
Subject: Re: Proposal: add IsReadOnly() handling inside ButtonOption

Posted by [dolik.rce](#) on Sat, 01 May 2010 16:01:26 GMT

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

Mindtraveller wrote on Sat, 01 May 2010 16:07I've made a number of attempts to make "chosen" button option irresponsible to user clicks and failed. Digging into code revealed that there is no way to make it.

Hi Pavel!

There is a way Maybe not as straightforward as your patch, but it works fine too. The trick is to check the value of ButtonOption in WhenAction callback and revert the action whenever user tries to "unclick" the button:

```
#include "CtrlLib/CtrlLib.h"
```

```
using namespace Upp;
```

```
struct App : TopWindow {
    typedef App CLASSNAME;
    Array<ButtonOption> btns;
    void Check(int n){
        if (~btns[n]==0) btns[n].Set(true);
        else for(int i = 0; i < btns.GetCount(); i++){
            if(i!=n) btns[i].Set(false);
        }
    }
    App(){
        btns.Add().SetLabel("Click me! :-)").SetRect(10,10,200,50);
        btns.Add().SetLabel("No, click me! (-:").SetRect(10,70,200,50);
        for(int i = 0; i < btns.GetCount(); i++){
            btns[i]<=<THISBACK1(Check,i);
            Add(btns[i]);
        }
    }
};
```

```
GUI_APP_MAIN{
    App().Run();
}
```