
Subject: RadioButton

Posted by [andrei_natanael](#) on Thu, 18 Jun 2009 20:13:00 GMT

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

Hi, today I was working at one of my programs and I was upset because Switch control doesn't set my preferred position and size for his items(because of his design?) so I've cooked a new Switch using a classic approach(RadioButton and RadioGroup). Here I'm attaching the sources with hope these will be useful for others who encounter the same problem. I don't know if it's the right approach but it solve my problem.

```
// RadioButton.h
#ifndef _RadioButton_h_
#define _RadioButton_h_

#include <CtrlLib/CtrlLib.h>
namespace Upp
{
class RadioButton : public Option
{
    typedef RadioButton CLASSNAME;
public:
    RadioButton();
    Callback WhenAction;
    Callback WhenPush;
private:
    Callback1<RadioButton*> WhenActionGroupNotify;
    void ActionEvent();
    void PushEvent();
    friend class RadioGroup;
};

class RadioGroup
{
    typedef RadioGroup CLASSNAME;
public:
    RadioGroup& Add(RadioButton*);
    RadioGroup& Remove(RadioButton*);
    void Disable();
    void Enable();
    RadioButton* Get();
    operator int();
private:
    Vector<RadioButton*> group;
    void ActionEvent(RadioButton*);
};

}
#endif
```

```

// RadioButton.cpp
#include "RadioButton.h"

using namespace Upp;

RadioButton::RadioButton(): Option()
{
    Pusher::WhenAction = THISBACK(ActionEvent);
    Pusher::WhenPush = THISBACK(PushEvent);
    SwitchImage();
}

void RadioButton::ActionEvent()
{
    Set(1);
    WhenActionGroupNotify(this);
    WhenAction();
}

void RadioButton::PushEvent()
{
    WhenPush();
}

RadioGroup& RadioGroup::Add(RadioButton* b)
{
    b->WhenActionGroupNotify = THISBACK(ActionEvent);
    group.Add(b);
    return *this;
}

RadioGroup& RadioGroup::Remove(RadioButton* b)
{
    for (int i = 0; i < group.GetCount(); i++)
        if (group[i] == b) {
            b->WhenActionGroupNotify.Clear();
            group.Remove(i);
            break;
        }
    return *this;
}

void RadioGroup::ActionEvent(RadioButton* b)
{
    for (int i = 0; i < group.GetCount(); i++)
        if (group[i] != b)
            group[i]->Set(0);
}

```

```
void RadioGroup::Disable()
{
    for (int i = 0; i < group.GetCount(); i++)
        group[i]->Disable();
}
```

```
void RadioGroup::Enable()
{
    for (int i = 0; i < group.GetCount(); i++)
        group[i]->Enable();
}
```

```
RadioButton* RadioGroup::Get()
{
    for (int i = 0; i < group.GetCount(); i++)
        if (group[i]->Get())
            return group[i];
    return 0;
}
```

```
RadioGroup::operator int()
{
    for (int i = 0; i < group.GetCount(); i++)
        if (group[i]->Get())
            return i;
    return Null;
}
```

And a small example:

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

```
using namespace Upp;
```

```
void RadioSelected(int x)
{
    Exclamation("WhenAction: You selected Button " + AsString(x));
}
```

```
GUI_APP_MAIN
```

```
{
    TopWindow win;
    win.SetRect(0, 0, 200, 200);
    RadioButton r1, r2, r3, r4;
    RadioGroup group;
    win.Add(r1.SetLabel("RadioButton 1").HSizePos(10, 10).TopPos(10, 25));
```

```
win.Add(r2.SetLabel("RadioButton 2").HSizePos(20, 10).TopPos(40, 25));
win.Add(r3.SetLabel("RadioButton 3").HSizePos(30, 10).TopPos(70, 25));
win.Add(r4.SetLabel("RadioButton 4").HSizePos(20, 10).TopPos(100, 25));
r4.WhenAction = callback1(RadioSelected, 4);
group.Add(&r1).Add(&r2).Add(&r3).Add(&r4);
win.Run();
group.Disable();
win.Run();
group.Enable();
win.Run();
group.Remove(&r3); r3.Hide();
win.Run();
switch(group) {
case 0:
    Exclamation("RadioGroup index start with 0...");
    break;
case 1:
    Exclamation("Button 2");
    break;
case 2:
case 3:
    PromptOK("You selected button number 3 or 4");
    break;
default:
    PromptOK("No button selected");
}
PromptOK("That's my button ;) " + (group.Get()? group.Get()->GetLabel(): "NO BUTTON"));
}
```
