
Subject: Re: AsyncWork<Vector<T>> fails due to lack of copy-constructor

Posted by [piotr5](#) on Mon, 06 Jan 2020 23:14:51 GMT

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

```
In file included from /home/p/upp/uppsrc/Core/Core.h:342,
                 from /home/p/upp/uppsrc/Draw/Draw.h:6,
                 from /home/p/upp/uppsrc/RichText/RichText.h:4,
                 from /home/p/upp/uppsrc/CtrlCore/CtrlCore.h:4,
                 from /home/p/upp/uppsrc/CtrlLib/CtrlLib.h:4,
                 from /home/p/MyApps/arrow2circle/arrow2circle.h:4,
                 from /home/p/MyApps/arrow2circle/main.cpp:1:
/home/p/upp/uppsrc/Core/CoWork.h: In instantiation of 'Ret2
Upp::AsyncWork<Ret>::Imp<Ret2>::Get() [with Ret2 = Upp::Vector<unsigned char>; Ret =
Upp::Vector<unsigned char>]':
/home/p/upp/uppsrc/Core/CoWork.h:195:99:   required from 'Ret Upp::AsyncWork<Ret>::Get()
[with Ret = Upp::Vector<unsigned char>]'
/home/p/MyApps/arrow2circle/arrow2circle.h:28:18:   required from here
/home/p/upp/uppsrc/Core/CoWork.h:173:50: error: use of deleted function 'constexpr
Upp::Vector<unsigned char>::Vector(const Upp::Vector<unsigned char>&)'
 173 |   Ret2 Get()           { return ret; }
      |           ^~~
```

similar if I add "pick(ret)" in place of "ret"

```
/home/p/upp/uppsrc/Core/CoWork.h: In instantiation of 'Ret2&
Upp::AsyncWork<Ret>::Imp<Ret2>::Get() [with Ret2 = Upp::Vector<unsigned char>; Ret =
Upp::Vector<unsigned char>]':
/home/p/upp/uppsrc/Core/CoWork.h:195:99:   required from 'Ret Upp::AsyncWork<Ret>::Get()
[with Ret = Upp::Vector<unsigned char>]'
/home/p/MyApps/arrow2circle/arrow2circle.h:28:18:   required from here
/home/p/upp/uppsrc/Core/CoWork.h:173:59: error: cannot bind non-const lvalue reference of type
'Upp::Vector<unsigned char>&' to an rvalue of type
'std::remove_reference<Upp::Vector<unsigned char>&>::type' {aka 'U
pp::Vector<unsigned char>'}
```

my code is just simple file-processing, here a short version:

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

```
#include <array>
```

```
using namespace Upp;
```

```
struct Parse {
    using CLASSNAME=Parse;
    std::array<AsyncWork<Vector<byte> >,3 > parser;
    FindFile path;
```

```
void init(const String& p) {path.Search(p); for(auto&& i:parser) {i.Cancel();SetWork(i);};}
```

```
Vector<byte> Work(const String& file);
```

```
void SetWork(AsyncWork<Vector<byte> >& where) {if(path&&path.IsFile()&&path.GetLength()>0)
```

```

{
    where.Do(THISFN(Work),path.GetPath());
    path.Next();
};}
void AddWork() { for(auto&& i:parser) if(i.IsFinished()) {
    auto out=i.Get();
    SetWork(i);
}}
};

```

```

Vector<byte> Parse::Work(const String& file) {
    Vector<byte> out;
    return out;
}
CONSOLE_APP_MAIN
{
    Parse p;
    p.init(CommandLine()[1]);
}

```

this works with

```
Ret Get() { return pick(ret); }
```

and also if I put "Ret&&" in place of "Ret" and use pick as above. using "std::forward<Ret2>(ret)" is a better idea here? (I compile with "-std=gnu++1z" and gcc-9.2.0)

Oh, and thanks for the hint. I'll try using CoWork to see if it's any better than my boilerplate...