
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] = YourStyle;
```

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] = YourStyle;
```

```
...
```

```
myCtrl.SetStyle(myCtrlStyle);
```

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