Subject: Re: [c++11] Problem with executing callback with Vector
Posted by mirek on Fri, 29 May 2015 06:32:01 GMT
View Forum Message <> Reply to Message

Zbych wrote on Thu, 28 May 2015 14:43Hi,

When I'm trying to pass a vector to a callback routine:

```
GUI_APP_MAIN
{
 Callback1< Vector<int> > SomeCallback;
 Vector<int> v;
 SomeCallback(v);
}
```

GCC in c++11 mode gives me error: "use of deleted function":

```
/home/zbych/upp/uppsrc/Core/Cbgen.h:122:7: error:   initializing argument 1 of void
Upp::Callback1<P1>::operator()(P1) const [with P1 = Upp::Vector<Upp::String>]
  void operator()(P1 p1) const   { Execute(p1); }
       ^
/home/zbych/upp/uppsrc/Core/Cbgen.h: In instantiation of void
Upp::Callback1<P1>::operator()(P1) const [with P1 = Upp::Vector<Upp::String>]:
/home/zbych/MyApps/GtkTest/main.cpp:46:16:   required from here
/home/zbych/upp/uppsrc/Core/Cbgen.h:122:45: error: use of deleted function constexpr
Upp::Vector<Upp::String>::Vector(const Upp::Vector<Upp::String>&)
  void operator()(P1 p1) const   { Execute(p1); }
                                             ^
/home/zbych/upp/uppsrc/Core/Cbgen.h:121:7: error:   initializing argument 1 of void
Upp::Callback1<P1>::Execute(P1) const [with P1 = Upp::Vector<Upp::String>]
  void Execute(P1 p1) const      { if(action) action->Execute(p1); }
       ^
/home/zbych/upp/uppsrc/Core/Cbgen.h: In instantiation of void Upp::Callback1<P1>::Execute(P1)
const [with P1 = Upp::Vector<Upp::String>]:
/home/zbych/upp/uppsrc/Core/Cbgen.h:122:45:   required from void
Upp::Callback1<P1>::operator()(P1) const [with P1 = Upp::Vector<Upp::String>]
/home/zbych/MyApps/GtkTest/main.cpp:46:16:   required from here
/home/zbych/upp/uppsrc/Core/Cbgen.h:121:46: error: use of deleted function constexpr
Upp::Vector<Upp::String>::Vector(const Upp::Vector<Upp::String>&)
  void Execute(P1 p1) const      { if(action) action->Execute(p1); }
                                             ^
/home/zbych/upp/uppsrc/Core/Cbgen.h:85:15: error:   initializing argument 1 of void
Upp::Callback1Action<P1>::Execute(P1) [with P1 = Upp::Vector<Upp::String>]
  virtual void Execute(P1 p1) = 0;
```

What am I doing wrong?

Passing Vector to callback...

Well, the trouble is you need copy and containers do not have "direct" copy.

Quick fix is IMO to use Callback1< WithDeepCopy< Vector<int> > >.

Another quick fix is to use Value[Array] instead.

If you insist on using Vector directly (both above options come at certain performance penalty), you need to reorganize the code so that you can pass reference or pointer.

Anyway, as long as you want to use PostCallback, I think your example is not correct, as PostCallback accepts simple Callback (no parameters). Perhaps you meant using something like THISBACK1 ?

If so, you could perhaps use lambda to overcome the problem...

Mirek
