Subject: Re: How to mark std::array<T, N> moveable if only T is moveable
Posted by mirek on Thu, 25 Jul 2024 07:19:38 GMT
View Forum Message <> Reply to Message

Lance wrote on Thu, 25 July 2024 03:16
The capability to manually mark a class might still be an asset. Suppose we receive a third party class whose objects are trivially relocatable but not trivially copyable, the simplest way to make Upp::Vector accept it is to mark it. Wrongfully marked class will usually result in immediate runtime error, hence should not be a big concern.

I disagree. They might be relocatable in that version of library with given compiler only. Immediate runtime error might happen when you do testing, but developer might not be even aware he is supposed to test it.

Also from practical point of view, this never happened

What might be usefull though is the capability to manually mark a class to be relocated by regular move/copy constructor:

- I still think it is a good idea not to do move/copy as default option to force effectivness

- but this would allow things like Vector<std::string> with some performance penalty...

Another thing to consider: Tuple is moveable if all its components are moveable. Are we able to express that somehow? (If not, no big deal, I can make Tuple moveable and request that elements are).

Mirek