
Subject: Re: Modal vs non-modal window
Posted by [NeonN](#) on Thu, 04 Oct 2007 16:59:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks for response...

I don't know if it is better to put the window in separate class or just in the existing one of the main window. I thought there is an option how to do everything (open window, check values on OK button click...) in a single class member. No other callbacks... one dialog in just one function. That is why I was trying to use `CtrlLayoutOKCancel` and:

```
*sent = false;
while (!(*sent))
{
    switch (layAddProject->Execute())
    {
        case IDOK:
        ...
        ...
        ...
```

And as I am thinking about it I guess it would be better to have some bigger or more important windows in a separate class. Could you pls tell me some pros and cons?

I was playing with it today again and can't find any difference between `Open()`, `Open(this)` and `OpenMain()` (except `OpenMain()` makes the new window visible on taskbar). None of them block the main window behind...

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

using namespace Upp;

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

```
class WindowTest : public WithLayout1<TopWindow> {
private:
    void onButton1Click();

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

void WindowTest::onButton1Click()
{
```

```
WithLayout2<TopWindow> * wnd = new WithLayout2<TopWindow>;
CtrlLayoutOKCancel(*wnd, "Title2");
//wnd->Open();
//wnd->Open(this);
wnd->OpenMain();
}
```

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

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

EDIT:

I just find out that new windows opened inside the app with OpenMain() can overlay the main window... but windows opened with Open() can't... but no option disable the main windows. But when I Execute() or Run() the window, the main window is disabled always ...
What am I doing wrong? Or am I so stupid?
