Subject: Linking in the std library
Posted by awksed on Wed, 31 Jul 2024 20:23:50 GMT
View Forum Message <> Reply to Message

To use the chrono library (which needs C++ 20) in older projects I built a static library in VS 2022.

Linking in my static lib to a U++ console project I get the following error (and other similar):

unresolved external symbol "public: static unsigned int const * const std::_General_precision_tables_2<float>::_Ordinary_X_table "

This is probably related to the use of std::format().

How can I link in the appropriate library or source?

Windows 10
U++ 17045
MSVS22x64

Thanks.

Subject: Re: Linking in the std library
Posted by koldo on Thu, 01 Aug 2024 06:16:42 GMT
View Forum Message <> Reply to Message

Sorry Awksed, are you referring to std::chrono?

Subject: Re: Linking in the std library
Posted by awksed on Thu, 01 Aug 2024 07:44:26 GMT
View Forum Message <> Reply to Message

Hi Iñak,

Yes it's std:chrono.

Subject: Re: Linking in the std library
Posted by koldo on Sun, 04 Aug 2024 15:08:46 GMT
View Forum Message <> Reply to Message

Hi Awksed

I have tried the code in the second answer here (enclosed below) and it has worked in CLANG (W10 and W11) and MSVC (W10):

```
#include <iostream>
#include <chrono>
#include <thread>

int main()
{
    using namespace std::literals;
    namespace chrono = std::chrono;
    using clock_type = chrono::high_resolution_clock;

    auto start = clock_type::now();

    for(;;) {
        auto first = clock_type::now();

        // note use of literal - this is c++14
        std::this_thread::sleep_for(500ms);

        // c++11 would be this:
        //        std::this_thread::sleep_for(chrono::milliseconds(500));

        auto last = clock_type::now();
        auto interval = last - first;
        auto total = last - start;

        // integer cast
        std::cout << "we just slept for " <<
chrono::duration_cast<chrono::milliseconds>(interval).count() << "ms\n";

        // another integer cast
        std::cout << "also known as " <<
chrono::duration_cast<chrono::nanoseconds>(interval).count() << "ns\n";

        // floating point cast
        using seconds_fp = chrono::duration<double, chrono::seconds::period>;
        std::cout << "which is " << chrono::duration_cast<seconds_fp>(interval).count() << "
seconds\n";

        std::cout << "  total time wasted: " <<
chrono::duration_cast<chrono::milliseconds>(total).count() << "ms\n";
        std::cout << "         in seconds: " << chrono::duration_cast<seconds_fp>(total).count() <<
"s\n";


        std::cout << std::endl;
    }
    return 0;
}
```

Subject: Re: Linking in the std library
Posted by awksed on Mon, 05 Aug 2024 16:46:57 GMT
View Forum Message <> Reply to Message

Hi Iñak,

My problem is with a static library. An exe will build ok as in your reply.

I fixed the problem by eliminating str:formt() from the library.

Many thanks for your reply.

Subject: Re: Linking in the std library
Posted by koldo on Tue, 06 Aug 2024 12:37:29 GMT
View Forum Message <> Reply to Message

Perfect.

In your query I didn't realise that you had compiled your library, which uses std::chrono, as a static library, not that you had done a static compilation.
It's been so long since I've done it, I hadn't thought about it.