
Subject: Re: What is the highest version of U++ that does not require C++11 ?

Posted by [mirek](#) on Wed, 19 Apr 2017 13:19:27 GMT

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

aftershock wrote on Tue, 18 April 2017 21:45I have similar problem

I have current stable IDE..(version 10804). Windows....Visual studio 17 64 bit

Why is it calling a copy constructor at map.h 212

```
threads[free_index].Run ( THISBACK5 ( execute_bot_in_background, bot1, a, params,
result_mode, stat_group_id ) );
```

```
main.cpp(1502): error C2280: 'Upp::VectorMap<Upp::String,double>::VectorMap(const
Upp::VectorMap<Upp::String,double> &)': attempting to reference
a deleted function
```

```
d:\upp\uppsrc\core\Map.h(212): note: compiler has generated
```

```
'Upp::VectorMap<Upp::String,double>::VectorMap' here
```

```
d:\upp\uppsrc\core\Map.h(212): note: 'Upp::VectorMap<Upp::String,double>::VectorMap(const
Upp::VectorMap<Upp::String,double> &)': function was implicitly deleted becau
se a base class invokes a deleted or inaccessible function
```

```
'Upp::AMap<K,T,Upp::Vector<T>>::AMap(const Upp::AMap<K,T,Upp::Vector<T>> &)'
```

```
with
```

```
[
```

```
    K=Upp::String,
```

```
    T=double
```

```
]
```

```
d:\upp\uppsrc\core\Map.h(190): note: 'Upp::AMap<K,T,Upp::Vector<T>>::AMap(const
```

```
Upp::AMap<K,T,Upp::Vector<T>> &)': function was implicitly deleted because a data membe
r invokes a deleted or inaccessible function 'Upp::Index<Upp::String>::Index(const
```

```
Upp::Index<Upp::String> &)'
```

```
with
```

```
[
```

```
    K=Upp::String,
```

```
    T=double
```

```
]
```

```
d:\upp\uppsrc\core\Index.h(210): note: 'Upp::Index<Upp::String>::Index(const
```

```
Upp::Index<Upp::String> &)': function was implicitly deleted because 'Upp::Index<Upp::Stri
ng>' has a user-defined move constructor
```

```
]
```

Hard to say without context. However, it looks like you are passing VectorMap as parameter to THISBACK5 - that is not possible. Legacy U++ (pre C++11) was tricky as it has silently used pick constructor here.

The situation is really only resolved with current U++, lamdas and C++14.

[http://www.ultimatepp.org/srcdoc\\$Core\\$Tutorial\\$en-us.html](http://www.ultimatepp.org/srcdoc$Core$Tutorial$en-us.html)

