you're needs, but you found the solution :)

---