
Subject: Callback: C++11 lambda support
Posted by [mirek](#) on Mon, 17 Mar 2014 18:07:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

You can now assign lambda to callback. Have to say, simplifies things quite a lot, no more need to add methods to dialogs...

Subject: Re: Callback: C++11 lambda support
Posted by [Klugier](#) on Sat, 22 Mar 2014 12:28:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Mirek,

For me this is killer feature. Thanks!

Sincerely,
Klugier

Subject: Re: Callback: C++11 lambda support
Posted by [Didier](#) on Thu, 27 Mar 2014 21:14:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Mirek,

It looks like the additional features broke some existing code:

I get the following errors when trying to compile docking
package:/home/didier/upp/uppsrc/Docking/DockCont.cpp: In constructor

```
tabbar.WhenCloseSome = THISBACK(TabsClosed);
^
/home/didier/upp/uppsrc/Docking/DockCont.cpp:824:23: note: candidate is:
In file included from /home/didier/upp/uppsrc/Core/Callback.h:6:0,
      from /home/didier/upp/uppsrc/Core/Core.h:301,
      from /home/didier/upp/uppsrc/Draw/Draw.h:6,
      from /home/didier/upp/uppsrc/RichText/RichText.h:4,
      from /home/didier/upp/uppsrc/CtrlCore/CtrlCore.h:4,
      from /home/didier/upp/uppsrc/CtrlLib/CtrlLib.h:4,
      from /home/didier/upp/uppsrc/Docking/Docking.h:4,
      from /home/didier/upp/uppsrc/Docking/DockWindow.cpp:1,
      from /home/didier/upp.out/MyApps/Docking/GCC.Blitz.Gui.Mt.Shared/$blitz.cpp:3:
/home/didier/upp/uppsrc/Core/Cbgen.h:117:16: note: Upp::Callback1<P1>&
```

```
Upp::Callback1<P1>::operator=(const Upp::Callback1<P1>&) [with P1 = Upp::ValueArray]
Callback1<P1>& Callback1<P1>::operator=(const Callback1& c)
^
/home/didier/upp/uppsrc/Core/Cbgen.h:117:16: note:  no known conversion for argument 1 from
```

Subject: Re: Callback: C++11 lambda support
Posted by [Didier](#) on Thu, 27 Mar 2014 21:22:08 GMT

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

In Docking/DockCont.h & .cpp replacing :
void TabsClosed(Vector<Value> vv);
with
void TabsClosed(ValueArray vv);
solves the issue

Subject: Re: Callback: C++11 lambda support
Posted by [koldo](#) on Mon, 31 Mar 2014 11:35:01 GMT

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

Hello Mirek

Could you put a simple sample of it?

Subject: Re: Callback: C++11 lambda support
Posted by [mirek](#) on Mon, 31 Mar 2014 12:14:06 GMT

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

foo.WhenAction = [\$]() { foo <<= (int)~foo + 1; }

Anyway, there is still some work to be done; the main issues are overloading problems. Only Callback::operator= works so far.

The problem is that in order to capture lambda, you have to provide completely 'all-catch' template

template <class T>
void CaptureLambda(T x);

which then is causing overloading problems..

I am almost inclined to add some stronger typing via macro, like

```
foo <=> lambda() { foo <=> (int)~foo + 1; }
```

and lambda would provide a case to something more reasonable, like `#define lambda (LambdaCallback)[&]`.

Subject: Re: Callback: C++11 lambda support

Posted by [koldo](#) on Mon, 31 Mar 2014 14:22:47 GMT

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

Subject: Re: Callback: C++11 lambda support

Posted by [mirek](#) on Sat, 12 Apr 2014 17:32:15 GMT

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

After some initial experience, overloading problems proved to be too ugly, so putting lambda into Callback now requires call to "lambda" function or using LAMBDA macro:

```
Callback1<int> cb = lambda(&)(int x) { something(x); };
```

```
Callback cb = LAMBDA(int x) { something(x); };
```
