
Subject: Re: Modal vs non-modal window
Posted by [NeonN](#) on Thu, 04 Oct 2007 18:08:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ok, I think i got what i needed ... I thought out the mrjt's post again and used the THISBACK system of dealing with the OK/Cancel buttons instead of CtrlLayoutOKCancel().

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

using namespace Upp;

#define LAYOUTFILE <WindowTest/WindowTest.lay>
#include <CtrlCore/lay.h>

class WindowTest : public WithLayout1<TopWindow> {
private:
    WithLayout2<TopWindow> * wnd;
    void onButton1Click();
    void onExitWindow2();
    void DealWithWindow2();

public:
    typedef WindowTest CLASSNAME;
    WindowTest();
};

void WindowTest::onButton1Click()
{
    wnd = new WithLayout2<TopWindow>;
    CtrlLayout(*wnd, "Title2");
    //wnd->Open();
    //wnd->Open(this);
    wnd->OpenMain();
    this->Disable();
    wnd->WhenClose = wnd->ok <=< THISBACK(DealWithWindow2);
}

void WindowTest::DealWithWindow2()
{
    this->Enable();
    delete wnd;
}

WindowTest::WindowTest()
{
    CtrlLayout(*this, "Title1");
    Button1 <=< THISBACK(onButton1Click);
```

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

It's working but I don't know if it is ok... or if there is an easier/better option?

Next question is the pros and cons of having the new dialog window in a separate class (I am talking about window with about 10-20 edits or other components to fill + saving it all to DB).

I would be glad of any response ... and sorry for stupid questions . I am porting one app from PHP and I am not used to use classes or any GUI objects...
