Subject: C2280
Posted by NilaT on Thu, 02 Feb 2017 11:49:13 GMT
View Forum Message <> Reply to Message

Hello everyone,

I have another, very urgent question.
I want to compile my project with 2015 sources and MSC15...
Some errors here, some warnings there, not worth be mentioned.
But, one error is very annoying and I can't get rid of it.
It's C2280...
test.cpp (191): error C2280: "Upp::Array<TEST> &Upp::Array<TEST>::operator =(const Upp::Array<TEST> &)" : Es wurde versucht, auf eine gelöschte Funktion zu verweisen

I already found this:
http://stackoverflow.com/questions/31264984/c-compiler-error-c2280-attempting-to-reference-a-deleted-function-in-visual

which says the Array class needs a "= default" constructor and copy constructor or something.
I tried a few things but nothing worked.

Please help.
Very urgent!
Any help appreciated.
Thanks in advance.

PS: Please help me with my other Problem (RPC_METHOD) as well, thank you.
PPS: My exact version of Upp is 10249

//edit: When I change
void operator=(Array&& v) { if(this != &v) { Free(); vector = pick(v.vector); } }
to
Array& operator=(const Array&) = default;
in Vcont.h, the error won't occur but I don't think this is the right solution...

And then the same error occurs for Upp::XmlNode...
Very annoying...

Subject: Re: C2280
Posted by NilaT on Fri, 03 Feb 2017 10:11:32 GMT
View Forum Message <> Reply to Message

Please?

File Attachments
1) Unbenannt.png, downloaded 390 times

Subject: Re: C2280
Posted by cbpporter on Fri, 03 Feb 2017 10:55:09 GMT
View Forum Message <> Reply to Message

I Google translated that error to "An attempt was made to refer to a deleted function".

It is not the U++ Array you need to add the = default to, but the class you are using.

It is hard to fix this without seeing the code and if your class is copyable or not, but try something from:

http://www.ultimatepp.org/srcdoc$Core$pick_$en-us.html

rval_default(Foo)

Subject: Re: C2280
Posted by NilaT on Sat, 04 Feb 2017 00:29:04 GMT
View Forum Message <> Reply to Message

Hello and thanks for your reply.

But it MUST be the U++ Array, as my variables are Array<String> and Array of a own structure.
Even the U++ ArrayMap and XmlNode have this issue.

I can post you some code tomorrow if you want, but I guarantee it won't help here.

Any other suggestions?
I will try the rval_default part tomorrow though

Subject: Re: C2280
Posted by Mindtraveller on Sat, 04 Feb 2017 20:16:42 GMT
View Forum Message <> Reply to Message

Most of time this error means you need to write
b = pick(a);
or
b = clone(a);
instead of
b = a;

Subject: Re: C2280
Posted by NilaT on Sun, 05 Feb 2017 12:50:32 GMT
View Forum Message <> Reply to Message

Thanks Mindtraveller, this seems to do the job.
May you can explain WHY?

Thanks ;)

Subject: Re: C2280
Posted by Mindtraveller on Mon, 06 Feb 2017 12:15:23 GMT
View Forum Message <> Reply to Message

http://www.ultimatepp.org/srcdoc$Core$pick_$en-us.html

U++ style of coding insists on explicit definition what you want to do: to clone object or to move it.
You mustn't write trivial code like
b = a;
fot U++ objects because it is hard to say whether you want to clone object or pick/move it from a to b.

P.S. It is rather uncommon for C++, but I do personally like this style. It is more strict and less error prone.

Subject: Re: C2280
Posted by NilaT on Mon, 06 Feb 2017 15:31:53 GMT
View Forum Message <> Reply to Message

Thank you, seems clear to me.
This must be something C++11 invented, right?
Because we use 2010 sources as well and there was no need for that.
Now, as we updated to 2015, it must be explicit defined...

Well, however, thank you for your help.

Subject: Re: C2280
Posted by Didier on Thu, 09 Feb 2017 19:29:37 GMT
View Forum Message <> Reply to Message

Hi,

In fact, the origin of the error is probably readable in the code
==> you may have something like the following:

xxxx operator=(xxxx i) = deleted;

Apparently (I haven't looked precisely) deleted indicates that the code is deprecated but can't be used anymore (it is still present to make it easier to do corrections: if you know what has been

removed).

I ran into the same type of error 2 days ago, and all the 'xx operator(const xx t)' where marked deleted ==> notice that only the CONST ones where deleted.
From my point of view these modifications are linked to the management of the move semantics introduced in C++11
