
Subject: Re: Window without title bar

Posted by [Werner](#) on Thu, 14 Feb 2008 21:06:31 GMT

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

Ok. Just to reconcile those singleton-haters I have created a tiny demo, featuring multiple main windows and multiple subwindows (at most 1 subwindow in 1 main window lest Windows (!) crashes) with slide effect and fade effect - it goes without saying: WITHOUT using "Single", to say nothing of "singleton".

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

```
using namespace Upp;
```

```
class PopUpWin : public LineEdit
```

```
{
```

```
private:
```

```
    bool poppedUp_;
```

```
    void RightDown(Point p, dword keyFlags)
```

```
    {
```

```
        Close();
```

```
        poppedUp_ = false;
```

```
    }
```

```
public:
```

```
    PopUpWin() : poppedUp_(false) { }
```

```
    void SetPoppedUp(bool yesNo) { poppedUp_ = yesNo; }
```

```
    bool IsPoppedUp() const { return poppedUp_; }
```

```
};
```

```
class MainWin : public TopWindow
```

```
{
```

```
private:
```

```
    typedef MainWin CLASSNAME;
```

```
    static int mainWinCount;
```

```
    PopUpWin popUp;
```

```
    virtual void Paint(Draw& w)
```

```
    {
```

```
        w.DrawRect(GetSize(), WhiteGray);
```

```
        w.DrawText(0, 0, "Click left for context menu");
```

```
    }
```

```
    virtual void Close()
```

```
    {
```

```
delete this;      // window is on the heap
}
```

```
void AddMainWin()
{
    (new MainWin)->OpenMain();
}
```

```
void NoEffect()
{
    if (popUp.IsPoppedUp())
        return;
    popUp.SetPoppedUp(true);
    Rect mainRect = GetScreenRect();
    popUp.SetRect(mainRect.left + 20, mainRect.top + 20, 300, 225);
    popUp.SetColor(TextCtrl::PAPER_NORMAL, LtBlue);
    popUp.SetColor(TextCtrl::INK_NORMAL, Yellow);
    popUp.Set(String("no effect\nclick right to close"));
    popUp.PopUp
    (
        this,      // Ctrl* owner = NULL
        true,      // bool savebits = true
        true,      // bool activate = true
        true,      // bool dropshadow = false
        false     // bool topmost = false
    );
}
```

```
void SlideEffect()
{
    if (popUp.IsPoppedUp())
        return;
    popUp.SetPoppedUp(true);
    Rect mainRect = GetScreenRect();
    popUp.SetRect(mainRect.left + 20, mainRect.top + 20, 4, 3);
    popUp.SetColor(TextCtrl::PAPER_NORMAL, LtRed);
    popUp.SetColor(TextCtrl::INK_NORMAL, White);
    popUp.Set(String("slide effect\nclick right to close"));
    popUp.PopUp
    (
        this,      // Ctrl* owner = NULL
        true,      // bool savebits = true
        true,      // bool activate = true
        true,      // bool dropshadow = false
        false     // bool topmost = false
    );
    Ctrl::ProcessEvents();
    Animate(popUp, RectC(mainRect.left + 20, mainRect.top + 20, 300, 225), GUIEFFECT_SLIDE);
}
```

```

}

void FadeEffect()
{
    if (popUp.IsPoppedUp())
        return;
    popUp.SetPoppedUp(true);
    Rect mainRect = GetScreenRect();
    popUp.SetRect(mainRect.left + 20, mainRect.top + 20, 300, 225);
    popUp.SetColor(TextCtrl::PAPER_NORMAL, LtGreen);
    popUp.SetColor(TextCtrl::INK_NORMAL, Blue);
    popUp.Set(String("fade effect\n click right to close"));
    popUp.PopUp
    (
        this,      // Ctrl* owner = NULL
        true,      // bool savebits = true
        true,      // bool activate = true
        true,      // bool dropshadow = false
        false      // bool topmost = false
    );
    Ctrl::ProcessEvents();
    Animate(popUp, RectC(mainRect.left + 20, mainRect.top + 20, 300, 225), GUIEFFECT_FADE);
}

void local_menu(Bar& bar)
{
    MenuBar local_menu;

    bar.Add("new main window", THISBACK(AddMainWin));
    bar.Add("subwindow with no effect", THISBACK(NoEffect));
    bar.Add("subwindow with slide effect", THISBACK(SlideEffect));
    bar.Add("subwindow with fade effect", THISBACK(FadeEffect));
    local_menu.Execute();
}

void LeftDown(Point p, dword keyFlags)
{
    MenuBar::Execute(THISBACK(local_menu));
}

public:
MainWin()
{
    ++mainWinCount;
    String title("Multi-windowed App with Subwindow: Window # ");
    Title(title + AsString(mainWinCount));
    SetRect(40 * mainWinCount, 40 * mainWinCount, 400, 300);
    NoCenter();
}

```

```
}

~MainWin()
{
    --mainWinCount;
}
};

int MainWin::mainWinCount = 0;

GUI_APP_MAIN
{
    (new MainWin)->OpenMain();    // anytime deleteable
    Ctrl::EventLoop();
}
```

Enjoy!

Werner
