Subject: Re: error: call to implicitly-deleted copy constructor
Posted by peterh on Tue, 23 Feb 2021 18:44:10 GMT
View Forum Message <> Reply to Message

Finding the problem was very hard for me some months ago, because I am not very skilled and have no experience and only basic knowlege about STL.

So an explanation how to locate this error:
In the Message list there is one error displayed, deeply buried in a template. Click this and a long list of messages opens.
It is the best to use clang, because the error messages are more helpful than those given by MSVC:
The last messages read:

Quote:
C:\upp\MyApps\test\test.cpp (53): note: in instantiation of member function 'std::__1::deque<SignalStatusMessage, std::__1::allocator<SignalStatusMessage>>::push_back' requested here
 (): deqSigStatus.push_back((sigStatus));
C:\upp\MyApps\test\test.cpp (24): note: copy constructor of 'SignalStatusMessage' is implicitly deleted because field 'status' has a deleted copy constructor
 (): Vector<SignalStatus> status;
C:\upp/uppsrc\Core/Vcont.h (288): note: copy constructor is implicitly deleted because 'Vector<SignalStatus>' has a user-declared move constructor
 (): Vector(Vector&& v)   { Pick(pick(v)); }
 (): 1 error generated.

Look in this list for notes about your own code. These indicate where the error could be caused, but it could only be detected later, when the template was evaluated.
If you know this, it becomes easier.

About the STL: I am not sure it is a design, it is evolutionary grown.
I have watched a lot of Bjarne Stroustrups lectures on youtube and the most important thing for him is progress, but equally important preserving backward compatibility and maintenance of some 20 years old very large code bases.
He discusses also the problem of unnecessary copies and how to avoid and which language and STL features are in the pipeline in the hope to improve this.
I think "concepts" added to templates will in future help to detect and display such errors early.