
Subject: Re: How to combine two widget-class in the topwindow

Posted by [fudadmin](#) on Sat, 01 Apr 2006 12:31:14 GMT

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

Here we go:

```
#include <CtrlLib/CtrlLib.h>
//tabs example- not very correct but might be useful for learning...

//personalized tab (1 child) widget
class LuigiTab : public ParentCtrl {
    Button btn;
    DocEdit doc;
public:

    typedef LuigiTab CLASSNAME;
    LuigiTab();
    ~LuigiTab(){};
};

LuigiTab::LuigiTab() {
    btn.SetLabel("Just Button");
    btn.SetRect(20,20,100,25);
    Add(btn);
    //I strongly recommend to learn positioning from Designer- use Ctrl-T!
    doc.LeftPosZ(30, 500).TopPosZ(50, 300);
    Add(doc);
}
```

```
//personalized tabs (many -container) widget
class LuigiTabs : public TabCtrl {
    LuigiTab tab1,tab2,tab3;
public:

    typedef LuigiTabs CLASSNAME;
    LuigiTabs();
    ~LuigiTabs(){};
};

LuigiTabs::LuigiTabs() {
    tab1.SizePos();
    tab2.SizePos();
    tab3.SizePos();
}
```

```

Add(tab1,"one");
Add(tab2,"two");
Add(tab3,"three");
}

```

//you can keep this (and more like this) class declarations in one yourApp.h file

```

class App : public TopWindow {
private:
    bool numbers_enabled;
    //prepare to add your widgets by declaring below...
    MenuBar    menu;
    LuigiTabs  tabs; //inserted declaration of personalized... - see class
    StatusBar  status;
    InfoCtrl   info1;
public:
    void Exit() {Close();}

    void EnableNumbers()    {numbers_enabled = !numbers_enabled;}
    void ShowNumber(int n)  {PromptOK(AsString(n));}
    void TestEdit()         {PromptOK("fromEdit");} //added No3 - your func
    void Dummy()            {PromptOK("your function needs to be implemented!");}
    //useful to have one dummy function while work in progress...
    //...to fill calling "gaps"
    //add here more funcs to call from menus,,,

    //bar declarations
    void SubSubMenu(Bar& bar);
    void FileMenu(Bar& bar);
    void EditMenu(Bar& bar);
    void HelpMenu(Bar& bar);
    void TopMenus(Bar& bar);

```

```

typedef App CLASSNAME;

```

```

App();
};

```

//while all implementations below you can keep in one or more *.cpp files (split by classes)

```

void App::SubSubMenu(Bar& bar) {
    for(int i = 0; i < 10; i++)
        bar.Add(AsString(i), THISBACK1(ShowNumber, i));
}

```

```

void App::FileMenu(Bar& bar) { //confusing name changed from "Menu" to "File"...
    bar.Add("Enable numbers", THISBACK(EnableNumbers)).Check(numbers_enabled);
}

```

```

bar.Add(numbers_enabled, "Numbers", THISBACK(SubSubMenu));
bar.Add("Exit", THISBACK(Exit)).Key(K_CTRL_E);
}

```

```

//added No2 - your extra menu item and subItems
void App::EditMenu(Bar& bar) {
bar.Add("EditMy_1", THISBACK(TestEdit)).Key(K_CTRL_D);
bar.Add("Copy", THISBACK(Dummy)).Key(K_CTRL_C);
bar.Add("Cut", THISBACK(Dummy)).Key(K_CTRL_X);
bar.Add("Paste", THISBACK(Dummy)).Key(K_CTRL_V);
//add more items here...
}

```

```

void App::HelpMenu(Bar& bar)
{
bar.Add("Help", THISBACK(Dummy)).Key(K_F1);
bar.Add("About...", THISBACK(Dummy));
}

```

```

void App::TopMenus(Bar& bar){ //row of top menu "labels" - what you see without opening
menus...
bar.Add("File", THISBACK(FileMenu));
bar.Add("Edit", THISBACK(EditMenu)); //added Step1 - extra menu item on Main & callback
bar.Add("Help", THISBACK(HelpMenu));
}

```

```

App::App()
{
//initialize your members on App creation (constructor)
numbers_enabled = true;
AddFrame(menu);
menu.Set(THISBACK(TopMenus));

```

```

//use Add not AddFrame -in most cases. use Assistant to see the return values
Add(tabs.SizePos());

```

```

//status stuff
status.AddFrame(info1.Width(250));
AddFrame(status.Height(25));
info1="info1: Welcome to the Ultimate++ !";
}

```

```

GUI_APP_MAIN
{

```

```
App().Title("Forlano Menu, Tabs and Status -v1").Zoomable().Sizeable().Run();  
}
```

Luigi, please study this example and ask questions immediately. Don't waste your time struggling alone!...
