Subject: U++ does not appear to like playing nice with the Boost algorithm string library?
Posted by ptkacz on Sat, 04 Apr 2020 01:07:41 GMT
View Forum Message <> Reply to Message

When I compile the following (console application) code in U++:

```
#include <iostream>
#include <string>
#include <iostream>
using namespace std;

#include <Core/Core.h>
using namespace Upp;

#include <boost/algorithm/string.hpp>

CONSOLE_APP_MAIN
{
    std::string str = "   Lots of filler space   ";
    boost::algorithm::trim(str);
    std::cout<<"\""<< str <<"\""<<std::endl;
}
```

the following error is resulting:

```
----- Core ( GCC DEBUG SHARED DEBUG_FULL BLITZ POSIX LINUX ) (1 / 2)
----- BoostStringTest ( MAIN GCC DEBUG SHARED DEBUG_FULL BLITZ POSIX LINUX ) (2 / 2)
BoostStringTest.cpp
In file included from /home/ptkacz/upp/uppsrc/Core/i18n.h:17:0,
                 from /home/ptkacz/upp/uppsrc/Core/Core.h:337,
                 from /home/ptkacz/MyApps/BoostStringTest/BoostStringTest.cpp:6:
/usr/include/boost/core/ref.hpp: In constructor
'boost::reference_wrapper<T>::reference_wrapper(T&)':
/home/ptkacz/upp/uppsrc/Core/t_.h:9:24: error: class 'boost::reference_wrapper<T>' does not
have any field named 't_GetLngString'
 #define t_(x)         t_GetLngString(x)
                       ^
BoostStringTest: 1 file(s) built in (0:01.55), 1551 msecs / file, duration = 1551 msecs

There were errors. (0:01.55)
```

I do know that U++ has it's only string utility functions, but was attempting to make use of code brought in from another application that I'm working on. Independently of any existing code, I wrote a simple C++ application in Code::Blocks, the code compiles and runs as expected.

Peter

Subject: Re: U++ does not appear to like playing nice with the Boost algorithm string library?
Posted by mirek on Sat, 04 Apr 2020 08:18:27 GMT
View Forum Message <> Reply to Message

ptkacz wrote on Sat, 04 April 2020 03:07When I compile the following (console application) code in U++:

```
#include <iostream>
#include <string>
#include <iostream>
using namespace std;

#include <Core/Core.h>
using namespace Upp;

#include <boost/algorithm/string.hpp>

CONSOLE_APP_MAIN
{
    std::string str = "    Lots of filler space    ";
    boost::algorithm::trim(str);
    std::cout<<"\""<< str <<"\""<<std::endl;
}
```

the following error is resulting:

```
----- Core ( GCC DEBUG SHARED DEBUG_FULL BLITZ POSIX LINUX ) (1 / 2)
----- BoostStringTest ( MAIN GCC DEBUG SHARED DEBUG_FULL BLITZ POSIX LINUX ) (2 / 2)
BoostStringTest.cpp
In file included from /home/ptkacz/upp/uppsrc/Core/i18n.h:17:0,
                 from /home/ptkacz/upp/uppsrc/Core/Core.h:337,
                 from /home/ptkacz/MyApps/BoostStringTest/BoostStringTest.cpp:6:
/usr/include/boost/core/ref.hpp: In constructor
'boost::reference_wrapper<T>::reference_wrapper(T&)':
/home/ptkacz/upp/uppsrc/Core/t_.h:9:24: error: class 'boost::reference_wrapper<T>' does not
have any field named 't_GetLngString'
 #define t_(x)          t_GetLngString(x)
                        ^
BoostStringTest: 1 file(s) built in (0:01.55), 1551 msecs / file, duration = 1551 msecs

There were errors. (0:01.55)
```

I do know that U++ has it's only string utility functions, but was attempting to make use of code brought in from another application that I'm working on. Independently of any existing code, I wrote a simple C++ application in Code::Blocks, the code compiles and runs as expected.

Peter

Well, this looks like simple nameclash. Try

#undef t_

after #include <Core/Core.h>.

t_ is used to designate localized strings that are subject to language translation.

Maybe it does not have to be macro, then this would be solved by namespaces. Will try...

Mirek

Subject: Re: U++ does not appear to like playing nice with the Boost algorithm string library?
Posted by ptkacz on Sat, 04 Apr 2020 17:51:43 GMT
View Forum Message <> Reply to Message

Thanks Mirek, I just tried that and it worked! To get around that issue, I did update my code to make use of the trim ends function that U++ has available. I do think undefining things is probably not a recommended solution, I couldn't possibly think of what else could go wrong.

Subject: Re: U++ does not appear to like playing nice with the Boost algorithm string library?
Posted by mirek on Sun, 05 Apr 2020 08:15:23 GMT
View Forum Message <> Reply to Message

ptkacz wrote on Sat, 04 April 2020 19:51Thanks Mirek, I just tried that and it worked! To get around that issue, I did update my code to make use of the trim ends function that U++ has available. I do think undefining things is probably not a recommended solution, I couldn't possibly think of what else could go wrong.

I have actually replaced that macro with inline function in the trunk, so very likely it will work without that #undef witht the trunk for now.

(That said, I am not yet sure whether this change is OK, it is possible I will be forced to remove yet to maintain compatibility).

Mirek
