

# How to get MDEntryTime with nanosecond precision using Spectra API

MOEX Market Data Multicast FIX/FAST Platform uses custom UTCTimeOnly, UTCTimestamp format in the FAST messages.

Please refer to the official documentation for more details: <http://ftp.moex.com/pub/FAST/Spectra> e.g. [http://ftp.moex.com/pub/FAST/Spectra/prod/docs/spectra\\_fastgate\\_en.pdf](http://ftp.moex.com/pub/FAST/Spectra/prod/docs/spectra_fastgate_en.pdf)

## Example of handling time from MDEntryTime

### Application.cpp

```
void ConvertTimeToStr(System::u64 entry, std::string* strTime)
{
    static std::size_t const TIME_LENGTH = sizeof( "HHMMSSssssssss" ) - 1;

    std::stringstream stream;
    stream << entry; // convert u64 to std::string.
    std::string time(stream.str());
    if (TIME_LENGTH > time.size())
        time.insert(0, TIME_LENGTH - time.size(), '0');

    strTime->assign(time, 0, 2); // HH
    strTime->append( ":" );
    strTime->append(time, 2, 2); // MM
    strTime->append( ":" );
    strTime->append(time, 4, 2); // SS
    strTime->append( "." );
    strTime->append(time, 6, 9); // ssssssss
}

bool InstrumentHandler::onIncrement( Spectra::SpectraSubscriptionItem const& subsItem, Engine::FIXMessage
const* const* msgs, size_t incrementsCount )
{
    // ...
    for( ; i < incrementsCount; ++i ) {
        const Engine::FIXGroup& gr = msgs[i].getAsGroup( FIXFields::NoMDEntries );
        for( int j = 0; j < gr.size(); ++j ) {
            const Engine::TagValue& entry = *gr.getEntry(j);
            std::string timeNs;
            if (entry.hasValue(FIXFields::MDEntryTime))
                ConvertTimeToStr(entry.getAsUInt64(FIXFields::MDEntryTime), &timeNs);
            // ..
        }
    }
    // ...
}
```