
Subject: Callbacks mechanism doesn't handle overloaded functions?

Posted by [Peter](#) on Wed, 10 Oct 2012 19:05:55 GMT

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

Hi.

I'm a new U++ user with little experience, especially in how the framework important features work internally. That's why I'm not sure if I discovered a bug or merely a framework limitation. Here's a code snippet to demonstrate the problem:

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

using namespace Upp;

class MyWindow : public TopWindow
{
    Button btn;

    void Fun() {PromptOK(AsString(5));}
    void Fun(int n){PromptOK(AsString(n));}

public:
    typedef MyWindow CLASSNAME;
    MyWindow()
    {
        btn <<= THISBACK(Fun);
    }
};

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

In the code above you can see two different, overloaded functions named Fun. When I try to compile it, compiler reports the following error:

```
error C2668: 'Upp::callback' : ambiguous call to overloaded function
```

Surprisingly, if I change the following line:

```
btn <<= THISBACK(Fun); //reference to void Fun()
```

to:

```
btn <<= THISBACK1(Fun, 1); // reference to void Fun(int)
```

the code compiles fine.

What causes the error in the first case? I don't see any ambiguity here since `void Fun()` and `void Fun(int)` are two distinct functions and `THISBACK` macro should most likely treat them as such. Is it a design/implementation oversight or rather a natural consequence of how the callback mechanism internally works and callbacks simply don't work with overloaded functions the way one would expect? If the latter is the case, however, how come there's no ambiguity in the second case?

Peter

Subject: Re: Callbacks mechanism doesn't handle overloaded functions?

Posted by [mirek](#) on Wed, 10 Oct 2012 19:50:20 GMT

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

Peter wrote on Wed, 10 October 2012 15:05Hi.

I'm a new U++ user with little experience, especially in how the framework important features work internally. That's why I'm not sure if I discovered a bug or merely a framework limitation. Here's a code snippet to demonstrate the problem:

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

using namespace Upp;

class MyWindow : public TopWindow
{
    Button btn;

    void Fun() {PromptOK(AsString(5));}
    void Fun(int n){PromptOK(AsString(n));}

public:
    typedef MyWindow CLASSNAME;
    MyWindow()
    {
        btn <<= THISBACK(Fun);
    }
};
```

GUI_APP_MAIN

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

In the code above you can see two different, overloaded functions named Fun. When I try to compile it, compiler reports the following error:

error C2668: 'Upp::callback' : ambiguous call to overloaded function

Well, while it is clear that btn.WhenAction is no-parameter Callback, THISBACK(Fun) is evaluated (by compiler) independently of this fact. As such, it is indeed ambiguous whether you mean Callback with no parameters or with single parameter at this point.

Unfortunately, there is no nice solution here (C++ language limitation). You have to name your methods differently...

Mirek
