
Subject: easiest way to convert a Topwindow into a control

Posted by [yeus](#) on Sat, 28 Oct 2006 18:06:34 GMT

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

Hi everyone, in this nice forum, desperately defending "real" c++

I started programming an application. it has its own layout and everything. After a while (now) it turned out that this window would also make a very nice control element for a window of another application I'm, working on.

I played with adding frames and stuff, and inheriting my own topwindow class from frames and vice versa, to add my window into the layout editor as "user class". But nothing seemed to work. Now I want to know: what's the easiest way to turn my window into a control without having to copy&paste code fragments into a new class and completely rewriting paint-events and similar things?

How does a user-class have to be declared, for being accepted as user class in the layout designer resource file?

(btw another thing: what about "spin-off" products of my applications? I have written several control classes, which are kind of practicable. Is there anyone who I can give those to? Who can check the code and decides, whether to add them to ultimate++ or not?)

Greetings, Tom

Subject: Re: easiest way to convert a Topwindow into a control

Posted by [fallingdutch](#) on Sat, 28 Oct 2006 18:35:30 GMT

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

Hi Tom,

as far as I know it just needs to be derived from class Ctrl and, if you want your class to be able to act as a Frame, FrameCtrl.

so change your class myNiceClass : public TopWindow
to class myNiceClass : public FrameCtrl<Ctrl> //both: Ctrl and Frame
class myNiceClass: public Ctrl //only Ctrl, not acting as a Frame

but then GUI_APP_RUN won't work so you have to add your Ctrl to your new TopWindow with
AddFrame(instanceOfMyNiceClass); //for the Frame version
Add(instanceOfMyNiceClass); //both only Ctrl and "Ctrl and Frame"

yeus wrote on Sat, 28 October 2006 20:06 How does a user-class have to be declared, for being accepted as user class in the layout designer resource file?

you need a description of your class in an .usc file ... but I don't know how - if you know it tell me, please

Bas

Subject: Re: easiest way to convert a Topwindow into a control
Posted by [fallingdutch](#) on Sat, 28 Oct 2006 19:28:06 GMT

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

i just reread some usc files and wrote a simple example:

ExampleCtrl.usc

```
ctrl ExampleCtrl {  
    group "Complex"; //will add this Ctrl to the Complex group  
    GetMinSize() {return Size(100,50);};  
    GetStdSize() {return Size(400,150);};  
  
    bool privateVar = true; // so you can configure your private variables in the Layout Designer  
  
    Paint(w) {  
        r = GetRect();  
        //have a look at CtrlLib.usc for many usefull functions  
        //but as far as i have read you can "only" draw rects, images and text  
  
        //these three lines draw a white Ctrl with a black thin border  
        w.DrawRect(r,:SBlack);  
        DeflateRect(r); //makes the rect smaller by one on each side  
        w.DrawRect(r,:SWhite);  
    }  
}
```

Bas

Subject: Re: easiest way to convert a Topwindow into a control
Posted by [mirek](#) on Sun, 29 Oct 2006 03:22:28 GMT

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

yeus wrote on Sat, 28 October 2006 14:06

After a while (now) it turned out that this window would also make a very nice control element for a window of another application I'm, working on.

Depends on how much clean solution you need.

First of all, despite being derived from TopWindow, you can still use such class as child Ctrl.

OTOH, usually it is very simple to combine child Ctrl with TopWindow (MyCtrl c; TopWindow win; win.Add(c.SizePos())).

Sometimes an interesting solution is to add TopWindow as Ctrl's option - at first, it might look a little bit strange (because C++ and GUI containment are reversed):

```
class MyCtrl {  
...  
    TopWindow win;  
  
    void OpenAsWindow() {  
        win.Add(this->SizePos());  
        win.Open();  
    }  
};
```

Mirek

Subject: Re: easiest way to convert a Topwindow into a control

Posted by [mirek](#) on Sun, 29 Oct 2006 03:27:53 GMT

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

yeus wrote on Sat, 28 October 2006 14:06

How does a user-class have to be declared, for being accepted as user class in the layout designer resource file?

Well, depends on desired level of integration.

Usually I just do nothing (I am lazy person) and use "User class", then fill in the name of class, being happy with empty rectangle to visualize the widget.

If you want higher level, you need .usc script.

Quote:

(btw another thing: what about "spin-off" products of my applications? I have written several control classes, which are kind of practicable. Is there anyone who I can give those to? Who can check the code and decides, whether to add them to ultimate++ or not?)

For now I recommend "useful code snippets" forum, or development forums.

There also seem to be suggestions for "U++ extensions" separate sourceforge project. Perhaps a nice idea, I see that at the moment a lot of useful is happening, but core U++ team is quite limited in resources (-> not much time to check).

Mirek

Mirek

Mirek

Subject: Re: easiest way to convert a Topwindow into a control
Posted by [yeus](#) on Sun, 29 Oct 2006 21:08:03 GMT

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

yeah.. thx for the quick replies... But my Problem differs a little bit from what you thought.

I have uploaded an exmapl project with the same error messages as my own object provides.

basically what i'm trying to achieve is: a class which I decalred as:

```
class testcontrol : public Withestcontrol<TopWindow>
```

and now I want to turn it into:

```
class testcontrol : public FrameCtrl<Ctrl>
```

something like that.. well In fact All I want to do is using the layout-file without having to write a *.usc file...

File Attachments

1) [frametester.zip](#), downloaded 1698 times

Subject: Re: easiest way to convert a Topwindow into a control
Posted by [mirek](#) on Mon, 30 Oct 2006 00:18:57 GMT

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

Layout is not required to be based on TopWindow.

In the same time, FrameCtrl still can be used as Ctrl.

So you can have e.g.

```
struct Foo : WithMyLayout<Ctrl>
```

struct Foo : FrameCtrl< WithMyLayout<TopWindow> >

or anything like that...

Mirek

P.S.: I am not quite sure how that relates to layouts and *.usc.

Subject: Re: easiest way to convert a Topwindow into a control

Posted by [yeus](#) on Mon, 30 Oct 2006 09:23:25 GMT

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

yay ! thx... that helped me a lot...

I also found out another thing:

it is very important to include the .h-file of your class BEFORE you define LAYOUTFILE:

```
#include "mylayout.h"
```

```
#define LAYOUTFILE <test5/test5.lay>
```

Otherwise it won't compile... jesus it took me about 2 days to find that out ...

Now I only got one Problem left:

the userclass thing with layouts and all that stuff works now...

My Window doesn't paint any background. Is there a method to paint the background automatically without drawing a rectangle in the "Paint" function?

greetings, Tom

greetings Tom...

Subject: Re: easiest way to convert a Topwindow into a control

Posted by [mirek](#) on Mon, 30 Oct 2006 09:32:59 GMT

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

yeus wrote on Mon, 30 October 2006 04:23yay ! thx... that helped me a lot...

I also found out another thing:

it is very important to include the .h-file of your class BEFORE you define LAYOUTFILE:

```
#include "mylayout.h"  
  
#define LAYOUTFILE <test5/test5.lay>
```

Otherwise it won't compile... jesus it took me about 2 days to find that out ...

Yes. That is because .lay uses real classes.

Anyway, you can have in layout even class that is not defined yet. Just leave the classname field empty and add it as member into your class.

Quote:

My Window doesn't paint any background. Is there a method to paint the background automatically without drawing a rectangle in the "Paint" function?

No. Use DrawRect(GetSize(), White()). (Note that setting color of background would be done in single line - just as is drawing actual background in Draw).

Mirek
