
Subject: Close context menu from within Execute() handler
Posted by [Mindtraveller](#) on Wed, 17 Feb 2010 23:23:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

I have a EditField control inside MenuBar, which is shown with Execute(). WhenEnter virtual function of EditField is overridden so that user may write some custom text in this field and then press Enter. Of course, after that, context menu should be closed.

I tried a number of functions (Close/CloseMenu/Hide/etc.) to close context menu automatically inside WhenEnter, but none of them worked.

Debugging led me to a condition in Ctrl.cpp @ 675:

```
void Ctrl::Close()
{
    GuiLock __;
    Ctrl *q = GetTopCtrl();
    if(!q->top) return;
    //...
```

which is never true in this case.

Please help, I meet this issue for the third time and still have no solution for it.

Subject: Re: Close context menu from within Execute() handler
Posted by [andrei_natanael](#) on Thu, 18 Feb 2010 00:08:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,
What about using menuBar.Break(); ?
It works for me.

Andrei

Subject: Re: Close context menu from within Execute() handler
Posted by [Mindtraveller](#) on Thu, 18 Feb 2010 09:39:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

No, Break() doesn't work too. Looks like it's due to MenuBar contains control which has focus. If I do Hide() for that control and then do Break() for MenuBar, it doesn't work either.

Subject: Re: Close context menu from within Execute() handler
Posted by [andrei_natanael](#) on Thu, 18 Feb 2010 10:44:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mindtraveller wrote on Thu, 18 February 2010 11:39No, Break() doesn't work too. Looks like it's due to MenuBar contains control which has focus.

If I do Hide() for that control and then do Break() for MenuBar, it doesn't work either. Yes, you're right. Perhaps using menuBar.PopUp(); ? That definitely works.

Subject: Re: Close context menu from within Execute() handler
Posted by [andrei_natanael](#) on Thu, 18 Feb 2010 11:07:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

A little sample;

```
class MenuTest : public TopWindow {
public:
    typedef MenuTest CLASSNAME;
    MenuTest()
    {
        menuBar.Set(THISBACK(ThisMenu));
        ef.WhenEnter = THISBACK(EnterPressed);
    }
    void ThisMenu(Bar& bar)
    {
        bar.Gap(10);
        bar.Add(ef.SizePos(), Size(200, 20));
        bar.Gap(10);
        bar.Add("One", THISBACK(Action));
        bar.Add("Two", THISBACK(Action));
    }
    void EnterPressed()
    {
        PromptOK(String(~ef));
    }
    void RightDown(Point p, dword d)
    {
        if (menuBar.IsOpen()) {
            menuBar.Close();
            menuBar.PopUp();
        } else
            menuBar.PopUp();
    }
    void Action() {}
    EditField ef;
    MenuBar menuBar;
};
```

Subject: Re: Close context menu from within Execute() handler
Posted by [Mindtraveller](#) on Fri, 19 Feb 2010 08:35:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

Showing popup dialog certainly closes previously opened context menu, so it's not exactly what I wanted.

I've modified your example a bit, and your idea using PopUp/Close worked! Thanks a lot!

Still I want to warn everyone: don't show menu with Execute() if you want it to be closed automatically, it doesn't work in any way. Just use PopUp/Close.

Slightly modified Andrei's working example is following
(thank you, Andrei, again!)

```
#include <CtrlLib/CtrlLib.h>
using namespace Upp;

class MenuTest : public TopWindow {
public:
    typedef MenuTest CLASSNAME;
    MenuTest()
    {
        menuBar.Set(THISBACK(ThisMenu));
        ef.WhenEnter = THISBACK(EnterPressed);
    }
    void ThisMenu(Bar& bar)
    {
        bar.Gap(10);
        bar.Add(ef.SizePos(), Size(200, 20));
        bar.Gap(10);
        bar.Add("One", THISBACK(Action));
        bar.Add("Two", THISBACK(Action));
    }
    void EnterPressed()
    {
        //...
        //some internal work
        //...
        menuBar.Close(); // <-- try to close menu here
    }
    void RightDown(Point p, dword d)
    {
        if (menuBar.IsOpen()) {
            menuBar.Close();
            menuBar.PopUp();
        } else
            menuBar.PopUp();
    }
    void Action() {}
    EditField ef;
    MenuBar menuBar;
};
```

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

Subject: Re: Close context menu from within Execute() handler
Posted by [mirek](#) on Mon, 22 Feb 2010 08:21:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

I believe that Break should work on Execute, if it does not, I will try to fix it.

Anyway, testcase would be helpful to be sure I am following exactly your situation.

Subject: Re: Close context menu from within Execute() handler
Posted by [gxl117](#) on Mon, 22 Feb 2010 12:55:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

For testcase, if close context menu by the press ESC key. menu can't call again by mouse rightkey!

Subject: Re: Close context menu from within Execute() handler
Posted by [gxl117](#) on Mon, 22 Feb 2010 14:36:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

The following example is normal.

```
#include <CtrlLib/CtrlLib.h>
using namespace Upp;
```

```
class MenuTest : public TopWindow {
public:
    typedef MenuTest CLASSNAME;
    MenuTest()
    {
        ef.WhenEnter = THISBACK(EnterPressed);
    }
    void ThisMenu(Bar& bar)
    {
        bar.Gap(10);
        bar.Add(ef.SizePos(), Size(200, 20));
        bar.Gap(10);
        bar.Add("One", THISBACK(Action));
        bar.Add("Two", THISBACK(Action));
    }
}
```

```

void EnterPressed()
{
    //...
    //some internal work
    //...
    //PromptOK(AsString(~ef));
    CloseTopCtrls(); // <-- try to close menu here

}
void RightDown(Point p, dword d)
{
    MenuBar::Execute(THISBACK(ThisMenu));

}
void Action() {}
EditField ef;

};

GUI_APP_MAIN
{
    MenuTest().Run();
}

```

Subject: Re: Close context menu from within Execute() handler

Posted by [mirek](#) on Fri, 26 Feb 2010 15:06:22 GMT

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

It needed a little fix in MenuBar code and it is not quite transparent and usual code, but it works now:

```

#include <CtrlLib/CtrlLib.h>

using namespace Upp;

class MenuTest : public TopWindow {
public:
    MenuBar menu;

    typedef MenuTest CLASSNAME;
    MenuTest()
    {
        ef.WhenEnter = THISBACK(EnterPressed);
    }
    void ThisMenu(Bar& bar)

```

```
{
    bar.Gap(10);
    bar.Add(ef.SizePos(), Size(200, 20));
    bar.Gap(10);
    bar.Add("One", THISBACK(Action));
    bar.Add("Two", THISBACK(Action));
}
```

```
void EnterPressed()
```

```
{
    menu.EndLoop();
}
```

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

```
{
    menu.Set(THISBACK(ThisMenu));
    menu.Execute();
}
```

```
void Action() {}
```

```
EditField ef;
```

```
};
```

```
GUI_APP_MAIN
```

```
{
    MenuTest().Run();
}
```