MTConnect Testing Gadgets

Prepared by: John Michaloski

Prepared on: March 2, 2012

# Table of Contents

[Table of Contents 2](#_Toc318448450)

[Revision History 3](#_Toc318448451)

[Copyright 3](#_Toc318448452)

[Disclaimer 3](#_Toc318448453)

[Terminology 4](#_Toc318448454)

[Background 5](#_Toc318448455)

[SHDRPlayback 5](#_Toc318448456)

[Shdr Backtground: 6](#_Toc318448457)

[Configuration: 6](#_Toc318448458)

[Build Prequisites: 7](#_Toc318448459)

[Comments or Questions 7](#_Toc318448460)

# Revision History

**Date Description Author Version**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Version** |
| 3/1/2012 | Initial Gadget Set | John Michaloski | 1.0 |

# 

# Copyright

*Software produced by the National Institute of Standards and Technology (NIST), an agency of the U.S. government and by statute is not subject to copyright in the United States. Recipients of this software assume all responsibility associated with its operation, modification, maintenance, and subsequent redistribution.*

# 

# Disclaimer

*No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied. Certain commercial equipment, instruments, or materials are identified in this report in order to facilitate understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.*

# Terminology

**Adapter** An optional software component that connects the Agent to the Device. W

**Agent** A process that implements the MTConnect specification, acting as an interface

to the device.

**Current** A snapshot request to the Agent to retrieve the current values of all the data items specified in the path parameter. If no path parameter is given, then the values for all components are provided.

**Device** A piece of equipment capable of performing an operation. A device is composed of a set of components that provide data to the application. The device is a separate entity with at least one Controller managing its operation.

**HTTP Hyper-Text Transport Protocol** The protocol used by all web browsers and web applications.

**Sample** A sample is a data point for continuous data items, that is, the value of a data item at a point in time.

# Background

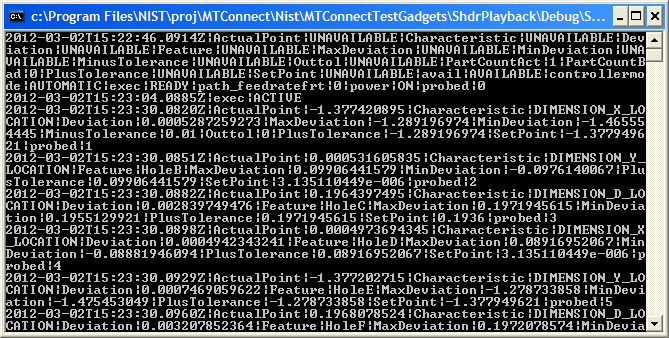
MTConnect is a new standard sponsored by the Association for Manufacturing Technology (who run IMTS) developed to facilitate the exchange of data on the manufacturing floor. The MTConnect open specification provides for cost effective data acquisition on the manufacturing floor for machine tools and related devices.

This document describes some software tools (gadgets) to help in implementing and validating MTConnect software.

* SHDR Playback is a tool that reads files containing SHDR output and then transmits the strings based on the timing of the Timestamps in the file. Override capabilities to speed up or slow down the playback is possible.

# SHDRPlayback

SHDR Playback is a tool that reads files containing SHDR output and then transmits the strings based on the timing of the Timestamps in the file. Override capabilities to speed up or slow down the playback is possible. SHDR Playback is either a Windows application (which will not be discussed) or a Console application, like the screen shot below.



To be a console application, SHDRPlayback expects a ShdrPlayback.ini file in the same directory as the executable.

This SHDR output could theoretically be hand coded, but telnet capture and saving is clearly an easier mechanism. Telnet SHDR generates only a LF at the end of line. However, the SHDR playback ***expects well-formed SHDR where the LF has been formatted into CR/LF***. This makes is easier for humans to read the file, which is important if you want to tweak the timing of the SHDR as often the real world is a lot slower than a MTConnect software developer/tester would like to endure(even with 5x override speedup). More robustness in the SHDR parsing may be developed in later versions.

### Shdr Backtground:

The SHDR protocol is not part of the MTConnect® specification, and readers are advised to read “MTConnect® Standard SHDR Protocol Companion Specification” for more details. Briefly, the SHDR protocol “was designed to be as simple as possible”. SHDR provides a TCP socket stream such that one or more Agents can connect to the SHDR socket and listen to the data stream, with behavior similarl to telnet. The communication format is a pipe ‘|’ delimited data stream that begins with a timestamp and then follows with a timestamp in ISO 8601 date time format with optional decimal places. The time is required to be in UTC and therefore must have a trailing Z to indicate that no time zone (+0) is being used. SHDR handles Samples, Events, Conditions, Assets, and Alarms (deprecated).

Below is an EBNF representation of the SHDR including the new multiline asset implementation:

<SHDR> ::= UTCDATE "|" <StatementList>

<StatementList> ::= <Statement> | <Statement> EOL <StatementList>

<Statement> ::= <SimpleStatement> | < MultilineStatement >

<SimpleStatement> ::= <Tag> "|" <Value> { "|" <Value>}\*

<MultilineStatement> ::= "@" <Tag> "@" "|" ID "|" --multiline—{A-Z}+ .\* --multiline—{A-Z}+  
<Tag> ::= <Sample> | <Event> | <Condition>

<Sample> ::= ID

<Event> ::= ID

<Condition> ::= ID “|” <Level> “|” <NativeCode> “|” <NativeSeverity> “|” <Qualifier> “|” <Description>

<Level> ::= UNVAILABLE | NORMAL | WARNING | FAULT

<NativeCode> ::= NUMBER | TEXT

<NativeSeverity> ::= TEXT

<Qualifier> :== HIGH, LOW, …

<Description> ::= TEXT

Here is a sample SHDR string for transmitting s data, and the line must be terminated with a line feed (\n) which is ASCII character 10 or 0xA.

2010-09-29T23:59:33.460470Z|Xact|1.4198908806\n

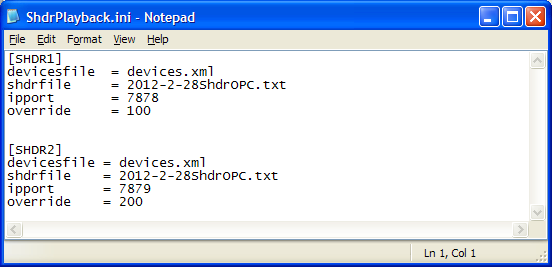
### Configuration:

The ShdrPlayback executable is both an Windows and a Console application. If a “Playback.ini” file is in the same directory as the executable, then the application runs as a Console application.

This table summarizes the configuration parameters in the ini file:

|  |  |
| --- | --- |
| [xxx] | Contains a section name with the following list of parameters. For each section, a new SHDR generator will be spawned. |
| devicesfile=devices.xml | The name of a devices file in which to extract the samples and event names/ids to send to any connecting listener. If a full path is not specified, the Exe path is preprended to the devicesfile file path. |
| shdrfile= shdrfile.txt | Provides a text file containing the SHDR to play back to listeners. Uses the internal timestamps to determine the wait between updates. If a full path is not specified, the Exe path is preprended to the shdr file path. |
| override=100 | The amount of override speedup for this playback, in percentage. 50% is slower, 200% is twice as fast, etc. |
| ipport = 7878 | Configures the TCP socket port for listeners to connect to. |

Below is a snapshot of sample configuration file:



### Build Prequisites:

The SHDR Playback is a Visual C++ implementation. It depends on the following libraries:

* WTL – You will need to download and unzip WTL and place it into the visual studio C++ include path. (Used for the graphical Windows interface).
* BOOST – you will need to download C++ boost libraries, as it uses ASIO library to simplify the telnet implementation.

# Comments or Questions

Please contact:

John Michaloski

Intelligent Systems Division

National Institute of Standards and Technology

[John.Michaloski@nist.gov](mailto:dugenske@gatech.edu)

+1 301-975-3458