
Subject: Re: Proposed change to U++ to allow owning children.

Posted by [Lance](#) on Wed, 20 Apr 2011 19:23:36 GMT

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

Out put of sample program:

And the main content of the sample program:

```
using namespace Upp;
```

```
class MyLabel : public Label
{
public:
    MyLabel(String& s){ SetLabel(s); }
};
```

```
struct ButtonApp : TopWindow {
```

```
    ButtonApp() : count(0)
    {
        Sizeable().Zoomable();
        Click();
    }
```

```
    void Click()
    {
        int x, y;
        Ctrl * p;
        x=count%max_row_button*(button_width+5)+2;
        y=count/max_row_button*(button_height+5)+2;
```

```
        if( count & 1)
            p=&NewChild<MyLabel>(String().Cat()<<"["<<count<<" ] A Label");
        else
        {
            p=&NewChild<Button>()
                .SetLabel(String().Cat()<<"["<<count<<" ] Click Me");
            *(Button*)p<<=THISBACK(Click);
        }
```

```
        p->LeftPos(x,button_width).TopPos(y,button_height);
```

```
        ++count;
    }
```

```
typedef ButtonApp CLASSNAME;  
  
private:  
const static int button_width=180;  
const static int button_height=40;  
const static int max_row_button=5;  
  
int count;  
};
```

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

File Attachments

1) [DYNAMIC CHILD.PNG](#), downloaded 951 times
