
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

Subject: Re: Lambda support in MenuBar
Posted by [Klugier](#) on Mon, 06 Jun 2016 20:03:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Mirek,

It seems that this is a future that everybody waits for. It will help develop several things inside TheIDE. I will plan to use it in our "New file" implementation in near future.

Sincerely and thanks for information and implementation ,
Klugier

Subject: Re: Lambda support in MenuBar
Posted by [Klugier](#) on Sat, 11 Jun 2016 09:23:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Mirek - Now more serious,

I tried to integrated new lambda callback with ide New file menu - but there is a problem. Here is my new code:

```
void WorkspaceWork::NewMenu(Bar& bar)
{
  bar.Add("File", CtrlImg::File(), THISBACK2(NewPackageFile, "New file", ""));
  bar.Add("Separator", THISBACK(AddSeparator))
}
```

```

.Help("Add text separator line");

bar.Separator();
for (int i = 0; i < categories.GetCount(); i++) {
    // Lambda in place - as argument
    bar.Add(categories.GetKey(i), [&, i](Bar& subBar) {
        for (FileType& fileType : categories[i]) {
            String name = fileType.GetName();
            String extension = fileType.GetExtension();

            if (fileType.IsSeparator())
                subBar.Separator();
            else
                subBar.Add(name, fileType.GetImage(), THISBACK2(NewPackageFile, "New " + name,
extension));
        }
    });
}
}

```

And i have got following compilation issue:

```

/home/klugier/upp/uppsrc/Core/Callback.h: In instantiation of 'Res Upp::Function<Res(ArgTypes
...)>::Wrapper<F>::Execute(ArgTypes ...) [with F =
WorkspaceWork::NewMenu(Upp::Bar&)::<lambda(Upp::Bar&)>; Res = void; ArgTypes = {}]
':
/home/klugier/upp/uppsrc/ide/UppWspc.cpp:1238:1: required from here
/home/klugier/upp/uppsrc/Core/Callback.h:15:60: error: no match for call to
'(WorkspaceWork::NewMenu(Upp::Bar&)::<lambda(Upp::Bar&)>) ()'
    virtual Res Execute(ArgTypes... args) { return fn(args...); }
            ^
/home/klugier/upp/uppsrc/ide/UppWspc.cpp:931:51: note: candidate:
WorkspaceWork::NewMenu(Upp::Bar&)::<lambda(Upp::Bar&)>
    bar.Add(categories.GetKey(i), [&, i](Bar& subBar) {
            ^
/home/klugier/upp/uppsrc/ide/UppWspc.cpp:931:51: note: candidate expects 1 argument, 0
provided
In file included from /home/klugier/upp/uppsrc/Core/Core.h:293:0,
    from /home/klugier/upp/uppsrc/Esc/Esc.h:4,
    from /home/klugier/upp/uppsrc/ide/Core/Core.h:4,
    from /home/klugier/upp/uppsrc/ide/Common/Common.h:4,
    from /home/klugier/upp/uppsrc/ide/ide.h:4,
    from /home/klugier/upp/uppsrc/ide/UppWspc.cpp:1:
/home/klugier/upp/uppsrc/Core/Callback.h:15:60: error: return-statement with a value, in function
returning 'void' [-fpermissive]
    virtual Res Execute(ArgTypes... args) { return fn(args...); }

```

Is that problem here that bar lambda is created in THISBACK method?

Sincerely,
Klugier

Subject: Re: Lambda support in MenuBar
Posted by [Klugier](#) on Sat, 11 Jun 2016 10:55:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

Solved! Just replace bar.Add with bar.Sub when the lambda sequence occurred and everything works fine.

Here is polished version of above wrong code (Maybe someone come in handy):

```
void WorkspaceWork::NewMenu(Bar& bar)
{
    bar.Add("File", CtrlImg::File(), [=] { NewPackageFile("New file", ""); });
    bar.Add("Separator", [=] { AddSeparator(); })
        .Help("Add text separator line");

    bar.Separator();
    for (int i = 0; i < categories.GetCount(); i++) {
        bar.Sub(categories.GetKey(i), [=](Bar& subBar) {
            for (FileType& fileType : categories[i]) {
                String name = fileType.GetName();
                String extension = fileType.GetExtension();

                if (fileType.IsSeparator())
                    subBar.Separator();
                else
                    subBar.Add(name, fileType.GetImage(), [=] { NewPackageFile("New " + name, extension); });
            }
        });
    }
}
```

Sincerely,
Klugier

Subject: Re: Lambda support in MenuBar
Posted by [slashupp](#) on Tue, 16 Aug 2016 20:00:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

thx mirek

doesn't seem to support auto lambdas with parameters (yet?)
e.g.:

```
void Menu(Bar& bar)
{
    //auto num_ok = [&](bool b){ numbers_enabled = b; };
    //std::function<void(bool)> num_ok = [&](bool b){ numbers_enabled = b; };

    //bar.Add("Enable numbers", [=] { numbers_enabled = !numbers_enabled; }) //--orig code
    //bar.Add("Enable numbers", THISBACK1(num_ok, !numbers_enabled) ) //fail
    //bar.Add("Enable numbers", Callback1<bool>(num_ok, !numbers_enabled) ) //fail
    .Check(numbers_enabled);

    bar.Add(numbers_enabled, "Numbers", THISBACK(SubMenu));
    bar.Add("Exit", [=] { Exit(); })
        .Key(K_CTRL_E);
}
```