
Subject: Lambda support in MenuBar

Posted by [mirek](#) on Mon, 06 Jun 2016 18:29:20 GMT

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

Menu bar now has new methods for direct lambda support, as demonstrated by upgraded reference example:

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

using namespace Upp;

struct App : public TopWindow {
    bool numbers_enabled;

    void Exit()
    {
        Close();
    }

    void SubMenu(Bar& bar)
    {
        for(int i = 0; i < 10; i++)
            bar.Add(~AsString(i), [=] { PromptOK(AsString(i)); });
    }

    void Menu(Bar& bar)
    {
        bar.Add("Enable numbers", [=] { numbers_enabled = !numbers_enabled; })
            .Check(numbers_enabled);
        bar.Add(numbers_enabled, "Numbers", THISBACK(SubMenu));
        bar.Add("Exit", [=] { Exit(); })
            .Key(K_CTRL_E);
    }

    void MainBar(Bar& bar)
    {
        bar.Add("Numbers", THISBACK(Menu));
        bar.Sub("Items", [=](Bar& bar) {
            bar.Add("Item 1", [&] { Exclamation("Item 1 invoked"); });
            bar.Add("Item 2", [&] { Exclamation("Item 2 invoked"); });
        });
    }
};

MenuBar menu;

typedef App CLASSNAME;
```

```
App()
{
  numbers_enabled = false;
  AddFrame(menu);
  menu.Set(THISBACK(MainBar));
}
};
```

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

Nice thing is the possibility to define submenu 'inline'. Unfortunately, I had to rename the method name to 'Sub' (instead of Add) because of overloading ambiguity issues...

Mirek
