Subject: AsyncWork<Vector<T>> fails due to lack of copy-constructor
Posted by piotr5 on Mon, 06 Jan 2020 14:38:13 GMT
View Forum Message <> Reply to Message

I created AsyncWork<Vector<char>> and when I make use of the Get() member compiler complains Vector only has move-constructor.
is this a bug? CoWork.h reads:

```
template <class Ret>
class AsyncWork {
 template <class Ret2>
 struct Imp {
  CoWork co;
  Ret2   ret;
...
  const Ret2& Get()                       { return ret; }
 };
...
 Ret        Get()                       { ASSERT(imp); imp->co.Finish(); return imp->Get(); }
```

it's strange there is a 2nd template parameter for this class inside of AsyncWork.
sounds more like it was meant to be:

```
  Ret   ret;
...
  Ret2 Get()                       { return ret; }
```

and then use "Imp<const Ret&> imp;" in case Ret has copy-constructor?
or maybe rewrite to use if constexpr?
is there a speed-gain from using the copy-constructor?
I doubt there is one for Vector as return-type...

or did I make a mistake and AsyncWork is not for this kind of usage?
should I rewrite my code to use Thread?