

NIST Advanced Manufacturing Series 200-2

**Connecting, Deploying, and Using the
Smart Manufacturing Systems Test Bed**

Moneer Helu
Thomas Hedberg, Jr.

This publication is available free of charge from:
<https://doi.org/10.6028/NIST.AMS.200-2>

NIST
National Institute of
Standards and Technology
U.S. Department of Commerce

NIST Advanced Manufacturing Series 200-2

Connecting, Deploying, and Using the Smart Manufacturing Systems Test Bed

Moneer Helu
Thomas Hedberg, Jr.
*Systems Integration Division
Engineering Laboratory*

This publication is available free of charge from:
<https://doi.org/10.6028/NIST.AMS.200-2>

July 2020



U.S. Department of Commerce
Wilbur L. Ross, Jr., Secretary

National Institute of Standards and Technology
Walter Copan, NIST Director and Undersecretary of Commerce for Standards and Technology

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

**National Institute of Standards and Technology
Advanced Manufacturing Series 200-2
Natl. Inst. Stand. Technol. Adv. Man. Ser. 200-2, 53 pages (July 2020)**

**This publication is available free of charge from:
<https://doi.org/10.6028/NIST.AMS.200-2>**

Abstract

This report describes how different smart manufacturing standards and technologies, such as MTConnect, were integrated in the National Institute of Standards and Technology (NIST) Smart Manufacturing Systems (SMS) Test Bed to support the collection, curation, and re-use of manufacturing data. The intended audience of this report includes researchers, technical-assessment personnel, and end-users who would like to collect data from manufacturing equipment on the shop floor to improve the performance of production systems. Included in this document is basic guidance to replicate the NIST SMS Test Bed in research and production environments, including the protocols, information systems, and automation that were required.

Key words

Data collection; Data curation; Data management; MTConnect; Network design; Smart Manufacturing; Test Bed.

Table of Contents

1	Introduction	1
1.1	Purpose	1
1.2	Implementation Approach and Scope	1
1.3	Disclaimer	3
1.4	Overview	3
2	MTConnect Implementation	3
2.1	Background	3
2.2	General Approach	5
3	Implementation in the Test Bed	6
4	Automation	10
5	Summary	12
	References	13
	Appendix A: Devices Extensible Markup Language (XML) File	14
	Appendix B: Agent Configuration File	35
	Appendix C: MTConnect Agent Service	37
	Appendix D: Data Collection Service	37
	Appendix E: Data Re-Start and Archive bash Script	37
	Appendix F: Adapter Simulator Service	38
	Appendix G: Queryable Database Repository (QDR) Bulk Data Import Generator	38
	Appendix H: QDR Bulk Data Upload	43

List of Tables

Table 1	Available machine tools in the NIST Fabrication Technology Office (FTO) that currently provide data in the Test Bed (based on Ref. [1])	2
---------	---	---

List of Figures

Fig. 1	Basic architecture of the traditional implementation of MTConnect; the standard focuses only on the output from the Agent to a client application (based on Ref. [2])	4
Fig. 2	Overview of manufacturing equipment on the shop floor of the Test Bed divided into three “cells” for network connectivity; green indicates a machine that is available for data collection	8
Fig. 3	Overview of the network diagram that describes the infrastructure deployed in the Test Bed	9

Fig. 4 Overview of movement of data from devices to client applications mapped to each of the four layers of the Test Bed reference architecture (see Ref. [3]) 10

Acronyms

ANSI American National Standards Institute.

AP access point.

API application programming interface.

FTO Fabrication Technology Office.

HTTP Hypertext Transfer Protocol.

IP Internet Protocol.

MES manufacturing execution system.

MQTT MQ Telemetry Transport.

NIST National Institute of Standards and Technology.

QDR Queryable Database Repository.

REST Representational State Transfer.

SMS Smart Manufacturing Systems.

SRS software requirements specification.

UA Unified Architecture.

URL Uniform Resource Locator.

UTC Coordinated Universal Time.

VDS Volatile Data Stream.

XML Extensible Markup Language.

XSD XML Schema Definition.

1. Introduction

1.1 Purpose

The purpose of this document is to describe how different standards and technologies were integrated in the National Institute of Standards and Technology (NIST) Smart Manufacturing Systems (SMS) Test Bed (labeled “Test Bed” throughout the remainder of this document) to support the collection, curation, and re-use of manufacturing data. The intended audience of this report includes researchers, technical-assessment personnel, and end-users who would like to collect data from manufacturing equipment on the shop floor to improve the performance of production systems. Included in this document is basic guidance to replicate the Test Bed in research and production environments, including the protocols, information systems, and automation that were required.

1.2 Implementation Approach and Scope

The implementation of the Test Bed described in this document provides one example of a solution created using the recommendations provided by Ref. [4]:

Citation: Helu M, Hedberg Jr T (2020) Recommendations for collecting, curating, and re-using manufacturing data. National Institute of Standards and Technology, Report NIST AMS 300-11. doi: 10.6028/NIST.AMS.300-11

As Ref [4] describes, there are four steps that should be followed to collect, curate, and re-use manufacturing data. First, we must define our use case so that we clearly understand our requirements. We document our use case and general requirements in two reports: (1) a technical article that introduces the concept of the Test Bed (Ref. [5]), and (2) a report that describes the overall design and configuration of the Test Bed (Ref. [1]). These documents are cited respectively below:

Citation: Helu M, Hedberg Jr T (2015) Enabling Smart manufacturing research and development using a product lifecycle test bed. *Procedia Manufacturing*, 1:86-97. doi: 10.1016/j.promfg.2015.09.066

Citation: Hedberg Jr T, Helu M (2017) Design and configuration of the smart manufacturing systems test bed. National Institute of Standards and Technology, Report NIST AMS 200-1. doi: 10.6028/NIST.AMS.200-1

Our use case can be broadly described as distributing manufacturing data from the Test Bed via a web-based service with two components [1]: (1) a Volatile Data Stream (VDS) of time-synchronized manufacturing data [6], and (2) a Queryable Database Repository (QDR) for querying a database containing archived manufacturing data [7]. Identifying supported devices that provide this manufacturing data is the second step of Ref. [4]. The Test Bed is implemented in partnership with the NIST Fabrication Technology Of-

Table 1. Available machine tools in the NIST FTO that currently provide data in the Test Bed (based on Ref. [1])

Machine Tool	Manufacturer	Model	Type
GFAgie01	Agie Charmilles	Micron HPM600U	5-axis milling center
Mazak01	Mazak	Integrex 100-IV	Mill-turn center
Mazak03	Mazak	QuickTurn Nexus 300	Turning center
Hurco01	Hurco	VMX 24	3-axis milling center
Hurco02	Hurco	VMX 24	3-axis milling center
Hurco03	Hurco	VMX 24	3-axis milling center
Hurco04	Hurco	VMX 24	3-axis milling center
Hurco06	Hurco	VMX 64	3-axis milling center

vice (FTO), which operates like a contract manufacturer. Table 1 provides a subset of the machine tools available from the NIST FTO that are currently providing data in the Test Bed.

Continuing from Ref. [4], we require semantically-defined manufacturing data from the devices described in Table 1. We select MTConnect [8] as the primary standard to define these semantics since it provides one of the only standardized domain models for manufacturing equipment. Section 2 provides more information about the MTConnect standard and the different implementation implications of this decision.

The third step in Ref. [4] is to evaluate our network infrastructure and establish requirements to access and manage the data from our selected devices. These requirements and the resulting specifications for the Test bed are documented in a software requirements specification (SRS) document (Ref. [9]) that can be cited as follows:

Citation: Hedberg Jr T, Helu M, Newrock M (2017) Software requirements specification to distribute manufacturing data. National Institute of Standards and Technology, Report NIST AMS 300-2. doi: 10.6028/NIST.AMS.300-2

The specifications and requirements described in Ref. [9] drive the design of the reference four-tiered architecture of the Test Bed. This reference architecture manages the data generated by the Test Bed and provides segregated access to internal and external client applications to protect any sensitive and/or privileged information [3]. The architecture also enables the fusion of manufacturing and other product lifecycle data. The citation for this architecture document (Ref [3]) is:

Citation: Helu M, Hedberg Jr T, Barnard Feeney A (2017) Reference architecture to integrate heterogeneous manufacturing systems for the digital thread. CIRP Journal of Manufacturing Science and Technology doi: 10.1016/j.cirpj.2017.04.002

The requirements, specifications, and architecture documented in the previously cited

reports guide the implementation of the Test Bed that is described in this document. Once the design of our implementation is finalized, we can then proceed to the fourth and final step of Ref. [4]: execute the relevant integration activities. We briefly discuss the execution of our implementation later in this document.

1.3 Disclaimer

This work is an official contribution of NIST and not subject to copyright in the United States. Certain commercial systems are identified in this paper. Such identification does not imply recommendation or endorsement by NIST. Nor does it imply that the products identified are necessarily the best available for the purpose.

1.4 Overview

The remainder of this document describes the specifics of the implementation of data standards and connectivity solutions to collect manufacturing data and the automation necessary to distribute this data from the Test Bed to different client applications.

2. MTConnect Implementation

The primary manufacturing data standard used in the Test Bed is *MTConnect*. MTConnect is developed by the MTConnect Institute, which is an American National Standards Institute (ANSI)-accredited standards-development organization. The latest ANSI-accredited version of MTConnect was released in 2018 as ANSI/MTC1.4-2018 [8].

2.1 Background

MTConnect [8] provides a standard domain model for manufacturing equipment that enables the creation of structured, contextualized data for client applications [2]. It defines a semantic vocabulary and information model with no proprietary format that is designed to be extensible. In this way, MTConnect enables semantic interoperability between different manufacturing systems and client applications.

The standard is designed to be read-only and to integrate with other standards and protocols [2]. A basic minimum infrastructure is defined that must be implemented to collect data from manufacturing equipment using a software interface for compliance with the standard. However, it is important to note that the implementation of the standard provides some flexibility so that the right set of solutions may be leveraged by the end user. For example, MTConnect may be used with a machine-to-machine protocol, such as OPC Unified Architecture (UA) [10]; a companion specification has been released to support such a solution [11]). Similarly, MTConnect may be used with pub-sub protocols, such as MQ Telemetry Transport (MQTT) [12]; one recent example of such an implementation is Ref. [13].

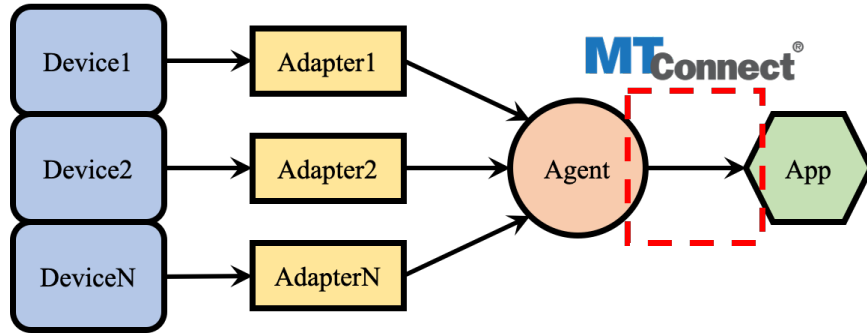


Fig. 1. Basic architecture of the traditional implementation of MTConnect; the standard focuses only on the output from the Agent to a client application (based on Ref. [2])

Figure 1 shows the basic architecture of a traditional implementation of MTConnect [2]. This implementation uses an *Adapter*, which is an optional piece of software or hardware that interfaces with the equipment application programming interface (API). The Adapter works with an *Agent*, which provides the software interface for client applications defined by the standard, to translate data from proprietary formats to the MTConnect semantic vocabulary, types, and units. The Agent acts as a special purpose Hypertext Transfer Protocol (HTTP) server that provides a Representational State Transfer (REST)ful interface. Through this interface, the Agent creates and publishes MTConnect response documents based on requests from client applications.

There are four types of MTConnect response documents, which each represent an information model defined by the standard [2, 8]: *MTConnectDevices*, *MTConnectStreams*, *MTConnectAssets*, and *MTConnectErrors*:

- **MTConnectDevices** contains the metadata (i.e., the meaning and logical structure) of a device
- **MTConnectStreams** contains the time-series data values reported by a device in a flattened hierarchy with consistent units and timestamps
- **MTConnectAssets** contains a key-value store that can be used for components that are not integral to the device
- **MTConnectErrors** contains protocol-related errors

Within the response documents, there are three categories of data items that can be collected [2, 8]:

- **Events** represent the state of a specific data item, discrete message, or count
- **Samples** represent continuous variables that are sampled at a specific rate
- **Conditions** represent the health of a component

It is important to note that the Agent only reports changes in data items [8]. Also, the

availability (or unavailability) of data is the only data item that a device must report for compliance with the standard.

To access data from the Agent through a response document, there are four requests that can be made [2, 8]:

- *probe* returns an MTConnectDevices response document
- *current* returns an MTConnectStreams response document at the most recent point in time
- *sample* returns an MTConnectStreams response document for a specified number of entries
- *asset* returns an MTConnectAssets response document for the requested asset

XPath can be used to filter the data provided by the Agent through a response document [2]. To use XPath, a path argument need only be appended to the request Uniform Resource Locator (URL). For example, to request only those data items associated with a specific component, such as axes, can be accomplished by appending `//Axes` to the request URL. Similarly, only those data items for a specific device may be requested by appending `/[device]` before the request URL. So, to request the most recent data for all axes components in a device *Machine01* with an Agent that can be accessed at the Internet Protocol (IP) address 192.168.1.1:5000, we would use the following URL:

<https://192.168.1.1:5000/Machine01/current//Axes>

2.2 General Approach

Implementing MTConnect to collect data from a piece of manufacturing equipment requires at least three items to execute: Agent, Device Extensible Markup Language (XML) file, and Agent configuration file. An open-source MTConnect C++ Reference Agent (called “Reference Agent” for the remainder of this document) can be downloaded and built from <https://github.com/mtconnect/cppagent>. Instructions on installing the Reference Agent can be found on the MTConnect Users Portal at http://mtcup.org/wiki/C%2B%2B_Agent. Alternatively, third-parties may provide an MTConnect Agent that can be used as well.

The Device XML file defines the information model for the piece of equipment in question, and thus it is essentially identical to an MTConnectDevice response document. The Agent uses the Devices XML file to organize the data provided by an Adapter into the different MTConnect response documents that may be requested by a client application. Appendix A provides the Device XML file created for the Test Bed as an example.

The Agent configuration file provides additional information to the Agent so that it can identify different adapter inputs and locate schemas and styles as needed. Appendix B provides an example of an Agent configuration file that can be used with a hypothetical device named *Machine01* that has an Adapter providing data through 192.168.1.1:7878.

In addition to the three items that have been previously described, it is often useful to create an MTConnect Agent service when implementing MTConnect in a Linux-based environment. Such a service can start or stop the Agent should the need arise (e.g., for any required maintenance). Appendix C provides an example of an MTConnect Agent service.

When implementing MTConnect, it is important to note that not every device provides the same data items nor should every device be modeled identically in the Device XML file. The best approach to structuring an implementation will depend on the specific use case and devices of interest. Furthermore, one should also keep in mind that not every device may offer data at an equivalent or consistent rate. These characteristics are entirely dependent on the implementation of the standard and are not in any way defined by the MTConnect documentation.

3. Implementation in the Test Bed

We first upgraded each piece of equipment in the Test Bed (see Section 1.2) so that it complied with the MTConnect standard by deploying appropriate Adapters on the machine tool controller or a separate computer. These Adapters were obtained from the machine tool vendor or a third-party solution provider. (Please note that the controller vendor may also provide an Adapter for their products.) Typically, these Adapters were bundled with a Reference Agent and Devices XML file. We chose to modify the Devices XML file in some instances to reflect the data available from a specific piece of equipment in the Test Bed. This modification generally involved removing additional data items and updating the controlled vocabulary to reflect the latest revision of MTConnect.

Figure 2 represents the layout of the equipment in the Test Bed on the shop floor. A goal of our implementation was to use wireless connections to the extent possible for ease of integration and added flexibility: e.g., any changes in the shop-floor layout would not require the additional physical effort of re-laying cable. However, using wireless connections throughout the entire Test Bed network proved challenging due to potential interference from several sources, including:

- Walls (e.g., Cell #2 in Fig. 2 exists in an area separate from the main shop floor)
- Metal in columns on the shop floor
- Metal in the equipment itself
- Rotating components in the equipment (e.g., spindles)
- Existing wireless networks for business operations

Bandwidth was also a concern when using wireless connections in the Test bed, particularly because we planned to add ad-hoc sensors, including relatively high-frequency data sources, such as accelerometers, over time. To balance these concerns, we used a “hybrid” approach where we divided the shop floor into three cells. All of the devices – machine tools controllers and ad-hoc sensors – in a cell were wired to a central point, and

communications from that central point to points outside the cell were wireless.

Figure 3 provides an overview of the resulting network architecture. Within each cell, we wired all devices to a network switch using Category 6 (Cat6) cable. We then wired the network switch to a wireless access point (AP). To compensate for the interference created by the wall that separates Cell #2 from the main floor, we used a relatively long cable run through the wall so that the wireless AP could be placed in the main shop-floor area. We installed all wireless APs approximately six meters above the shop floor so that the wireless signal can travel above the machine tools, which should minimize any potential interference. Furthermore, we selected a wireless band to avoid potential interference from the existing wireless network for business operations; this choice was further validated by a wireless interference map that our colleagues measured for the shop floor.

Each wireless AP was connected to a router that enables the Data Aggregator to manage and collect data throughout the Test Bed. We selected a Linux-based desktop computer tower for the Data Aggregator and used it to host much of the automation scripts that will be described in Sec. 4. The router also provided the gateway to the Research Equipment Network, which is a segmented internal network with strict firewall policies to protect the devices in the Test Bed. Such an architecture was needed because many – if not all – machine tool controllers run outdated operating systems that present a security risk. To ensure security, the network only allows data to be pushed from the Data Aggregator to the Organizational Network.

An important architectural concern when deploying MTConnect is whether to use one or many Agents. We decided on another hybrid approach where each machine would have an Agent that could provide daily data files in a pipe-delimited text format that mimics the typical output of an Adapter (called “SHDR” in MTConnect). This pipe-delimited format is structured as follows:

$$YYYY-MM-DDTHH:MM:SS.SSSZ|DataTag|DataValue,$$

where *YYYY-MM-DD* represents the date in year-month-day format, *HH:MM:SS.SSSZ* represents the time in hour:minute:second format at Coordinated Universal Time (UTC), *DataTag* represents the tag or label for the data item being collected, and *DataValue* represents the value of the data item being collected with the appropriate units (if appropriate) as defined by the MTConnect standard.

The collected data files were then “replayed” as Adapter files for one larger Agent to cover the entire Test Bed, which was hosted by a separate Linux-based server on the Organizational Network. This larger Agent then provided data for client application needs as shown in Fig. 4. Using this solution, we were able to maintain one interface for client applications while still providing some flexibility when maintaining the Test Bed since we would not have to shutdown the entire data stream to service a subset of devices.

An important aspect of the data pipeline shown in Fig. 4 is that the Test Bed hosts private clients through the Linux-based server on the Organization Network that was described previously and public clients through a cloud-based instance (please see <https://smstestbed.nist.gov/> for more information). The larger Test Bed Agent on the Linux-based server

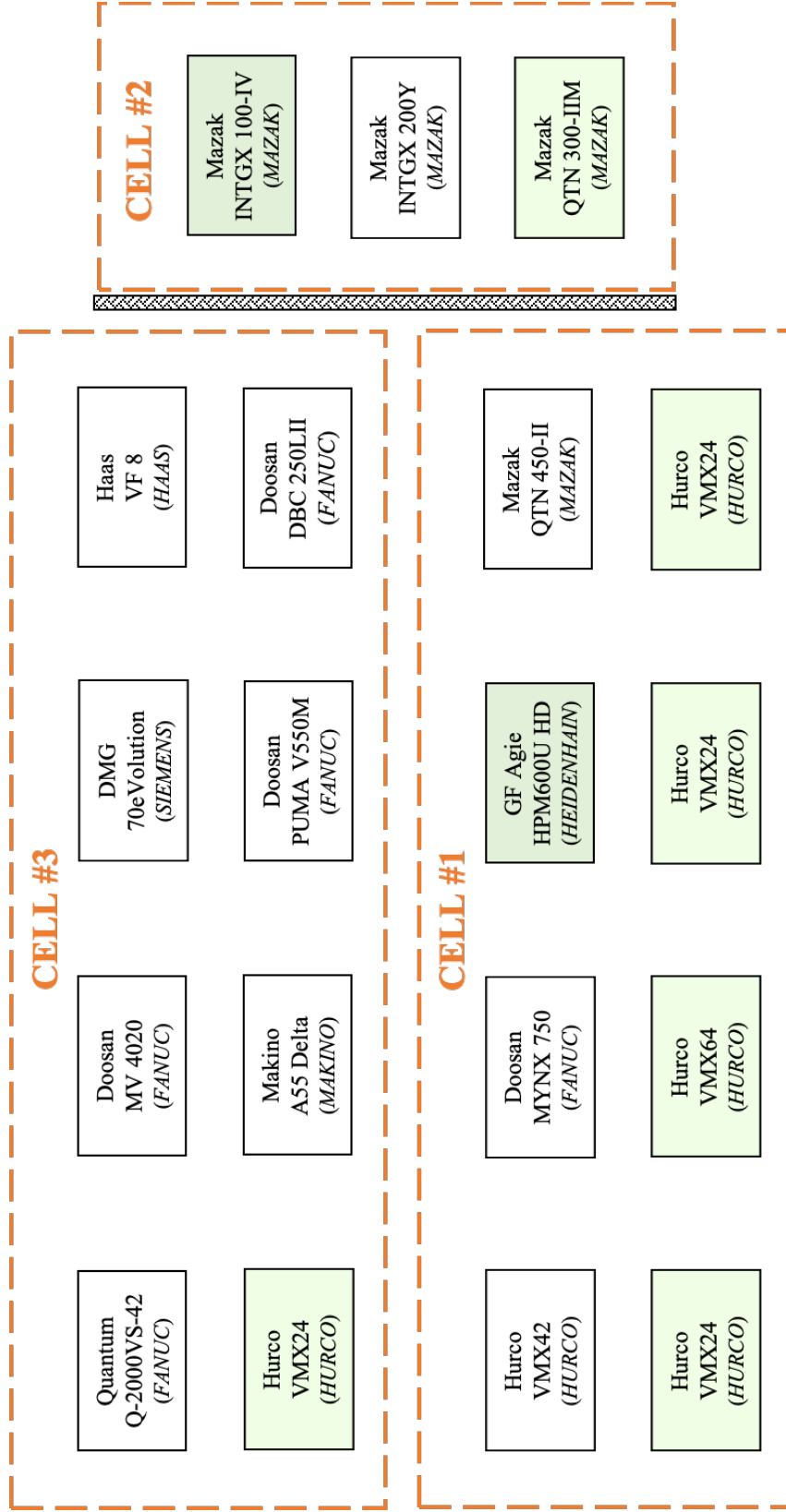


Fig. 2. Overview of manufacturing equipment on the shop floor of the Test Bed divided into three “cells” for network connectivity; green indicates a machine that is available for data collection

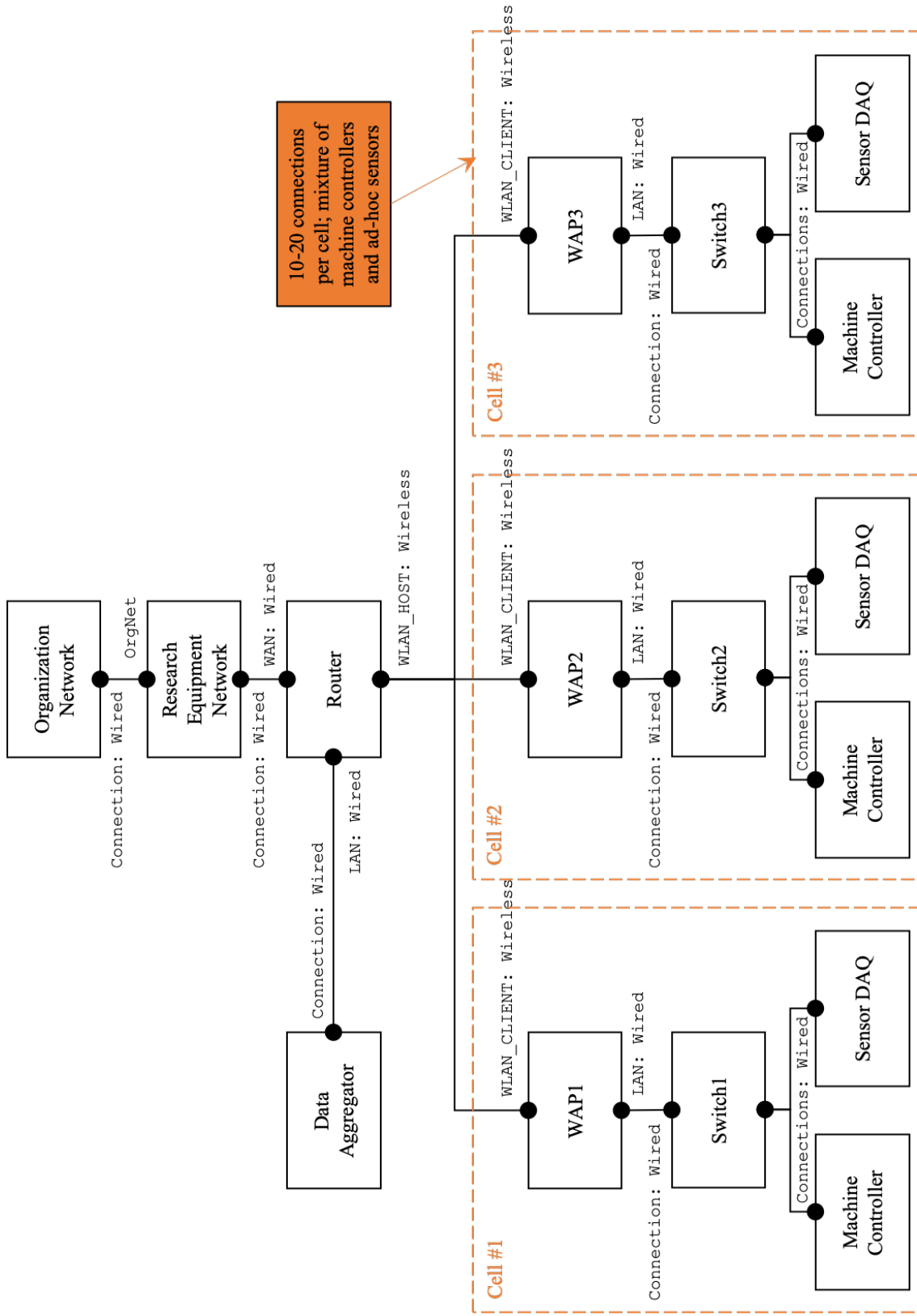


Fig. 3. Overview of the network diagram that describes the infrastructure deployed in the Test Bed

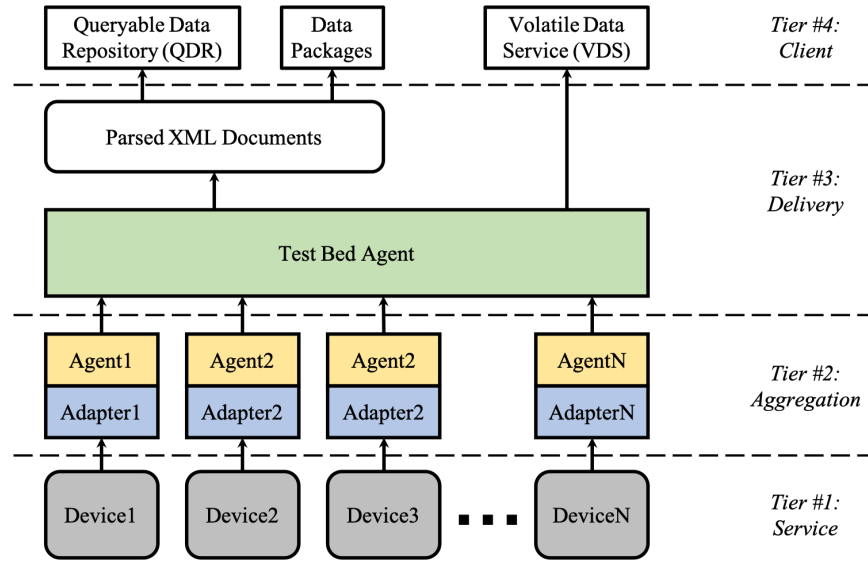


Fig. 4. Overview of movement of data from devices to client applications mapped to each of the four layers of the Test Bed reference architecture (see Ref. [3])

generates parsed XML documents for the QDR and data packages for both private and public clients. However, the daily data files generated from each machine are moved to the cloud-based instance to be replayed by an Agent there for the public VDS. So, while Fig. 4 represents the overall movement of data from the Test Bed, please note that there is a bifurcation that occurs in Tier #3 between the private and public VDS.

4. Automation

The Data Aggregator shown in Fig. 3 collects data from each device in the Test Bed and prepares this data for delivery by either the internal Linux-based server or external cloud-based instance to client applications. To accomplish these tasks, the Data Aggregator, internal Linux-based server, and external cloud-based instance host different automation files to remove the need for human intervention in the flow of data through the Test Bed. The general steps in this process include:

1. The Data Aggregator collects 24 hours of data and stores this data in separate files for each piece of equipment
2. The data collection services are re-started to begin the next 24-hour period of collection at a pre-selected time outside of the typical operations on the shop floor
3. The 24-hour data file for each piece of equipment is archived on the Data Aggregator and synced to the internal Linux-based server
4. The Linux-based server replays each data file – using the data file as an Adapter

simulator – for the private VDS and generates the parsed XML documents for the QDR and data packages

5. After a total holding period of seven days from data collection, the internal Linux-based server syncs each data file to a repository for further archiving; the cloud-based instance also captures the data files from this repository to be replayed again for the public VDS

We created a service on the Data Aggregator for each piece of equipment in Test Bed to collect and save data to a text file in a pipe-delimited format as described in Sec. 3. Appendix D provides an example configuration file for the data collection service for a hypothetical *Machine01* with an MTConnect Agent that can be accessed at the IP address [192.168.1.1:5000](#); this configuration file is named `dump_agent_Machine01.conf`. The data collection service used an open-source tool created in Ruby named `dump_agent.rb` that is provided as part of the Reference Agent.

To re-start data collection and archive data on the Data Aggregator, we created a bash script named `generic_recorder.sh`. Appendix E provides an example of this bash script. We then programmed a set of cron jobs in the Data Aggregator to execute `generic_recorder.sh` at the pre-selected time and sync the data archive on the Data Aggregator to the data archive on the internal Linux-based server.

To replay the collected data files using these files as an Adapter simulator, we used another open-source tool created in Ruby named `run_scenario.rb` that is also provided as part of the Reference Agent. We created a service for each piece of equipment to manage this process. Appendix F provides an example configuration file for this Adapter simulation service for the hypothetical *Machine01* named `mtc_adapter_Machine01.conf`.

To generate parsed XML documents for the QDR and data packages, we created a Python script named `data_bulkQDRinput.py`. This script essentially replays the collected data files through an MTConnect Agent to output XML files for input into the QDR. Appendix G provides a generalized example of `data_bulkQDRinput.py`.

The QDR uses a REST API for ingesting data into the database. To upload the parsed XML documents through the QDR API from `data_bulkQDRinput.py`, we created a Python script named `QDR_bulkUpload.py`. This script will upload all XML files located in a single directory. Appendix H provides an generalized example of `QDR_bulkUpload.py`.

One important note when managing all of these automation files (as well as the different MTConnect implementation files described in Sec. 2) is that it is best practice to locate all of these files in the same directory of the Linux operating system. Because they should be placed in different locations, though, based on accepted best practices, these files may be symbolically linked from the directory created to their required locations. In this way, there is one directory to maintain and backup, which eliminates the need to hunt and track multiple files in multiple directories. For the Test Bed, we decided to place our directory in `/opt`, which is reserved for all software and packages that are not included in the standard installation of the Linux distribution that we used.

5. Summary

Based on the implementation design described in Sec. 3, we deployed the networking infrastructure required for the Test Bed on the shop floor of the NIST FTO. Industrially-hardened networking equipment was used wherever feasible to ensure the reliability of our networking infrastructure. However, commercial-grade equipment should be sufficient as well; the choice of networking equipment should be balanced against other considerations depending on the specific use case that is being addressed.

The implementation presented in this document represents an on-going research effort at NIST. In the future, our goal is to expand on our data collection capabilities by further hardening our networking infrastructure and adding new data items for each piece of equipment. Additional sensors that we hope to integrate for different pieces of equipment in the future include powermeters, accelerometers, and thermocouples. We also hope to incorporate data from other manufacturing systems, such as manufacturing execution system (MES). As we expand our efforts in the Test Bed, we plan to update this document with additional details to share best practices about how to incorporate these data items into the larger implementation.

Acknowledgments

The authors wish to thank Andrew Mundy, Carolyn Rowland, Paul Dickey, Steve Bannon, Allison Barnard Feeney, and Brian Weiss from NIST and Athulan Vijayaraghavan, Will Sobel, and Kevin Fiejdasz from VIMANA and many other industry collaborators for their support in helping realize the NIST SMS Test Bed. They also wish to thank Allison Barnard Feeney and Vijay Srinivasan from NIST and Paul Huang from the Office of Naval Research Laboratory for their comments and input to this report. The authors prepared this report as partial fulfillment to the requirements covered by the reimbursable agreement N0001418IP00053 between NIST and the Office of Naval Research Laboratory.

References

- [1] Hedberg Jr T, Helu M (2017) Design and configuration of the smart manufacturing systems test bed (National Institute of Standards and Technology), Report NIST AMS 200-1. <https://doi.org/10.6028/NIST.AMS.200-1>
- [2] Sobel W (2015) Mtconnect architecture and research overview. Available at <https://static1.squarespace.com/static/54011775e4b0bc1fe0fb8494/t/55a95c85e4b0abba4f23f363/1437162629562/MTConnectTechnicalWorkshopAtNAMRC43.pdf>.
- [3] Helu M, Hedberg Jr T, Barnard Feeney A (2017) Reference architecture to integrate heterogeneous manufacturing systems for the digital thread. *Journal of Manufacturing Science and Technology* <https://doi.org/10.1016/j.cirpj.2017.04.002>
- [4] Helu M, Hedberg Jr T (2020) Recommendations for collecting, curating, and re-using manufacturing data (National Institute of Standards and Technology), Report NIST AMS 300-1. <https://doi.org/10.6028/NIST.AMS.300-11>
- [5] Helu M, Hedberg Jr T (2015) Enabling smart manufacturing research and development using a product lifecycle test bed. *Procedia Manufacturing* 1:86–97.
- [6] Helu M, Hedberg Jr T, Barnard Feeney A, Luce M (2016) Volatile data stream (VDS) for the smart manufacturing systems (SMS) test bed using MTConnect. <https://doi.org/10.18434/T4FK54>. Available at <https://smstestbed.nist.gov/vds>
- [7] Hedberg Jr T, Helu M, Barnard Feeney A, Luce M (2016) Query-able data repository (QDR) for the smart manufacturing systems (SMS) test bed using mtconnect. Available at <https://smstestbed.nist.gov/qdr>.
- [8] MTConnect Institute (2018) ANSI MTConnect Version 1.4.0 (ANSI/MTC1.4-2018). Available at <https://www.mtconnect.org/standard20181>.
- [9] Hedberg Jr T, Helu M, Newrock M (2017) Software requirements specification to distribute manufacturing data (National Institute of Standards and Technology), Report NIST AMS 300-2. <https://doi.org/10.6028/NIST.AMS.300-2>
- [10] International Electrochemical Commission (2016) OPC Unified Architecture – Part 1: Overview and Concepts (IEC TR 62541-1:2016).
- [11] MTConnect Institute, OPC Foundation (2019) OPC Unified Architecture for MTConnect (OPC 30070-1 – MTConnect Part 1: Device Model) Version 2.00. Available at <https://opcfoundation.org/developer-tools/specifications-opc-ua-information-models/opc-unified-architecture-for-mtconnect/>.
- [12] International Organization for Standardization, International Electrotechnical Commission (2016) Information technology — Message Queuing Telemetry Transport (MQTT) v3.1.1 (ISO/IEC 20922:2016). Available at <https://www.iso.org/standard/69466.html>.
- [13] Helu M, Sprock T, Hartenstine D, Venketesh R, Sobel W (2020) Scalable data pipeline architecture to support the industrial internet of things. *CIRP Annals – Manufacturing Technology* <https://doi.org/10.1016/j.cirp.2020.04.006>

Appendix A: Devices XML File

Listing 1 is an example of a Devices XML file from the Test Bed. This file is used with the Agent configuration file to set up an MTConnect Agent for data collection from the devices described by the Devices XML file.

Listing 1. Devices XML file from the Test Bed

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xml-stylesheet type="text/xsl" href="/styles/Devices.xsl"?>
3 <MTConnectDevices xmlns:m="urn:mtconnect.org:MTConnectDevices:1.3" xmlns="
  ↳ urn:mtconnect.org:MTConnectDevices:1.3" xmlns:xsi="http://www.w3.
  ↳ org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mtconnect.
  ↳ org:MTConnectDevices:1.3 /schemas/MTConnectDevices_1.3.xsd">
4 <Header creationTime="2017-06-28T16:31:49Z" sender="95427d753b45"
  ↳ instanceId="1498573777" version="1.3.0.17" assetBufferSize="1024"
  ↳ assetCount="0" bufferSize="1048576"/>
5 <Devices>
6 <Device id="GFAgie01" name="GFAgie01" uuid="mtc_adapter001">
7 <Description manufacturer="Agie Charmilles" model="HPM600U">Agie
  ↳ Mikron HPM600U - GF Agie Charmilles HPM600U</Description>
8 <DataItems>
9 <DataItem category="EVENT" id="GFAgie01-dtop_1" name="avail" type="
  ↳ AVAILABILITY"/>
10 <DataItem category="EVENT" id="GFAgie01-dtop_2" name="estop" type="
  ↳ EMERGENCY_STOP"/>
11 <DataItem category="CONDITION" id="GFAgie01-dtop_3" name="system"
  ↳ type="SYSTEM"/>
12 <DataItem category="EVENT" id="GFAgie01_asset_chg" type="
  ↳ ASSET_CHANGED"/>
13 <DataItem category="EVENT" id="GFAgie01_asset_rem" type="
  ↳ ASSET_REMOVED"/>
14 </DataItems>
15 <Components>
16 <Axes id="GFAgie01-axes_1" name="axes">
17 <Components>
18 <Linear id="GFAgie01-X_1" name="X">
19 <DataItems>
20 <DataItem category="SAMPLE" id="GFAgie01-X_2" name="
  ↳ Xposition" nativeUnits="MILLIMETER" subType="ACTUAL"
  ↳ type="POSITION" units="MILLIMETER"/>
21 </DataItems>
22 </Linear>
23 <Linear id="GFAgie01-Y_1" name="Y">
24 <DataItems>
25 <DataItem category="SAMPLE" id="GFAgie01-Y_2" name="
  ↳ Yposition" nativeUnits="MILLIMETER" subType="ACTUAL"
```

```

25         ↪ type="POSITION" units="MILLIMETER"/>
26     </DataItems>
27 </Linear>
28 <Linear id="GFAGie01-Z_1" name="Z">
29     <DataItems>
30         <DataItem category="SAMPLE" id="GFAGie01-Z_2" name="
31             ↪ Zposition" nativeUnits="MILLIMETER" subType="ACTUAL"
32             ↪ type="POSITION" units="MILLIMETER"/>
33     </DataItems>
34 </Linear>
35 <Rotary id="GFAGie01-C_1" name="C">
36     <DataItems>
37         <DataItem category="SAMPLE" id="GFAGie01-C_2" name="
38             ↪ Cposition" nativeUnits="DEGREE" subType="ACTUAL" type
39             ↪ ="ANGLE" units="DEGREE"/>
40     </DataItems>
41 </Rotary>
42 <Rotary id="GFAGie01-A_1" name="A">
43     <DataItems>
44         <DataItem category="SAMPLE" id="GFAGie01-A_2" name="
45             ↪ Aposition" nativeUnits="DEGREE" subType="ACTUAL" type
46             ↪ ="ANGLE" units="DEGREE"/>
47     </DataItems>
48 </Rotary>
49 </Components>
50 </Axes>
51 <Controller id="GFAGie01-controller_basic_1" name="controller_basic
52     ↪ ">
53 <DataItems>
54     <DataItem category="EVENT" id="GFAGie01-controller_basic_2"
55         ↪ name="Fovr" nativeUnits="PERCENT" subType="OVERRIDE"
56         ↪ type="PATH_FEEDRATE_OVERRIDE" units="PERCENT"/>
57     <DataItem category="EVENT" id="GFAGie01-controller_basic_3"
58         ↪ name="Sovr" nativeUnits="PERCENT" type="
59         ↪ ROTARY_VELOCITY_OVERRIDE" units="PERCENT"/>
60     <DataItem category="CONDITION" id="GFAGie01-controller_basic_4"
61         ↪ name="servo" type="ACTUATOR"/>
62     <DataItem category="CONDITION" id="GFAGie01-controller_basic_5"
63         ↪ name="comms" type="COMMUNICATIONS"/>
64     <DataItem category="CONDITION" id="GFAGie01-controller_basic_6"
65         ↪ name="pneumatic" type="COMMUNICATIONS"/>
66     <DataItem category="CONDITION" id="GFAGie01-controller_basic_7"
67         ↪ name="hydraulic" type="COMMUNICATIONS"/>
68     <DataItem category="CONDITION" id="GFAGie01-controller_basic_8"
69         ↪ name="logic" type="LOGIC_PROGRAM"/>
70     <DataItem category="CONDITION" id="GFAGie01-controller_basic_9"

```

```

55         ↪ name="motion" type="MOTION_PROGRAM"/>
        <DataItem category="CONDITION" id="GFAGie01-controller_basic_10
56         ↪ " name="cnc_temp" type="TEMPERATURE"/>
57     </DataItems>
58     <Components>
59         <Path id="GFAGie01-path_basic_1" name="path_basic">
60             <DataItems>
61                 <DataItem category="EVENT" id="GFAGie01-path_basic_2" name=
62                 ↪ "execution" type="EXECUTION"/>
63                 <DataItem category="EVENT" id="GFAGie01-path_basic_3" name=
64                 ↪ "mode" type="CONTROLLER_MODE"/>
65                 <DataItem category="EVENT" id="GFAGie01-path_basic_4" name=
66                 ↪ "program" type="PROGRAM"/>
67                 <DataItem category="EVENT" id="GFAGie01-path_basic_5" name=
68                 ↪ "line" type="LINE"/>
69                 <DataItem category="EVENT" id="GFAGie01-path_basic_6" name=
70                 ↪ "move" type="x:MOTION"/>
71                 <DataItem category="SAMPLE" id="GFAGie01-path_basic_7" name
72                 ↪ ="path_pos" nativeUnits="MILLIMETER_3D" type="
73                 ↪ PATH_POSITION" units="MILLIMETER_3D"/>
74             </DataItems>
75         </Path>
76     </Components>
77 </Controller>
78 </Components>
79 </Device>
80 <Device id="Mazak01" name="Mazak01" uuid="mtc_adapter002">
81     <Description manufacturer="Mazak" model="Integrex" serialNumber="
82     ↪ M7303290458">Mazak Integrex 100-IV</Description>
83     <DataItems>
84         <DataItem category="EVENT" id="Mazak01-dtop_1" name="avail" type="
85         ↪ AVAILABILITY"/>
86         <DataItem category="EVENT" id="Mazak01_asset_chg" type="
87         ↪ ASSET_CHANGED"/>
88         <DataItem category="EVENT" id="Mazak01_asset_rem" type="
89         ↪ ASSET_REMOVED"/>
90     </DataItems>
91     <Components>
92         <Axes id="Mazak01-base" name="base">
93             <DataItems>
94                 <DataItem category="CONDITION" id="Mazak01-base_1" name="
95                 ↪ servo_cond" type="ACTUATOR"/>
96             </DataItems>
97         <Components>
98             <Linear id="Mazak01-X" name="X">
99                 <DataItems>

```

```

87     <DataItem category="SAMPLE" coordinateSystem="MACHINE" id="
      ↪ Mazak01-X_1" name="Xabs" nativeUnits="MILLIMETER"
      ↪ subType="ACTUAL" type="POSITION" units="MILLIMETER"/>
88     <DataItem category="CONDITION" id="Mazak01-X_2" name="
      ↪ Xtravel" type="POSITION"/>
89     <DataItem category="SAMPLE" id="Mazak01-X_3" name="Xload"
      ↪ nativeUnits="PERCENT" type="LOAD" units="PERCENT"/>
90     <DataItem category="SAMPLE" id="Mazak01-X_4" name="Xfrt"
      ↪ nativeUnits="MILLIMETER/SECOND" type="AXIS_FEEDRATE"
      ↪ units="MILLIMETER/SECOND"/>
91   </DataItems>
92 </Linear>
93 <Linear id="Mazak01-Y" name="Y">
94   <DataItems>
95     <DataItem category="SAMPLE" coordinateSystem="MACHINE" id="
      ↪ Mazak01-Y_1" name="Yabs" nativeUnits="MILLIMETER"
      ↪ subType="ACTUAL" type="POSITION" units="MILLIMETER"/>
96     <DataItem category="CONDITION" id="Mazak01-Y_2" name="
      ↪ Ytravel" type="POSITION"/>
97     <DataItem category="SAMPLE" id="Mazak01-Y_3" name="Yload"
      ↪ nativeUnits="PERCENT" type="LOAD" units="PERCENT"/>
98     <DataItem category="SAMPLE" id="Mazak01-Y_4" name="Yfrt"
      ↪ nativeUnits="MILLIMETER/SECOND" type="AXIS_FEEDRATE"
      ↪ units="MILLIMETER/SECOND"/>
99   </DataItems>
100 </Linear>
101 <Linear id="Mazak01-Z" name="Z">
102   <DataItems>
103     <DataItem category="SAMPLE" coordinateSystem="MACHINE" id="
      ↪ Mazak01-Z_1" name="Zabs" nativeUnits="MILLIMETER"
      ↪ subType="ACTUAL" type="POSITION" units="MILLIMETER"/>
104     <DataItem category="CONDITION" id="Mazak01-Z_2" name="
      ↪ Ztravel" type="POSITION"/>
105     <DataItem category="SAMPLE" id="Mazak01-Z_3" name="Zload"
      ↪ nativeUnits="PERCENT" type="LOAD" units="PERCENT"/>
106     <DataItem category="SAMPLE" id="Mazak01-Z_4" name="Zfrt"
      ↪ nativeUnits="MILLIMETER/SECOND" type="AXIS_FEEDRATE"
      ↪ units="MILLIMETER/SECOND"/>
107   </DataItems>
108 </Linear>
109 <Rotary id="Mazak01-B" name="B">
110   <DataItems>
111     <DataItem category="SAMPLE" id="Mazak01-B_1" name="Bload"
      ↪ nativeUnits="PERCENT" type="LOAD" units="PERCENT"/>
112     <DataItem category="SAMPLE" id="Mazak01-B_2" name="Bfrt"
      ↪ nativeUnits="DEGREE/MINUTE" type="ANGULAR_VELOCITY"

```



```

113         ↪ units="DEGREE/SECOND"/>
        <DataItem category="CONDITION" id="Mazak01-B_3" name="
114         ↪ Btravel" type="ANGLE"/>
        <DataItem category="SAMPLE" id="Mazak01-B_4" name="Bdeg"
115         ↪ nativeUnits="DEGREE" subType="ACTUAL" type="ANGLE"
        ↪ units="DEGREE"/>
    </DataItems>
116 </Rotary>
117 <Rotary id="Mazak01-C" name="C">
118     <DataItems>
119         <DataItem category="SAMPLE" id="Mazak01-C_1" name="Cload"
        ↪ nativeUnits="PERCENT" type="LOAD" units="PERCENT"/>
120         <DataItem category="SAMPLE" id="Mazak01-C_2" name="Sload"
        ↪ nativeUnits="PERCENT" type="LOAD" units="PERCENT"/>
121         <DataItem category="CONDITION" id="Mazak01-C_3" name="
        ↪ Ctravel" type="ANGLE"/>
122         <DataItem category="SAMPLE" id="Mazak01-C_4" name="Cfrt"
        ↪ nativeUnits="DEGREE/MINUTE" type="ANGULAR_VELOCITY"
        ↪ units="DEGREE/SECOND"/>
123         <DataItem category="SAMPLE" id="Mazak01-C_5" name="Srpm"
        ↪ nativeUnits="REVOLUTION/MINUTE" subType="ACTUAL" type
        ↪ ="ROTARY_VELOCITY" units="REVOLUTION/MINUTE"/>
124         <DataItem category="EVENT" id="Mazak01-C_6" name="Sovr"
        ↪ nativeUnits="PERCENT" type="ROTARY_VELOCITY_OVERRIDE"
        ↪ units="PERCENT"/>
125         <DataItem category="SAMPLE" id="Mazak01-C_7" name="Stemp"
        ↪ nativeUnits="CELSIUS" type="TEMPERATURE" units="
        ↪ CELSIUS"/>
126         <DataItem category="SAMPLE" id="Mazak01-C_8" name="Cdeg"
        ↪ nativeUnits="DEGREE" subType="ACTUAL" type="ANGLE"
        ↪ units="DEGREE"/>
127         <DataItem category="CONDITION" id="Mazak01-C_9" name="
        ↪ Sload_cond" type="LOAD"/>
128         <DataItem category="CONDITION" id="Mazak01-C_10" name="
        ↪ Stemp_cond" type="TEMPERATURE"/>
129     </DataItems>
130 </Rotary>
131 <Rotary id="Mazak01-C2" name="C">
132     <DataItems>
133         <DataItem category="SAMPLE" id="Mazak01-C2_1" name="S2load"
        ↪ nativeUnits="PERCENT" type="LOAD" units="PERCENT"/>
134         <DataItem category="SAMPLE" id="Mazak01-C2_2" name="S2rpm"
        ↪ nativeUnits="REVOLUTION/MINUTE" subType="ACTUAL" type
        ↪ ="ROTARY_VELOCITY" units="REVOLUTION/MINUTE"/>
135         <DataItem category="SAMPLE" id="Mazak01-C2_3" name="S2temp"
        ↪ nativeUnits="CELSIUS" type="TEMPERATURE" units="

```

```

136         ↪ CELSIUS"/>
        <DataItem category="CONDITION" id="Mazak01-C2_4" name="
137         ↪ S2load_cond" type="LOAD"/>
        <DataItem category="CONDITION" id="Mazak01-C2_5" name="
138         ↪ S2temp_cond" type="TEMPERATURE"/>
139     </DataItems>
140 </Rotary>
141 </Components>
142 </Axes>
143 <Controller id="Mazak01-controller" name="controller">
144     <DataItems>
145         <DataItem category="CONDITION" id="Mazak01-controller_1" name="
146         ↪ comms_cond" type="COMMUNICATIONS"/>
147         <DataItem category="CONDITION" id="Mazak01-controller_2" name="
148         ↪ logic_cond" type="LOGIC_PROGRAM"/>
149         <DataItem category="CONDITION" id="Mazak01-controller_3" name="
150         ↪ system_cond" type="SYSTEM"/>
151         <DataItem category="EVENT" id="Mazak01-controller_4" name="
152         ↪ estop" type="EMERGENCY_STOP"/>
153     </DataItems>
154     <Components>
155         <Path id="Mazak01-path" name="path">
156             <DataItems>
157                 <DataItem category="EVENT" id="Mazak01-path_1" name="
158                 ↪ program" type="PROGRAM"/>
159                 <DataItem category="EVENT" id="Mazak01-path_2" name="
160                 ↪ subprogram" subType="x:SUB" type="PROGRAM"/>
161                 <DataItem category="EVENT" id="Mazak01-path_3" name="line"
162                 ↪ type="LINE"/>
163                 <DataItem category="EVENT" id="Mazak01-path_4" name="
164                 ↪ unitNum" type="x:UNIT"/>
165                 <DataItem category="EVENT" id="Mazak01-path_5" name="
166                 ↪ sequenceNum" type="x:SEQUENCE_NUMBER"/>
167                 <DataItem category="EVENT" id="Mazak01-path_6" name="
168                 ↪ PartCountAct" type="PART_COUNT"/>
169                 <DataItem category="SAMPLE" coordinateSystem="WORK" id="
170                 ↪ Mazak01-path_7" name="Fact" nativeUnits="FOOT/MINUTE"
171                 ↪ subType="ACTUAL" type="PATH_FEEDRATE" units="
172                 ↪ MILLIMETER/SECOND"/>
173                 <DataItem category="EVENT" id="Mazak01-path_8" name="
174                 ↪ Frapidovr" nativeUnits="PERCENT" subType="RAPID" type
175                 ↪ ="PATH_FEEDRATE_OVERRIDE" units="PERCENT"/>
176                 <DataItem category="EVENT" id="Mazak01-path_9" name="Fovr"
177                 ↪ nativeUnits="PERCENT" subType="PROGRAMMED" type="
178                 ↪ PATH_FEEDRATE_OVERRIDE" units="PERCENT"/>
179                 <DataItem category="EVENT" id="Mazak01-path_10" name="

```

```

162         ↪ Tool_number" type="TOOL_NUMBER"/>
        <DataItem category="EVENT" id="Mazak01-path_11" name="
163         ↪ Tool_group" type="x:TOOL_GROUP"/>
        <DataItem category="EVENT" id="Mazak01-path_12" name="
164         ↪ Tool_suffix" type="x:TOOL_SUFFIX"/>
        <DataItem category="EVENT" id="Mazak01-path_13" name="
165         ↪ execution" type="EXECUTION"/>
        <DataItem category="EVENT" id="Mazak01-path_14" name="mode"
166         ↪ type="CONTROLLER_MODE"/>
        <DataItem category="EVENT" id="Mazak01-path_15" name="
167         ↪ program_cmt" type="x:PROGRAM_COMMENT"/>
        <DataItem category="EVENT" id="Mazak01-path_16" name="
168         ↪ subprogram_cmt" subType="x:SUB" type="
        ↪ x:PROGRAM_COMMENT"/>
        <DataItem category="SAMPLE" id="Mazak01-path_17" name="
169         ↪ auto_time" subType="x:AUTO" type="ACCUMULATED_TIME"/>
        <DataItem category="SAMPLE" id="Mazak01-path_18" name="
170         ↪ total_time" subType="x:TOTAL" type="ACCUMULATED_TIME"
        ↪ />
        <DataItem category="SAMPLE" id="Mazak01-path_19" name="
171         ↪ cut_time" subType="x:CUT" type="ACCUMULATED_TIME"/>
        <DataItem category="CONDITION" id="Mazak01-path_20" name="
172         ↪ motion_cond" type="MOTION_PROGRAM"/>
        <DataItem category="CONDITION" id="Mazak01-path_21" name="
        ↪ path_system" type="SYSTEM"/>
173     </DataItems>
174 </Path>
175 </Components>
176 </Controller>
177 <Systems id="Mazak01-systems" name="systems">
178     <Components>
179         <Electric id="Mazak01-electric" name="electric">
180             <DataItems>
181                 <DataItem category="EVENT" id="Mazak01-electric_1" name="
                    ↪ power" type="POWER_STATE"/>
182                 <DataItem category="CONDITION" id="Mazak01-electric_2" name
                    ↪ ="electric_temp" type="TEMPERATURE"/>
183             </DataItems>
184         </Electric>
185         <Hydraulic id="Mazak01-hydraulic" name="hydraulic">
186             <DataItems>
187                 <DataItem category="CONDITION" id="Mazak01-hydraulic_1"
                    ↪ name="hydra_cond" type="PRESSURE"/>
188             </DataItems>
189         </Hydraulic>
190         <Coolant id="Mazak01-coolant" name="coolant">

```

```

191     <DataItems>
192     <DataItem category="CONDITION" id="Mazak01-coolant_1" name=
        ↪ "coolant_pres" type="PRESSURE"/>
193     <DataItem category="CONDITION" id="Mazak01-coolant_2" name=
        ↪ "coolant_temp" type="TEMPERATURE"/>
194     <DataItem category="CONDITION" id="Mazak01-coolant_3" name=
        ↪ "coolant_level" type="LEVEL"/>
195     </DataItems>
196 </Coolant>
197 <Pneumatic id="Mazak01-pneumatic" name="pneumatic">
198     <DataItems>
199     <DataItem category="CONDITION" id="Mazak01-pneumatic_1"
        ↪ name="pneu_cond" type="PRESSURE"/>
200     </DataItems>
201 </Pneumatic>
202 </Components>
203 </Systems>
204 </Components>
205 </Device>
206 <Device id="Mazak03" name="Mazak03" uuid="mtc_adapter004">
207     <Description manufacturer="Mazak" model="QTN">Mazak QuickTurn -
        ↪ Mazak QuickTurn Nexus 300</Description>
208     <DataItems>
209     <DataItem category="EVENT" id="Mazak03-dtop_1" name="avail" type="
        ↪ AVAILABILITY"/>
210     <DataItem category="EVENT" id="Mazak03_asset_chg" type="
        ↪ ASSET_CHANGED"/>
211     <DataItem category="EVENT" id="Mazak03_asset_rem" type="
        ↪ ASSET_REMOVED"/>
212     </DataItems>
213     <Components>
214     <Axes id="Mazak03-base_1" name="base">
215     <DataItems>
216     <DataItem category="CONDITION" id="Mazak03-base_2" name="
        ↪ servo_cond" type="ACTUATOR"/>
217     </DataItems>
218     <Components>
219     <Linear id="Mazak03-X_1" name="X">
220     <DataItems>
221     <DataItem category="SAMPLE" coordinateSystem="MACHINE" id="
        ↪ Mazak03-X_2" name="Xabs" nativeUnits="MILLIMETER"
        ↪ subType="ACTUAL" type="POSITION" units="MILLIMETER"/>
222     <DataItem category="CONDITION" id="Mazak03-X_3" name="
        ↪ Xtravel" type="POSITION"/>
223     <DataItem category="SAMPLE" id="Mazak03-X_4" name="Xload"
        ↪ nativeUnits="PERCENT" type="LOAD" units="PERCENT"/>

```

```

224         <DataItem category="SAMPLE" id="Mazak03-X_5" name="Xfrt"
           ↪ nativeUnits="MILLIMETER/SECOND" type="AXIS_FEEDRATE"
           ↪ units="MILLIMETER/SECOND"/>
225     </DataItems>
226 </Linear>
227 <Linear id="Mazak03-Z_1" name="Z">
228     <DataItems>
229         <DataItem category="SAMPLE" coordinateSystem="MACHINE" id="
           ↪ Mazak03-Z_2" name="Zabs" nativeUnits="MILLIMETER"
           ↪ subType="ACTUAL" type="POSITION" units="MILLIMETER"/>
230         <DataItem category="CONDITION" id="Mazak03-Z_3" name="
           ↪ Ztravel" type="POSITION"/>
231         <DataItem category="SAMPLE" id="Mazak03-Z_4" name="Zload"
           ↪ nativeUnits="PERCENT" type="LOAD" units="PERCENT"/>
232         <DataItem category="SAMPLE" id="Mazak03-Z_5" name="Zfrt"
           ↪ nativeUnits="MILLIMETER/SECOND" type="AXIS_FEEDRATE"
           ↪ units="MILLIMETER/SECOND"/>
233     </DataItems>
234 </Linear>
235 <Rotary id="Mazak03-C_1" name="C">
236     <DataItems>
237         <DataItem category="SAMPLE" id="Mazak03-C_2" name="Cload"
           ↪ nativeUnits="PERCENT" type="LOAD" units="PERCENT"/>
238         <DataItem category="SAMPLE" id="Mazak03-C_3" name="Cfrt"
           ↪ nativeUnits="DEGREE/MINUTE" type="ANGULAR_VELOCITY"
           ↪ units="DEGREE/SECOND"/>
239         <DataItem category="SAMPLE" id="Mazak03-C_4" name="Cdeg"
           ↪ nativeUnits="DEGREE" subType="ACTUAL" type="ANGLE"
           ↪ units="DEGREE"/>
240         <DataItem category="CONDITION" id="Mazak03-C_5" name="
           ↪ Ctravel" type="ANGLE"/>
241     </DataItems>
242 </Rotary>
243 <Rotary id="Mazak03-S_1" name="S" nativeName="S">
244     <DataItems>
245         <DataItem category="SAMPLE" id="Mazak03-S_2" name="Srpm"
           ↪ nativeUnits="REVOLUTION/MINUTE" subType="ACTUAL" type
           ↪ ="ROTARY_VELOCITY" units="REVOLUTION/MINUTE"/>
246         <DataItem category="SAMPLE" id="Mazak03-S_3" name="S1load"
           ↪ nativeUnits="PERCENT" type="LOAD" units="PERCENT">
247             <Source>Sload</Source>
248         </DataItem>
249         <DataItem category="SAMPLE" id="Mazak03-S_4" name="Stemp"
           ↪ nativeUnits="CELSIUS" type="TEMPERATURE" units="
           ↪ CELSIUS"/>
250         <DataItem category="CONDITION" id="Mazak03-S_5" name="

```

```

251         ↪ Sload_cond" type="LOAD"/>
        <DataItem category="CONDITION" id="Mazak03-S_6" name="
252         ↪ Stemp_cond" type="TEMPERATURE"/>
    </DataItems>
253 </Rotary>
254 <Rotary id="Mazak03-S2_1" name="S2" nativeName="S2">
255     <DataItems>
256         <DataItem category="SAMPLE" id="Mazak03-S2_2" name="S2rpm"
                ↪ nativeUnits="REVOLUTION/MINUTE" subType="ACTUAL" type
                ↪ ="ROTARY_VELOCITY" units="REVOLUTION/MINUTE"/>
257         <DataItem category="SAMPLE" id="Mazak03-S2_3" name="S2load"
                ↪ nativeUnits="PERCENT" type="LOAD" units="PERCENT">
258             <Source>S2load</Source>
259         </DataItem>
260         <DataItem category="SAMPLE" id="Mazak03-S2_4" name="S2temp"
                ↪ nativeUnits="CELSIUS" type="TEMPERATURE" units="
                ↪ CELSIUS"/>
261         <DataItem category="CONDITION" id="Mazak03-S2_5" name="
                ↪ S2load_cond" type="LOAD"/>
262         <DataItem category="CONDITION" id="Mazak03-S2_6" name="
                ↪ S2temp_cond" type="TEMPERATURE"/>
263     </DataItems>
264 </Rotary>
265 </Components>
266 </Axes>
267 <Controller id="Mazak03-controller_1" name="controller">
268     <DataItems>
269         <DataItem category="CONDITION" id="Mazak03-controller_2" name="
                ↪ comms_cond" type="COMMUNICATIONS"/>
270         <DataItem category="CONDITION" id="Mazak03-controller_3" name="
                ↪ logic_cond" type="LOGIC_PROGRAM"/>
271         <DataItem category="CONDITION" id="Mazak03-controller_4" name="
                ↪ system_cond" type="SYSTEM"/>
272         <DataItem category="EVENT" id="Mazak03-controller_5" name="
                ↪ pallet_num" type="x:PALLET_NUMBER"/>
273     </DataItems>
274     <Components>
275         <Path id="Mazak03-path_1" name="path">
276             <DataItems>
277                 <DataItem category="EVENT" id="Mazak03-path_2" name="Sovr"
                        ↪ nativeUnits="PERCENT" type="ROTARY_VELOCITY_OVERRIDE"
                        ↪ units="PERCENT"/>
278                 <DataItem category="EVENT" id="Mazak03-path_3" name="
                        ↪ program" type="PROGRAM"/>
279                 <DataItem category="EVENT" id="Mazak03-path_4" name="
                        ↪ subprogram" subType="x:SUB" type="PROGRAM"/>

```

```

280 <DataItem category="EVENT" id="Mazak03-path_5" name="line"
      ↪ type="LINE"/>
281 <DataItem category="EVENT" id="Mazak03-path_6" name="
      ↪ unitNum" type="x:UNIT"/>
282 <DataItem category="EVENT" id="Mazak03-path_95" name="
      ↪ sequenceNum" type="x:SEQUENCE_NUMBER"/>
283 <DataItem category="EVENT" id="Mazak03-path_96" name="
      ↪ PartCountAct" type="PART_COUNT"/>
284 <DataItem category="SAMPLE" id="Mazak03-path_7" name="Fact"
      ↪ nativeUnits="MILLIMETER/SECOND" subType="ACTUAL"
      ↪ type="PATH_FEEDRATE" units="MILLIMETER/SECOND"/>
285 <DataItem category="EVENT" id="Mazak03-path_8" name="
      ↪ Frapidovr" nativeUnits="PERCENT" subType="RAPID" type
      ↪ ="PATH_FEEDRATE_OVERRIDE" units="PERCENT"/>
286 <DataItem category="EVENT" id="Mazak03-path_9" name="Fovr"
      ↪ nativeUnits="PERCENT" subType="PROGRAMMED" type="
      ↪ PATH_FEEDRATE_OVERRIDE" units="PERCENT"/>
287 <DataItem category="EVENT" id="Mazak03-path_10" name="
      ↪ execution" type="EXECUTION"/>
288 <DataItem category="EVENT" id="Mazak03-path_11" name="mode"
      ↪ type="CONTROLLER_MODE"/>
289 <DataItem category="EVENT" id="Mazak03-path_12" name="
      ↪ program_cmt" type="PROGRAM_COMMENT"/>
290 <DataItem category="EVENT" id="Mazak03-path_13" name="
      ↪ subprogram_cmt" subType="x:SUB_PROGRAM" type="
      ↪ PROGRAM_COMMENT"/>
291 <DataItem category="SAMPLE" id="Mazak03-path_14" name="
      ↪ auto_time" nativeUnits="SECOND" subType="x:AUTO" type
      ↪ ="ACCUMULATED_TIME" units="SECOND"/>
292 <DataItem category="SAMPLE" id="Mazak03-path_15" name="
      ↪ total_time" nativeUnits="SECOND" subType="x:TOTAL"
      ↪ type="ACCUMULATED_TIME" units="SECOND"/>
293 <DataItem category="SAMPLE" id="Mazak03-path_16" name="
      ↪ cut_time" nativeUnits="SECOND" subType="x:CUT" type="
      ↪ ACCUMULATED_TIME" units="SECOND"/>
294 <DataItem category="CONDITION" id="Mazak03-path_17" name="
      ↪ motion_cond" type="MOTION_PROGRAM"/>
295 <DataItem category="CONDITION" id="Mazak03-path_18" name="
      ↪ motion_cond2" subType="x:MOTION_PROGRAM_2" type="
      ↪ MOTION_PROGRAM"/>
296 <DataItem category="CONDITION" id="Mazak03-path_19" name="
      ↪ motion_cond3" subType="x:MOTION_PROGRAM_3" type="
      ↪ MOTION_PROGRAM"/>
297 <DataItem category="CONDITION" id="Mazak03-path_20" name="
      ↪ motion_cond4" subType="x:MOTION_PROGRAM_4" type="
      ↪ MOTION_PROGRAM"/>

```

```

298     <DataItem category="CONDITION" id="Mazak03-path_21" name="
      ↪ path_system" type="SYSTEM"/>
299     <DataItem category="CONDITION" id="Mazak03-path_22" name="
      ↪ path_system2" subType="x:SYSTEM_1" type="SYSTEM"/>
300     <DataItem category="CONDITION" id="Mazak03-path_23" name="
      ↪ path_system3" subType="x:SYSTEM_2" type="SYSTEM"/>
301     <DataItem category="CONDITION" id="Mazak03-path_24" name="
      ↪ path_system4" subType="x:SYSTEM_3" type="SYSTEM"/>
302     <DataItem category="EVENT" id="Mazak03-path_25" name="
      ↪ Tool_number" type="TOOL_ASSET_ID"/>
303     <DataItem category="EVENT" id="Mazak03-path_26" name="
      ↪ Tool_group" type="x:TOOL_GROUP"/>
304     <DataItem category="EVENT" id="Mazak03-path_27" name="
      ↪ Tool_suffix" type="x:TOOL_SUFFIX"/>
305     </DataItems>
306   </Path>
307 </Components>
308 </Controller>
309 <Systems id="Mazak03-systems_1" name="systems">
310   <Components>
311     <Electric id="Mazak03-electric_1" name="electric">
312       <DataItems>
313         <DataItem category="EVENT" id="Mazak03-electric_2" name="
          ↪ power" type="POWER_STATE"/>
314         <DataItem category="CONDITION" id="Mazak03-electric_3" name
          ↪ ="electric_temp" type="TEMPERATURE"/>
315       </DataItems>
316     </Electric>
317     <Hydraulic id="Mazak03-hydraulic_1" name="hydraulic">
318       <DataItems>
319         <DataItem category="CONDITION" id="Mazak03-hydraulic_2"
          ↪ name="hydra_cond" type="PRESSURE"/>
320       </DataItems>
321     </Hydraulic>
322     <Coolant id="Mazak03-coolant_1" name="coolant">
323       <DataItems>
324         <DataItem category="CONDITION" id="Mazak03-coolant_2" name=
          ↪ "coolant_pres" type="PRESSURE"/>
325         <DataItem category="CONDITION" id="Mazak03-coolant_3" name=
          ↪ "coolant_temp" type="TEMPERATURE"/>
326         <DataItem category="CONDITION" id="Mazak03-coolant_4" name=
          ↪ "coolant_level" type="LEVEL"/>
327       </DataItems>
328     </Coolant>
329     <Pneumatic id="Mazak03-pneumatic_1" name="pneumatic">
330       <DataItems>

```



```

331         <DataItem category="CONDITION" id="Mazak03-pneumatic_2"
           ↪ name="pneu_cond" type="PRESSURE"/>
332     </DataItems>
333 </Pneumatic>
334 </Components>
335 </Systems>
336 </Components>
337 </Device>
338 <Device id="Hurco01" name="Hurco01" uuid="mtc_adapter005">
339     <Description manufacturer="Hurco" model="VMX24">Hurco VMX 24 #1</
           ↪ Description>
340     <DataItems>
341         <DataItem category="EVENT" id="Hurco01-dtop_1" name="avail" type="
           ↪ AVAILABILITY"/>
342         <DataItem category="EVENT" id="Hurco01_asset_chg" type="
           ↪ ASSET_CHANGED"/>
343         <DataItem category="EVENT" id="Hurco01_asset_rem" type="
           ↪ ASSET_REMOVED"/>
344     </DataItems>
345     <Components>
346         <Axes id="Hurco01-axes" name="Axes">
347             <Components>
348                 <Rotary id="Hurco01-C" name="C">
349                     <DataItems>
350                         <DataItem category="SAMPLE" id="Hurco01-C_1" name="
           ↪ Spindle_Speed" nativeUnits="REVOLUTION/MINUTE"
           ↪ subType="ACTUAL" type="ROTARY_VELOCITY" units="
           ↪ REVOLUTION/MINUTE">
351                             <Source>spindle_speed</Source>
352                         </DataItem>
353                         <DataItem category="EVENT" id="Hurco01-C_2" name="
           ↪ Spindle_Override" nativeUnits="PERCENT" type="
           ↪ ROTARY_VELOCITY_OVERRIDE" units="PERCENT">
354                             <Source>SspeedOvr</Source>
355                         </DataItem>
356                     </DataItems>
357                 </Rotary>
358             </Components>
359         </Axes>
360     <Controller id="Hurco01-controller" name="controller">
361         <DataItems>
362             <DataItem category="EVENT" id="Hurco01-controller_1" name="
           ↪ EStop_State" type="EMERGENCY_STOP"/>
363         </DataItems>
364     <Components>
365         <Path id="Hurco01-path" name="Path">

```

```

366     <DataItems>
367         <DataItem category="EVENT" id="Hurco01-path_1" name="
           ↳ Block_Number" type="LINE"/>
368         <DataItem category="EVENT" id="Hurco01-path_2" name="
           ↳ Program_Name_Running" type="PROGRAM"/>
369         <DataItem category="EVENT" id="Hurco01-path_3" name="
           ↳ Program_Name_Editing" type="PROGRAM_EDIT_NAME"/>
370         <DataItem category="EVENT" id="Hurco01-path_4" name="
           ↳ Part_Count" subType="ALL" type="PART_COUNT"/>
371         <DataItem category="EVENT" id="Hurco01-path_5" name="
           ↳ Program_Status" type="EXECUTION"/>
372         <DataItem category="EVENT" id="Hurco01-path_6" name="
           ↳ Current_Tool" type="TOOL_NUMBER"/>
373         <DataItem category="SAMPLE" id="Hurco01-path_7" name="
           ↳ Program_Runtime_Seconds" nativeUnits="SECOND" type="
           ↳ ACCUMULATED_TIME" units="SECOND"/>
374         <DataItem category="SAMPLE" id="Hurco01-path_8" name="
           ↳ Spindle_Time" nativeUnits="SECOND" type="
           ↳ ACCUMULATED_TIME" units="SECOND"/>
375         <DataItem category="SAMPLE" id="Hurco01-path_9" name="
           ↳ Feed_Rate" nativeUnits="MILLIMETER/SECOND" subType="
           ↳ ACTUAL" type="PATH_FEEDRATE" units="MILLIMETER/SECOND
           ↳ "/>
376         <DataItem category="SAMPLE" id="Hurco01-path_10" name="
           ↳ Feed_Override" nativeUnits="PERCENT" type="
           ↳ PATH_FEEDRATE_OVERRIDE" units="PERCENT"/>
377         <DataItem category="SAMPLE" id="Hurco01-path_11" name="
           ↳ Rapid_Override" nativeUnits="PERCENT" subType="RAPID"
           ↳ type="PATH_FEEDRATE_OVERRIDE" units="PERCENT"/>
378     </DataItems>
379 </Path>
380 </Components>
381 </Controller>
382 </Components>
383 </Device>
384 <Device id="Hurco02" name="Hurco02" uuid="mtc_adapter006">
385     <Description manufacturer="Hurco" model="VMX24">Hurco VMX 24 #2</
           ↳ Description>
386     <DataItems>
387         <DataItem category="EVENT" id="Hurco02-dtop_1" name="avail" type="
           ↳ AVAILABILITY"/>
388         <DataItem category="EVENT" id="Hurco02_asset_chg" type="
           ↳ ASSET_CHANGED"/>
389         <DataItem category="EVENT" id="Hurco02_asset_rem" type="
           ↳ ASSET_REMOVED"/>
390     </DataItems>

```

```

391 <Components>
392   <Axes id="Hurco02-axes" name="Axes">
393     <Components>
394       <Rotary id="Hurco02-C" name="C">
395         <DataItems>
396           <DataItem category="SAMPLE" id="Hurco02-C_1" name="
             ↳ Spindle_Speed" nativeUnits="REVOLUTION/MINUTE"
             ↳ subType="ACTUAL" type="ROTARY_VELOCITY" units="
             ↳ REVOLUTION/MINUTE">
397           <Source>spindle_speed</Source>
398         </DataItem>
399         <DataItem category="EVENT" id="Hurco02-C_2" name="
             ↳ Spindle_Override" nativeUnits="PERCENT" type="
             ↳ ROTARY_VELOCITY_OVERRIDE" units="PERCENT">
400         <Source>SspeedOvr</Source>
401         </DataItem>
402       </DataItems>
403     </Rotary>
404   </Components>
405 </Axes>
406 <Controller id="Hurco02-controller" name="controller">
407   <DataItems>
408     <DataItem category="EVENT" id="Hurco02-controller_1" name="
             ↳ EStop_State" type="EMERGENCY_STOP"/>
409   </DataItems>
410   <Components>
411     <Path id="Hurco02-path" name="Path">
412       <DataItems>
413         <DataItem category="EVENT" id="Hurco02-path_1" name="
             ↳ Block_Number" type="LINE"/>
414         <DataItem category="EVENT" id="Hurco02-path_2" name="
             ↳ Program_Name_Running" type="PROGRAM"/>
415         <DataItem category="EVENT" id="Hurco02-path_3" name="
             ↳ Program_Name_Editing" type="PROGRAM_EDIT_NAME"/>
416         <DataItem category="EVENT" id="Hurco02-path_4" name="
             ↳ Part_Count" subType="ALL" type="PART_COUNT"/>
417         <DataItem category="EVENT" id="Hurco02-path_5" name="
             ↳ Program_Status" type="EXECUTION"/>
418         <DataItem category="EVENT" id="Hurco02-path_6" name="
             ↳ Current_Tool" type="TOOL_NUMBER"/>
419         <DataItem category="SAMPLE" id="Hurco02-path_7" name="
             ↳ Program_Runtime_Seconds" nativeUnits="SECOND" type="
             ↳ ACCUMULATED_TIME" units="SECOND"/>
420         <DataItem category="SAMPLE" id="Hurco02-path_8" name="
             ↳ Spindle_Time" nativeUnits="SECOND" type="
             ↳ ACCUMULATED_TIME" units="SECOND"/>

```

```

421         <DataItem category="SAMPLE" id="Hurco02-path_9" name="
           ↪ Feed_Rate" nativeUnits="MILLIMETER/SECOND" subType="
           ↪ ACTUAL" type="PATH_FEEDRATE" units="MILLIMETER/SECOND
           ↪ "/>
422         <DataItem category="SAMPLE" id="Hurco02-path_10" name="
           ↪ Feed_Override" nativeUnits="PERCENT" type="
           ↪ PATH_FEEDRATE_OVERRIDE" units="PERCENT"/>
423         <DataItem category="SAMPLE" id="Hurco02-path_11" name="
           ↪ Rapid_Override" nativeUnits="PERCENT" subType="RAPID"
           ↪ type="PATH_FEEDRATE_OVERRIDE" units="PERCENT"/>
424     </DataItems>
425 </Path>
426 </Components>
427 </Controller>
428 </Components>
429 </Device>
430 <Device id="Hurco03" name="Hurco03" uuid="mtc_adapter007">
431     <Description manufacturer="Hurco" model="VMX24">Hurco VMX 24 #3</
           ↪ Description>
432     <DataItems>
433         <DataItem category="EVENT" id="Hurco03-dtop_1" name="avail" type="
           ↪ AVAILABILITY"/>
434         <DataItem category="EVENT" id="Hurco03_asset_chg" type="
           ↪ ASSET_CHANGED"/>
435         <DataItem category="EVENT" id="Hurco03_asset_rem" type="
           ↪ ASSET_REMOVED"/>
436     </DataItems>
437     <Components>
438         <Axes id="Hurco03-axes" name="Axes">
439             <Components>
440                 <Rotary id="Hurco03-C" name="C">
441                     <DataItems>
442                         <DataItem category="SAMPLE" id="Hurco03-C_1" name="
                           ↪ Spindle_Speed" nativeUnits="REVOLUTION/MINUTE"
                           ↪ subType="ACTUAL" type="ROTARY_VELOCITY" units="
                           ↪ REVOLUTION/MINUTE">
443                             <Source>spindle_speed</Source>
444                         </DataItem>
445                         <DataItem category="EVENT" id="Hurco03-C_2" name="
                           ↪ Spindle_Override" nativeUnits="PERCENT" type="
                           ↪ ROTARY_VELOCITY_OVERRIDE" units="PERCENT">
446                             <Source>SspeedOvr</Source>
447                         </DataItem>
448                     </DataItems>
449                 </Rotary>
450             </Components>

```

```

451 </Axes>
452 <Controller id="Hurco03-controller" name="controller">
453   <DataItems>
454     <DataItem category="EVENT" id="Hurco03-controller_1" name="
         ↪ EStop_State" type="EMERGENCY_STOP"/>
455   </DataItems>
456   <Components>
457     <Path id="Hurco03-path" name="Path">
458       <DataItems>
459         <DataItem category="EVENT" id="Hurco03-path_1" name="
             ↪ Block_Number" type="LINE"/>
460         <DataItem category="EVENT" id="Hurco03-path_2" name="
             ↪ Program_Name_Running" type="PROGRAM"/>
461         <DataItem category="EVENT" id="Hurco03-path_3" name="
             ↪ Program_Name_Editing" type="PROGRAM_EDIT_NAME"/>
462         <DataItem category="EVENT" id="Hurco03-path_4" name="
             ↪ Part_Count" subType="ALL" type="PART_COUNT"/>
463         <DataItem category="EVENT" id="Hurco03-path_5" name="
             ↪ Program_Status" type="EXECUTION"/>
464         <DataItem category="EVENT" id="Hurco03-path_6" name="
             ↪ Current_Tool" type="TOOL_NUMBER"/>
465         <DataItem category="SAMPLE" id="Hurco03-path_7" name="
             ↪ Program_Runtime_Seconds" nativeUnits="SECOND" type="
             ↪ ACCUMULATED_TIME" units="SECOND"/>
466         <DataItem category="SAMPLE" id="Hurco03-path_8" name="
             ↪ Spindle_Time" nativeUnits="SECOND" type="
             ↪ ACCUMULATED_TIME" units="SECOND"/>
467         <DataItem category="SAMPLE" id="Hurco03-path_9" name="
             ↪ Feed_Rate" nativeUnits="MILLIMETER/SECOND" subType="
             ↪ ACTUAL" type="PATH_FEEDRATE" units="MILLIMETER/SECOND
             ↪ "/>
468         <DataItem category="SAMPLE" id="Hurco03-path_10" name="
             ↪ Feed_Override" nativeUnits="PERCENT" type="
             ↪ PATH_FEEDRATE_OVERRIDE" units="PERCENT"/>
469         <DataItem category="SAMPLE" id="Hurco03-path_11" name="
             ↪ Rapid_Override" nativeUnits="PERCENT" subType="RAPID"
             ↪ type="PATH_FEEDRATE_OVERRIDE" units="PERCENT"/>
470       </DataItems>
471     </Path>
472   </Components>
473 </Controller>
474 </Components>
475 </Device>
476 <Device id="Hurco04" name="Hurco04" uuid="mtc_adapter008">
477   <Description manufacturer="Hurco" model="VMX24">Hurco VMX 24 #4</
         ↪ Description>

```

```

478 <DataItems>
479   <DataItem category="EVENT" id="Hurco04-dtop_1" name="avail" type="
      ↪ AVAILABILITY"/>
480   <DataItem category="EVENT" id="Hurco04_asset_chg" type="
      ↪ ASSET_CHANGED"/>
481   <DataItem category="EVENT" id="Hurco04_asset_rem" type="
      ↪ ASSET_REMOVED"/>
482 </DataItems>
483 <Components>
484   <Axes id="Hurco04-axes" name="Axes">
485     <Components>
486       <Rotary id="Hurco04-C" name="C">
487         <DataItems>
488           <DataItem category="SAMPLE" id="Hurco04-C_1" name="
              ↪ Spindle_Speed" nativeUnits="REVOLUTION/MINUTE"
              ↪ subType="ACTUAL" type="ROTARY_VELOCITY" units="
              ↪ REVOLUTION/MINUTE">
489             <Source>spindle_speed</Source>
490           </DataItem>
491           <DataItem category="EVENT" id="Hurco04-C_2" name="
              ↪ Spindle_Override" nativeUnits="PERCENT" type="
              ↪ ROTARY_VELOCITY_OVERRIDE" units="PERCENT">
492             <Source>SspeedOvr</Source>
493           </DataItem>
494         </DataItems>
495       </Rotary>
496     </Components>
497   </Axes>
498   <Controller id="Hurco04-controller" name="controller">
499     <DataItems>
500       <DataItem category="EVENT" id="Hurco04-controller_1" name="
          ↪ EStop_State" type="EMERGENCY_STOP"/>
501     </DataItems>
502     <Components>
503       <Path id="Hurco04-path" name="Path">
504         <DataItems>
505           <DataItem category="EVENT" id="Hurco04-path_1" name="
              ↪ Block_Number" type="LINE"/>
506           <DataItem category="EVENT" id="Hurco04-path_2" name="
              ↪ Program_Name_Running" type="PROGRAM"/>
507           <DataItem category="EVENT" id="Hurco04-path_3" name="
              ↪ Program_Name_Editing" type="PROGRAM_EDIT_NAME"/>
508           <DataItem category="EVENT" id="Hurco04-path_4" name="
              ↪ Part_Count" subType="ALL" type="PART_COUNT"/>
509           <DataItem category="EVENT" id="Hurco04-path_5" name="
              ↪ Program_Status" type="EXECUTION"/>

```

```

510     <DataItem category="EVENT" id="Hurco04-path_6" name="
        ↪ Current_Tool" type="TOOL_NUMBER"/>
511     <DataItem category="SAMPLE" id="Hurco04-path_7" name="
        ↪ Program_Runtime_Seconds" nativeUnits="SECOND" type="
        ↪ ACCUMULATED_TIME" units="SECOND"/>
512     <DataItem category="SAMPLE" id="Hurco04-path_8" name="
        ↪ Spindle_Time" nativeUnits="SECOND" type="
        ↪ ACCUMULATED_TIME" units="SECOND"/>
513     <DataItem category="SAMPLE" id="Hurco04-path_9" name="
        ↪ Feed_Rate" nativeUnits="MILLIMETER/SECOND" subType="
        ↪ ACTUAL" type="PATH_FEEDRATE" units="MILLIMETER/SECOND
        ↪ "/>
514     <DataItem category="SAMPLE" id="Hurco04-path_10" name="
        ↪ Feed_Override" nativeUnits="PERCENT" type="
        ↪ PATH_FEEDRATE_OVERRIDE" units="PERCENT"/>
515     <DataItem category="SAMPLE" id="Hurco04-path_11" name="
        ↪ Rapid_Override" nativeUnits="PERCENT" subType="RAPID"
        ↪ type="PATH_FEEDRATE_OVERRIDE" units="PERCENT"/>
516     </DataItems>
517     </Path>
518     </Components>
519     </Controller>
520     </Components>
521 </Device>
522 <Device id="Hurco05" name="Hurco05" uuid="mtc_adapter009">
523     <Description manufacturer="Hurco" model="VMX42">Hurco VMX 42</
        ↪ Description>
524     <DataItems>
525         <DataItem category="EVENT" id="Hurco05-dtop_1" name="avail" type="
            ↪ AVAILABILITY"/>
526         <DataItem category="EVENT" id="Hurco05_asset_chg" type="
            ↪ ASSET_CHANGED"/>
527         <DataItem category="EVENT" id="Hurco05_asset_rem" type="
            ↪ ASSET_REMOVED"/>
528     </DataItems>
529     <Components>
530         <Axes id="Hurco05-axes" name="Axes">
531             <Components>
532                 <Rotary id="Hurco05-C" name="C">
533                     <DataItems>
534                         <DataItem category="SAMPLE" id="Hurco05-C_1" name="
                            ↪ Spindle_Speed" nativeUnits="REVOLUTION/MINUTE"
                            ↪ subType="ACTUAL" type="ROTARY_VELOCITY" units="
                            ↪ REVOLUTION/MINUTE">
535                             <Source>spindle_speed</Source>
536                         </DataItem>

```

```

537         <DataItem category="EVENT" id="Hurco05-C_2" name="
           ↳ Spindle_Override" nativeUnits="PERCENT" type="
           ↳ ROTARY_VELOCITY_OVERRIDE" units="PERCENT">
538         <Source>SspeedOvr</Source>
539     </DataItem>
540 </DataItems>
541 </Rotary>
542 </Components>
543 </Axes>
544 <Controller id="Hurco05-controller" name="controller">
545     <DataItems>
546         <DataItem category="EVENT" id="Hurco05-controller_1" name="
           ↳ EStop_State" type="EMERGENCY_STOP"/>
547     </DataItems>
548     <Components>
549         <Path id="Hurco05-path" name="Path">
550             <DataItems>
551                 <DataItem category="EVENT" id="Hurco05-path_1" name="
                   ↳ Block_Number" type="LINE"/>
552                 <DataItem category="EVENT" id="Hurco05-path_2" name="
                   ↳ Program_Name_Running" type="PROGRAM"/>
553                 <DataItem category="EVENT" id="Hurco05-path_3" name="
                   ↳ Program_Name_Editing" type="PROGRAM_EDIT_NAME"/>
554                 <DataItem category="EVENT" id="Hurco05-path_4" name="
                   ↳ Part_Count" subType="ALL" type="PART_COUNT"/>
555                 <DataItem category="EVENT" id="Hurco05-path_5" name="
                   ↳ Program_Status" type="EXECUTION"/>
556                 <DataItem category="EVENT" id="Hurco05-path_6" name="
                   ↳ Current_Tool" type="TOOL_NUMBER"/>
557                 <DataItem category="SAMPLE" id="Hurco05-path_7" name="
                   ↳ Program_Runtime_Seconds" nativeUnits="SECOND" type="
                   ↳ ACCUMULATED_TIME" units="SECOND"/>
558                 <DataItem category="SAMPLE" id="Hurco05-path_8" name="
                   ↳ Spindle_Time" nativeUnits="SECOND" type="
                   ↳ ACCUMULATED_TIME" units="SECOND"/>
559                 <DataItem category="SAMPLE" id="Hurco05-path_9" name="
                   ↳ Feed_Rate" nativeUnits="MILLIMETER/SECOND" subType="
                   ↳ ACTUAL" type="PATH_FEEDRATE" units="MILLIMETER/SECOND
                   ↳ "/>
560                 <DataItem category="SAMPLE" id="Hurco05-path_10" name="
                   ↳ Feed_Override" nativeUnits="PERCENT" type="
                   ↳ PATH_FEEDRATE_OVERRIDE" units="PERCENT"/>
561                 <DataItem category="SAMPLE" id="Hurco05-path_11" name="
                   ↳ Rapid_Override" nativeUnits="PERCENT" subType="RAPID"
                   ↳ type="PATH_FEEDRATE_OVERRIDE" units="PERCENT"/>
562             </DataItems>

```



```

563     </Path>
564   </Components>
565 </Controller>
566 </Components>
567 </Device>
568 <Device id="Hurco06" name="Hurco06" uuid="mtc_adapter010">
569   <Description manufacturer="Hurco" model="VMX64">Hurco VMX 64</
    ↪ Description>
570   <DataItems>
571     <DataItem category="EVENT" id="Hurco06-dtop_1" name="avail" type="
    ↪ AVAILABILITY"/>
572     <DataItem category="EVENT" id="Hurco06_asset_chg" type="
    ↪ ASSET_CHANGED"/>
573     <DataItem category="EVENT" id="Hurco06_asset_rem" type="
    ↪ ASSET_REMOVED"/>
574   </DataItems>
575   <Components>
576     <Axes id="Hurco06-axes" name="Axes">
577       <Components>
578         <Rotary id="Hurco06-C" name="C">
579           <DataItems>
580             <DataItem category="SAMPLE" id="Hurco06-C_1" name="
    ↪ Spindle_Speed" nativeUnits="REVOLUTION/MINUTE"
    ↪ subType="ACTUAL" type="ROTARY_VELOCITY" units="
    ↪ REVOLUTION/MINUTE">
581             <Source>spindle_speed</Source>
582             </DataItem>
583             <DataItem category="EVENT" id="Hurco06-C_2" name="
    ↪ Spindle_Override" nativeUnits="PERCENT" type="
    ↪ ROTARY_VELOCITY_OVERRIDE" units="PERCENT">
584             <Source>SspeedOvr</Source>
585             </DataItem>
586           </DataItems>
587         </Rotary>
588       </Components>
589     </Axes>
590     <Controller id="Hurco06-controller" name="controller">
591       <DataItems>
592         <DataItem category="EVENT" id="Hurco06-controller_1" name="
    ↪ EStop_State" type="EMERGENCY_STOP"/>
593       </DataItems>
594       <Components>
595         <Path id="Hurco06-path" name="Path">
596           <DataItems>
597             <DataItem category="EVENT" id="Hurco06-path_1" name="
    ↪ Block_Number" type="LINE"/>

```

```

598     <DataItem category="EVENT" id="Hurco06-path_2" name="
        ↪ Program_Name_Running" type="PROGRAM"/>
599     <DataItem category="EVENT" id="Hurco06-path_3" name="
        ↪ Program_Name_Editing" type="PROGRAM_EDIT_NAME"/>
600     <DataItem category="EVENT" id="Hurco06-path_4" name="
        ↪ Part_Count" subType="ALL" type="PART_COUNT"/>
601     <DataItem category="EVENT" id="Hurco06-path_5" name="
        ↪ Program_Status" type="EXECUTION"/>
602     <DataItem category="EVENT" id="Hurco06-path_6" name="
        ↪ Current_Tool" type="TOOL_NUMBER"/>
603     <DataItem category="SAMPLE" id="Hurco06-path_7" name="
        ↪ Program_Runtime_Seconds" nativeUnits="SECOND" type="
        ↪ ACCUMULATED_TIME" units="SECOND"/>
604     <DataItem category="SAMPLE" id="Hurco06-path_8" name="
        ↪ Spindle_Time" nativeUnits="SECOND" type="
        ↪ ACCUMULATED_TIME" units="SECOND"/>
605     <DataItem category="SAMPLE" id="Hurco06-path_9" name="
        ↪ Feed_Rate" nativeUnits="MILLIMETER/SECOND" subType="
        ↪ ACTUAL" type="PATH_FEEDRATE" units="MILLIMETER/SECOND
        ↪ "/>
606     <DataItem category="SAMPLE" id="Hurco06-path_10" name="
        ↪ Feed_Override" nativeUnits="PERCENT" type="
        ↪ PATH_FEEDRATE_OVERRIDE" units="PERCENT"/>
607     <DataItem category="SAMPLE" id="Hurco06-path_11" name="
        ↪ Rapid_Override" nativeUnits="PERCENT" subType="RAPID"
        ↪ type="PATH_FEEDRATE_OVERRIDE" units="PERCENT"/>
608     </DataItems>
609     </Path>
610     </Components>
611     </Controller>
612     </Components>
613     </Device>
614     </Devices>
615 </MTConnectDevices>

```

Appendix B: Agent Configuration File

Listing 2 is an example of an Agent configuration file for collecting data from a hypothetical *Machine01* that has an Adapter providing data through 192.168.1.1:7878. The relevant files for the Agent are in the directory `/path/to/agent` including the `Devices` XML file. The appropriate schemas and styles are in the subdirectories `/schemas` and `/styles`, respectively.

Listing 2. agent.cfg

```

1 Devices = /path/to/agent/Devices.xml
2 AllowPut = true

```

```

3 ReconnectInterval = 1000
4 BufferSize = 20
5 SchemaVersion = 1.3
6 Port = 5000 # default port number for agent
7 UppcaseDataItemValue = false
8 MonitorConfigFiles = true # True = reload devices.xml and agent.cfg if
    ↪ they are changed
9
10 Adapters {
11     Machine01 {
12         Host = 192.168.1.1
13         Port = 7878
14     }
15 }
16
17 Files {
18     schemas {
19         Path = /path/to/agent/schemas
20         Location = /schemas/
21     }
22     styles {
23         Path = /path/to/agent/styles
24         Location = /styles/
25     }
26     Favicon {
27         Path = /path/to/agent/styles/favicon.ico
28         Location = /favicon.ico
29     }
30 }
31
32 StreamsStyle {
33     Location = /styles/Streams.xml
34 }
35
36 DevicesStyle {
37     Location = /styles/Devices.xml
38 }
39
40 # Logger Configuration
41 logger_config
42 {
43     logging_level = info
44     output = file /var/log/mtc_agent.log
45 }

```

Appendix C: MTConnect Agent Service

Listing 3 is an example configuration file that would be used to create an MTConnect Agent service. The installed and built Agent is located at `/usr/local/bin`, which is the default install location for a Linux operating system, and the Agent configuration file is located at `/path/to/agent`.

Listing 3. `mtc_agent.conf`

```
1 start on (net-device-up
2         and local-fileSYSTEMS
3         and runlevel [2345])
4 stop on runlevel [016]
5
6 exec /usr/local/bin/agent run /path/to/agent/agent.cfg
7
8 respawn
9 respawn limit 10 600
```

Appendix D: Data Collection Service

Listing 4 is an example configuration file that would be used to create a data collection service for a hypothetical *Machine01*. The MTConnect Agent for *Machine01* can be accessed at the IP address 192.168.1.1:5000. The files associated with this data collection service are located in the directory `/path/to/data_collection`. The required open-source tool `dump_agent.rb` is located in the `/tools` subdirectory, and the data collected is stored in the `/current` subdirectory in a text file specific for *Machine01* named `Machine01-current.txt`.

Listing 4. `dump_agent_Machine01.conf`

```
1 start on (net-device-up
2         and local-fileSYSTEMS
3         and runlevel [2345])
4 stop on runlevel [016]
5
6 exec ruby /path/to/data_collection/tools/dump_agent.rb http
   ↪ ://192.168.1.1:5000/Machine01 /path/to/data_collection/current/
   ↪ Machine01-current.txt
7
8 respawn
9 respawn limit 10 600
```

Appendix E: Data Re-Start and Archive bash Script

Listing 5 is an example bash script that can be used to re-start the data collection service described in Appendix D and archive data in the Data Aggregator to `/path/to/data_archive`.

The collected data files in `/path/to/data_archive` are organized into different subdirectories for each piece of equipment. So for the hypothetical *Machine01*, archived data would be stored in the `/Machine01` subdirectory.

Listing 5. `generic_recorder.sh`

```
1 #!/bin/bash
2
3 today=$(date +%Y%m%d)
4 yesterday=$(date +%Y%m%d -d yesterday)
5
6 filename=$1'-'$yesterday'-'$today'.txt'
7
8 stop dump_agent_$1
9 mv /path/to/data_collection/current/$1-current.txt /path/to/data_archive/
   ↪ $1/$filename
10 start dump_agent_$1
```

Appendix F: Adapter Simulator Service

Listing 6 is an example configuration file that would be used to create an adapter simulator service for a hypothetical machine *Machine01*. The files associated with this adapter simulator service are located in the directory `/path/to/adapter_simulator`. The required open-source tool `run_scenario.rb` is located in this directory, and the collected data file to be replayed for *Machine01*, named `mtc_adapter_Machine01.txt`, is located in the `/data` subdirectory.

Listing 6. `mtc_adapter_Machine01.conf`

```
1 start on (net-device-up
2         and local-fileSYSTEMS
3         and runlevel [2345])
4 stop on runlevel [016]
5
6 exec ruby /path/to/adapter_simulator/run_scenario.rb -l -p 7878 /path/to/
   ↪ adapter_simulator/data/mtc_adapter001.txt
7
8 respawn
9 respawn limit 10 600
```

Appendix G: QDR Bulk Data Import Generator

Listing 7 is an example Python script that can be used to generate a set of parsed MTConnect XML documents for bulk uploading to the QDR and pushing to a Git repository for

archiving. The collected data files in /path/to/local/data_archive are organized into different subdirectories for each piece of equipment. The data is the raw output captured by generic_recorder.sh described in Appendix E. So for the hypothetical *Machine01*, archived data would be stored in the /Machine01 subdirectory. We found that QDR performance is maximized when data input blocs are limited to 10,000 lines. The limit is set on Line 50 in Listing 7.

Listing 7. data_bulkQDRinput.py

```

1 #!/usr/bin/env python
2
3 import os, datetime, subprocess, shutil, tempfile, urllib.request,
   ↪ threading, time
4 import xml.etree.ElementTree
5 from datetime import date, timedelta
6 from itertools import zip_longest
7
8 # collect data into fixed-length chunks or blocks
9 def grouper(n, iterable, fillvalue=None):
10  # grouper(3, 'ABCDEFG', 'x') --> ABC DEF Gxx
11  args = [iter(iterable)] * n
12  return zip_longest(fillvalue=fillvalue, *args)
13
14 class agentThread (threading.Thread):
15  def __init__(self, threadID, name, config):
16  super(agentThread, self).__init__()
17  self.threadID = threadID
18  self.name = name
19  self.config = config
20  self._stop = threading.Event()
21  def run(self):
22  print('Starting ' + self.name)
23  start_agent(self.name, self.config)
24  print('Exiting ' + self.name)
25  def stop(self):
26  self._stop.set()
27  print('Exiting ' + self.name)
28  def stopped(self):
29  return self._stop.isSet()
30
31 # define the agent function for the thread
32 def start_agent(threadName, config_file):
33  cmd='agent run '+config_file
34  p=subprocess.Popen(cmd, shell=True, stdout=subprocess.PIPE)
35  output, errors = p.communicate()
36
37 # set log path

```

```

38 log_path = '/path/to/qdr_data.log'
39 # set path of local git repo
40 git_path = '/path/to/local/git/repo/'
41 # set path of local data archive
42 data_path = '/path/to/local/data_archive/'
43 # set path of local temp directory
44 tmp_path = '/path/to/local/temp/'
45 # set path of agent configs
46 config_path = '/path/to/agent/configs/'
47 # set machine UUIDs and adapter data file name
48 machine_uuid = ('empty', 'Machine01', 'Machine02')
49 # define qdr input block size
50 n = 10000
51 # initialize qdr input index
52 index = 1
53 # set url for agent
54 agent_url = 'http://localhost:5000/'
55
56 # log start
57 with open(log_path, 'a') as out_file:
58     out_file.write('Start Data Parsing: %s\n' % datetime.datetime.now().
        ↪ strftime("%Y-%m-%d %H:%M:%S"))
59
60 if __name__ == '__main__':
61     print ("Starting")
62     print ("File: %s" % os.path.dirname(__file__))
63
64 # sync git repo
65 subprocess.call(['git', '--git-dir', git_path + '/.git', '--work-tree',
        ↪ git_path, 'pull'])
66
67 # set the git repo directory for qdr input data
68 qdr_git_path = os.path.join(git_path, 'raw/qdr_input/')
69 print ("QDR Input Data Git Path: %s" % qdr_git_path )
70
71 ## remove old qdr input data
72 for the_file in os.listdir(git_path):
73     file_path = os.path.join(git_path, the_file)
74     if os.path.isfile(file_path):
75         os.remove(file_path)
76
77 # copy and add data to git repo
78 for x in range(2,3):
79     seqFirst = 0 # reset sequence value
80
81 # stop old agents

```

```

82  if 'thread1' in locals():
83      thread1.stop()
84      time.sleep(10)
85
86  # remove old temp data
87  for the_file in os.listdir(tmp_path+machine_uuid[x]+'/''):
88      file_path = os.path.join(tmp_path+machine_uuid[x]+'/', the_file)
89      if os.path.isfile(file_path):
90          os.remove(file_path)
91
92  # parse the raw data by walking the directories
93  for in_file in os.listdir(data_path+machine_uuid[x]+"/"):
94      # set adapter data
95      data_file = os.path.join(data_path+machine_uuid[x]+"/",in_file)
96      print ("Current Data Archive File: %s" % data_file )
97
98      # check if data file exists
99      if not os.path.exists(data_file):
100         print ("Current Data does not exist. Expected: %s" % data_file )
101     else:
102         # split adapter files into chunks
103         with open(data_file) as f:
104             for i, g in enumerate(grouper(n, f, fillvalue=None)):
105                 with tempfile.NamedTemporaryFile('w', delete=False) as fout:
106                     for j, line in enumerate(g, 1): # count number of lines in
107                         ↪ group
108                         if line is None:
109                             j -= 1 # don't count this line
110                             break
111                         fout.write(line)
112                 short_file = os.path.join(tmp_path+machine_uuid[x]+'/', 'qdr_input
113                     ↪ {0:0=3d}'.format(index)+' .txt')
114                 print ("Short File: %s" % short_file )
115                 os.rename(fout.name, short_file)
116                 index += 1 # increment index up 1
117
118     # start agent
119     # create new threads
120     thread1 = agentThread(1, "Agent", config_path+machine_uuid[x]+' .cfg')
121     # set thread as daemon, no need to close the thread -- it will end
122     ↪ automatically when program ends
123     thread1.daemon = True
124
125     # generate qdr input files (output in MTConnect compliant XML)
126     for log_file in os.listdir(tmp_path+machine_uuid[x]+'/''):

```



```

125 log_path = os.path.join(tmp_path+machine_uuid[x]+'/', log_file)
126
127 # stop old agents
128 if 'thread1' in locals():
129     thread1.stop()
130     time.sleep(10)
131
132 # start new agent
133 thread1.start()
134
135 if os.path.splitext(log_file)[1] == '.txt':
136     print('Log File: %s' % log_path )
137
138 # start adapter
139 cmd='ruby /path/to/adapter_simulator/run_sample.rb '+log_path
140 p=subprocess.Popen(cmd, shell=True, stdout=subprocess.PIPE)
141 output, errors = p.communicate()
142
143 # determine first and last sequence
144 e = xml.etree.ElementTree.fromstring(urllib.request.urlopen(
145     ↪ agent_url+'current').read().decode("utf-8"))
146 for aHeader in e.findall('{urn:mtconnect.org:MTConnectStreams:1.3}
147     ↪ Header'):
148     if seqFirst==0:
149         seqFirst = int(aHeader.get('firstSequence'))
150     else:
151         seqFirst = seqLast + 1
152         seqLast = int(aHeader.get('lastSequence'))
153
154 # generate qdr input file using curl
155 count = seqLast - seqFirst + 1
156 sample_xml = urllib.request.urlopen(agent_url+'sample?&count='+str(
157     ↪ count)+'&from='+str(seqFirst)).read().decode("utf-8")
158 qdr_file = os.path.join/git_path, os.path.splitext(log_file)[0]+'.'
159     ↪ xml')
160 with open(qdr_file, 'w') as output:
161     output.write(sample_xml)
162 print ("QDR Input File: %s" % qdr_file )
163
164 # Use subprocess module to push revised data to github.
165 # Need to set both the --git-dir and --work-tree
166 # http://stackoverflow.com/questions/1386291/git-git-dir-not-working
167     ↪ -as-expected
168 subprocess.call(['git', '--git-dir', git_path + '/.git', '--work-tree'
169     ↪ , git_path, 'add', '-f', qdr_file])

```

```

165 #commit and sync git repo
166 subprocess.call(['git', '--git-dir', git_path + '/.git', '--work-tree',
    ↪ git_path, 'commit', '-m', 'Data Upload: ' + datetime.datetime.now().
    ↪ strftime("%Y-%m-%d %H:%M:%S")])
167 subprocess.call(['git', '--git-dir', git_path + '/.git', '--work-tree',
    ↪ git_path, 'push'])
168 print("Data comitted and pushed to Git Repo")
169
170 # log finish
171 with open(log_path, 'a') as out_file:
172     out_file.write('Complete Data Upload: %s\n' % datetime.datetime.now().
    ↪ strftime("%Y-%m-%d %H:%M:%S"))
173
174 # stop agent
175 if 'thread1' in locals():
176     thread1.stop()
177     time.sleep(10)

```

Appendix H: QDR Bulk Data Upload

Listing 8 is an example Python script that can be used to bulk upload the parsed MTConnect XML documents from the data_bulkQDRinput.py script described in Appendix G. All XML documents stored in a provided directory will be uploaded by the script to the QDR using a REST API. The script requires the XML Schema Definition (XSD) for the documents being uploaded, a descriptive title for the schema, and the location of the documents being uploaded. If the schema already exists in the QDR under the same title, then QDR will use the stored schema.

Listing 8. QDR_bulkUpload.py

```

1 #!/usr/bin/env python
2
3 from os.path import join
4 from os import listdir
5 import argparse
6 import requests
7 import json
8
9 # Script parameters
10 from rest_framework.status import HTTP_201_CREATED
11
12 # The QDR Bulk Upload Script takes three arguments to upload data through
    ↪ the QDR REST API
13 # 1. '-x' or '--xsd' must equal the full path the XSD file for the data
    ↪ upload
14 # E.g., '/full/path/to/MTConnectStreams_1.3.xsd'

```

```

15 #
16 # 2. '-t' or '--title' must equal a title for the XSD file to be stored in
    ↪ the QDR. Use the same title for a XSD schema already in the QDR
    ↪ database.
17 # E.g., 'MTConnectStreams_1.3'
18 #
19 # 3. '-p' or '--path' must equal the full path to the directory containing
    ↪ the xml files to be uploaded to the QDR.
20 # E.g., '/full/path/to/xml/files/'
21
22 parser = argparse.ArgumentParser(description='Add xml files from a folder
    ↪ and their associated schema.')
23 parser.add_argument('-x', '--xsd', action='store', required=True, help='
    ↪ Full path to the XSD file.')
24 parser.add_argument('-t', '--title', action='store', required=True, help='
    ↪ Schema title use into the application.')
25 parser.add_argument('-p', '--path', action='store', required=True, help='
    ↪ Full path to the directory containing xml files.')
26
27 args = vars(parser.parse_args())
28 xsd_filename = args['xsd']
29 title = args['title']
30 xml_data_dirpath = args['path']
31
32 # QDR user login information
33 user_name = ""
34 user_pwd = ""
35
36 # URI for QDR API
37 qdr_url = "http://localhost:8000"
38 # Paths to API calls
39 template_upload_url = "/rest/templates/add"
40 template_url = "/rest/templates/select"
41 xml_upload_url = "/rest/curate"
42
43 xsd_data_filepath = join("xsd", xsd_filename)
44
45 url = qdr_url + template_url + "?title=" + title
46
47 response = requests.get(url, auth=(user_name, user_pwd))
48 response_code = response.status_code
49 response_content = json.loads(response.text)
50 if response_code == 200:
51     # Existing schema found in QDR. Using that one.
52     print title + " schema has been taken from the database."
53     template_id = response_content[0]['id']

```

```

54 else:
55     # Uploading the schema to QDR
56     with open(join(xsd_data_filepath), "r") as template_file:
57         template_content = template_file.read()
58
59         url = qdr_url + template_upload_url
60         data = {
61             "title": title,
62             "filename": title,
63             "content": template_content
64         }
65
66         response = requests.post(url, data, auth=(user_name, user_pwd))
67         response_code = response.status_code
68         response_content = json.loads(response.text)
69         if '_id' in response_content:
70             template_id = str(response_content['_id']['$oid'])
71         else:
72             raise Exception("A problem occurred when uploading the schema (
73                 ↪ Error " + response_content['message'] + ")")
74         if response_code == HTTP_201_CREATED:
75             print title + " schema has been uploaded."
76         else:
77             raise Exception("A problem occurred when uploading the schema (
78                 ↪ Error " + response_code + ")")
79
80 # Add xml files to QDR
81 for xml_filename in listdir(xml_data_dirpath):
82     # Filtering xml files in the directory
83     if xml_filename.endswith(".xml"):
84         with open(join(xml_data_dirpath, xml_filename), "r") as xml_file:
85             xml_content = xml_file.read()
86
87             data = {
88                 "title": xml_filename,
89                 "schema": template_id,
90                 "content": xml_content
91             }
92
93             url = qdr_url + xml_upload_url
94             response_str = requests.post(url, data, auth=(user_name,
95                 ↪ user_pwd))
96             response_code = response_str.status_code
97             response = json.loads(response_str.text)
98
99             if response_code == HTTP_201_CREATED:

```

```
97         print response["title"] + " has been uploaded"
98     else:
99         print "Upload failed with status code " + str(response_code
    ↪ ) + ": " + response["message"]
```
