Subject: String isn't pickable?
Posted by piotr5 on Wed, 30 May 2012 16:41:23 GMT
View Forum Message <> Reply to Message

while compiling

```
Vector<String> v;
String a=s.Mid(from,to-from);
v.InsertPick(FindLowerBound(v,a),a);
```

I get the errors

```
In file included from C:\comp\upp\uppsrc/Core/Core.h:239:0,
                 from C:\comp\upp\uppsrc/Draw/Draw.h:6,
                 from C:\comp\upp\uppsrc/RichText/RichText.h:4,
                 from C:\comp\upp\uppsrc/CtrlCore/CtrlCore.h:4,
                 from C:\comp\upp\uppsrc/CtrlLib/CtrlLib.h:4,
                 from C:\Users\p\MyApps\bookcat\/bookcat.h:4,
                 from C:\Users\p\MyApps\bookcat\main.cpp:1:
C:\comp\upp\uppsrc/Core/Vcont.hpp: In member function 'T& Upp::Vector<T>::InsertPick(int,
const T&) [with T = Upp::String]':
C:\Users\p\MyApps\bookcat\main.cpp:129:38:   instantiated from here
C:\comp\upp\uppsrc/Core/Vcont.hpp:280:2: error: conversion from 'Upp::String' to 'void*' is
ambiguous
C:\comp\upp\uppsrc/Core/Vcont.hpp:280:2: note: candidates are:
C:\comp\upp\uppsrc/Core/String.h:54:2: note: Upp::AString<B>::operator const void*() const [with
B = Upp::String0] <near match>
C:\comp\upp\uppsrc/Core/String.h:54:2: note:   no known conversion for implicit 'this' parameter
from 'const void*' to 'void*'
C:\comp\upp\uppsrc/Core/String.h:53:2: note: Upp::AString<B>::operator const bchar*() const
[with B = Upp::String0, Upp::AString<B>::bchar = unsigne
 d char] <near match>
C:\comp\upp\uppsrc/Core/String.h:53:2: note:   no known conversion for implicit 'this' parameter
from 'const bchar* {aka const unsigned char*}' to '
 void*'
C:\comp\upp\uppsrc/Core/String.h:51:2: note: Upp::AString<B>::operator const tchar*() const [with
B = Upp::String0, Upp::AString<B>::tchar = char] <
 near match>
C:\comp\upp\uppsrc/Core/String.h:51:2: note:   no known conversion for implicit 'this' parameter
from 'const tchar* {aka const char*}' to 'void*'
c:\code\mingw\bin\../lib/gcc/mingw32/4.6.2/include/c++/new:103:14: error:   initializing argument 2
of 'void* operator new(std::size_t, void*)'
```

without the Pick it compiles normally. is that a new behaviour of gcc 4.6.2?

Subject: Re: String isn't pickable?

Posted by mirek on Thu, 31 May 2012 05:59:43 GMT
View Forum Message <> Reply to Message

Yes, String is not pickable. It is "full copy semantics" type with 'fast' deep copy.

In practice, only containers are pickable.

Subject: Re: String isn't pickable?
Posted by kohait00 on Tue, 05 Jun 2012 11:53:23 GMT
View Forum Message <> Reply to Message

what about copy on write, shared ptr otherwise?
or is the benefit neglectable?

Subject: Re: String isn't pickable?
Posted by mirek on Fri, 08 Jun 2012 13:37:59 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Tue, 05 June 2012 07:53what about copy on write, shared ptr otherwise?
or is the benefit neglectable?

Well, but now you are speaking about implementations, not interface semantics...

(In fact, String is using a lot of tricks to make copies fast, it has 3 storage levels, last one is using shared reference counted pointer, copy on write implementation).

Subject: Re: String isn't pickable?
Posted by kohait00 on Mon, 11 Jun 2012 07:00:50 GMT
View Forum Message <> Reply to Message

i didn't expect less from Upp
was just wondering..