MTConnect® Standard
Part 4.0 – Assets
Version 1.3.0

Prepared for: MTConnect Institute
Prepared by: William Sobel
Prepared on: September 30, 2014

MTConnect® Specification and Materials

Table of Contents

Table of Figures

1 Overview

MTConnect® is a standard based on an open protocol for data integration. MTConnect® is not intended to replace the functionality of existing products, but it strives to enhance the data acquisition capabilities of devices and applications and move toward a plug-and-play environment to reduce the cost of integration.

MTConnect® is built upon the most prevalent standards in the manufacturing and software industry, maximizing the number of tools available for its implementation and providing the highest level of interoperability with other standards and tools in these industries.

To facilitate this level of interoperability, a number of objectives are being met. Foremost is the ability to transfer data via a standard protocol which includes:

- A device identity (i.e. model number, serial number, calibration data, etc.).
- The identity of all the independent components of the device.
- Possibly a device's design characteristics (i.e. axis length, maximum speeds, device thresholds, etc.).
- Most importantly, data captured in real or near-real-time (i.e. current speed, position data, temperature data, program block, etc.) by a device that can be utilized by other devices or applications (e.g. utilized by maintenance diagnostic systems, management production information systems, CAM products, etc.).

The types of data that may need to be addressed in MTConnect® could include:

- Physical and actual device design data
- Measurement or calibration data
- Near-real-time data from the device

To accommodate the vast amount of different types of devices and information that may come into play, MTConnect® will provide a common high-level vocabulary and structure.

The first version of MTConnect® will focus on a limited set of the characteristics mentioned above that were selected based on the fact that they can have an immediate affect on the efficiency of operations.

1.1 MTConnect® Document Structure

The MTConnect® specification is subdivided using the following scheme:

Part 1: Overview and Protocol
Part 2: Components and Data Items
Part 3: Streams, Events, Samples, and Condition
Part 4.0: Assets
  Part 4.1: Cutting Tools

These four documents are considered the bases of the MTConnect standard. Information applicable to basic machine and device types will be included in these documents. Additional parts to the standard will be added to provide information and extensions to the standard focused on specific devices, components, or technologies considered requiring separate emphasis. All

information specific to the topic of each additional part **MUST** be included within that document even when it is a subject matter of one of the base parts of the standard.

Documents will be named (file name convention) as follows:

MTC_Part_<Number>_<Description>.doc.

For example, the file name for Part 2 of the standard is MTC_Part_2_Components.doc.

All documents will be developed in Microsoft® Word format and released in Adobe® PDF format.

2 Purpose of This Document

The four base MTConnect® documents are intended to:

- define the MTConnect® standard;
- specify the requirements for compliance with the MTConnect® standard;
- provide engineers with sufficient information to implement *Agents* for their devices;
- provide developers with the necessary guidelines to use the standard to develop applications.

Part 1 of the MTConnect Standard provides an overview of the MTConnect Architecture and Protocol; including communication, fault tolerance, connectivity, and error handling requirements.

Part 2 of the MTConnect® standard focuses on the data model and description of the information that is available from the device. The descriptive data defines how a piece of equipment should be modeled, the structure of the component hierarchy, the names for each component (if restricted), and allowable data items for each of the components.

Part 3 of the MTConnect standard focuses on the data returned from a `current` or `sample` request (for more information on these requests, see Part 1). This section covers the data representing the state of the machine.

Part 4 of the MTConnect® standard provides a semantic model for entities that are used in the manufacturing process, but are not considered to be a device nor a component. These entities are defined as MTConnect® Assets. These assets may be removed from a device without detriment to the function of the device, and can be associated with other devices during their lifecycle. The data associated with these assets will be retrieved from multiple sources that are responsible for providing their knowledge of the asset. The first type of asset to be addressed is Tooling.

2.1 Terminology

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

Agent A process that implements the MTConnect® HTTP protocol, XML generation, and MTConnect protocol.

Alarm An alarm indicates an event that requires attention and indicates a deviation from normal operation. Alarms are reported in MTConnect as `Condition`.

Application A process or set of processes that access the MTConnect® *Agent* to perform some task.

Attribute A part of an XML element that provides additional information about that XML element. For example, the `name` XML element of the `Device` is given as `<Device name="mill-1">...</Device>`

CDATA The text in a simple content element. For example, *`This is some text`*, in `<Message ...>This is some text</Message>`.

Component A part of a device that can have sub-components and data items. A component is a basic building block of a device.

Controlled Vocabulary The value of an element or attribute is limited to a restricted set of possibilities. Examples of controlled vocabularies are country codes: US, JP, CA, FR, DE, etc…

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

Data Item A data item provides the descriptive information regarding something that can be collected by the *Agent*.

Device A piece of equipment capable of performing an operation. A device may be composed of a set of components that provide data to the application. The device is a separate entity with at least one component or data item providing information about the device.

Discovery Discovery is a service that allows the application to locate *Agents* for devices in the manufacturing environment. The discovery service is also referred to as the *Name Service*.

Event An event represents a change in state that occurs at a point in time. Note: An event does not occur at predefined frequencies.

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

Instance When used in software engineering, the word *instance* is used to define a single physical example of that type. In object-oriented models, there is the class that describes the thing and the instance that is an example of that thing.

LDAP Lightweight Directory Access Protocol, better known as Active Directory in Microsoft Windows. This protocol provides resource location and contact information in a hierarchal structure.

MIME Multipurpose Internet Mail Extensions. A format used for encoding multipart mail and http content with separate sections separated by a fixed boundary.

Probe A request to determine the configuration and reporting capabilities of the device.

REST REpresentational State Transfer. A software architecture where the client and server move through a series of state transitions based solely on the request from the client and the response from the server.

Results A general term for the `Samples`, `Events`, and `Condition` contained in a `ComponentStream` as a response from a `sample` or `current` request.

Sample	A sample is a data point from within a continuous series of data points. An example of a Sample is the position of an axis.
Socket	When used concerning inter-process communication, it refers to a connection between two end-points (usually processes). Socket communication most often uses TCP/IP as the underlying protocol.
Stream	A collection of `Events, Samples,` and `Condition` organized by devices and components.
Service	An application that provides necessary functionality.
Tag	Used to reference an instance of an XML element.
TCP/IP	TCP/IP is the most prevalent stream-based protocol for inter-process communication. It is based on the IP stack (Internet Protocol) and provides the flow-control and reliable transmission layer on top of the IP routing infrastructure.
URI	Universal Resource Identifier. This is the official name for a web address as seen in the address bar of a browser.
UUID	Universally unique identifier.
XPath	XPath is a language for addressing parts of an XML Document. See the XPath specification for more information. http://www.w3.org/TR/xpath
XML	Extensible Markup Language. http://www.w3.org/XML/
XML Schema	The definition of the XML structure and vocabularies used in the XML Document.
XML Document	An instance of an XML Schema which has a single root XML element and conforms to the XML specification and schema.
XML Element	An element is the central building block of any XML Document. For example, in MTConnect® the Device XML element is specified as **`<Device>...</Device>`**
XML `NMTOKEN`	The data type for XML identifiers. It **MUST** start with a letter, an underscore "_" or a colon ":" and then it **MUST** be followed by a letter, a number, or one of the following ".", "-", "_", ":". An `NMTOKEN` cannot have any spaces or special characters.

2.2 Terminology and Conventions

Please refer to Part 1 "Overview and Protocol" Section 2 for XML Terminology and Documentation conventions.

3 Extension to Part 1, Overview and Protocol

As documented in Part 1, additional queries will be added to the *Agent* to support the storage and retrieval of assets. There is more detail in Part 1; what follows is a summary of the protocol additions:

Asset protocol:

- Request an asset by id:
  - `url: http://example.com/asset/hh1`
  - Returns the `MTConnectAssets` document for asset hh1
- Request multiple assets by id:
  - `url: http://example.com/asset/hh1;cc;123;g5`
  - Returns the `MTConnectAssets` document for asset hh1, cc, 123, and g5.
- Request for all the assets in the *Agent*:
  - `url: http://example.com/assets`
  - Returns all available MTConnect assets in the *Agent*. MTConnect **MAY** return a limited set if there are too many asset records. The assets **MUST** be added to the beginning with the most recently modified assets.
- Request for all assets of a given type in the *Agent*:
  - `url: http://example.com/assets?type=CuttingTool`
  - Returns all available `CuttingTool` assets from the MTConnect *Agent*. MTConnect **MAY** return a limited set if there are too many asset records. The assets **MUST** be added to the beginning with the most recently modified assets.
- Request for all assets of a given type in the *Agent* up to a maximum count:
  - `url: http://example.com/assets?type=CuttingTool&count=1000`
  - Returns all available `CuttingTool` assets from the MTConnect *Agent*. MTConnect **MUST** return up to 1000 assets beginning with the most recently modified assets if they exist.
- Request for all assets including assets that have been removed:
  - `url: http://example.com/assets?type=CuttingTool&removed=true`
  - Returns all available `CuttingTool` assets from the MTConnect *Agent*. With the removed flag, assets that have been removed but are included in the result set.

4 Extensions to Part 2, Components and Data Items

This document will add the following data item types to support change notification when an asset is added or updated. The data item **MUST** be placed in the `DataItems` collection of the top level device. The device **MUST** be the device that is supplying the asset data.

4.1 Data Item Types for `EVENT` Category

Data Item type/subtype	Description
`ASSET_CHANGED`	The value of the **CDATA** for the event **MUST** be the `assetId` of the asset that has been added or changed. There will not be a separate message for new assets.
`ASSET_REMOVED`	The value of the **CDATA** for the event **MUST** be the assetId of the asset that has been removed. The asset will still be visible if requested with the includeRemoved parameter as described in the protocol section. When assets are removed they are not moved to the beginning of the most recently modified list.

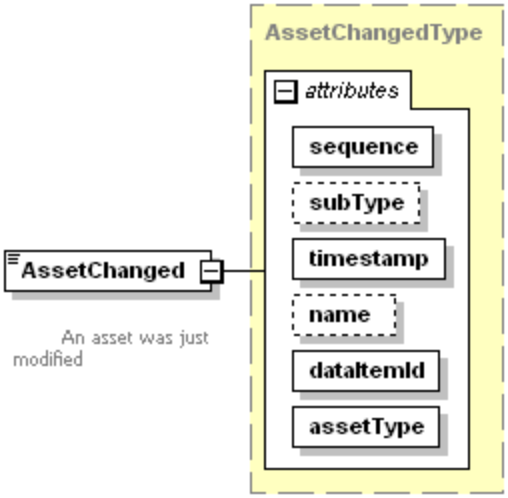
5 Extensions to Part 3, Streams, Events, Samples, and Condition

The associated modifications **MUST** be added to Part 3 to add the following event to the events in the streams.

5.1 Extension to `Events` section 3.9

The `AssetChanged` element extends the base `Event` type defined in *Part 3, Streams, Events, Samples, and Condition* and adds the `assetType` attribute to the base `Event`. This new event will signal whenever a new asset is added or the existing definition of an asset is updated. The asset Id is provide as the CDATA value and can be used to request the asset data from the *Agent* as described in *Part 1, Overview and Protocol.*

The `AssetRemoved` element extends the base `Event` type defined in *Part 3, Streams, Events, Samples, and Condition* and adds the `assetType` attribute to the base `Event`. This new event will signal whenever an asset has been removed from the agent. The asset will still be available if requested if the removed=true argument is suplied. The asset Id is provide as the CDATA value and can be used to request the asset data from the *Agent* as described in *Part 1, Overview and Protocol.*



`AssetChanged` An asset has been added or modified. The **CDATA** for the `AssetChanged` element **MUST** be the `assetId` of the asset that has been modified.

5.1.1 Additional `AssetChanged` attributes:

Attribute	Description	Occurrence
`assetType`	The type of asset that changed	1

`Add diagram…`

`AssetRemoved` An asset has been removed. The **CDATA** for the `AssetRemoved` element **MUST** be the `assetId` of the asset that has been removed.

5.1.2 Additional `AssetChanged` attributes:

Attribute	Description	Occurrence
`assetType`	The type of asset that changed	1

6 Assets

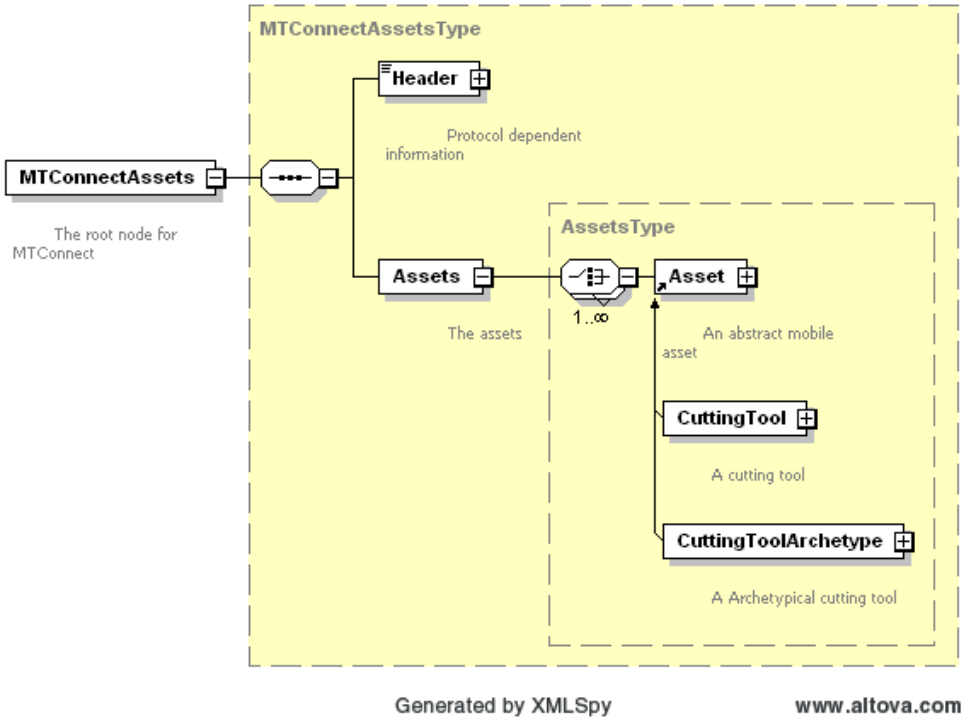


Figure 1: Assets Schema

An Asset is something that is associated with the manufacturing process that is not a component of a device, can be removed without detriment to the function of the device, and can be associated with other devices during their lifecycle. An asset does not have computational capabilities, but may carry information in some media physically attached to the asset.

Concrete examples of Assets are things like Cutting Tools, Workholding Systems, and Fixtures. Part 4 of the MTConnect standard will concern itself with the modeling of these assets and the management and communication of asset data using MTConnect.

At the top level of the MTConnectAssets document we have a standard header as documented in Part 1: Overview and Protocol and one or more assets. Each asset is required to have an assetId that serves as a unique identifier of that asset. The id allows the application to request the asset data from the agent, as prescribed in Part 1.

In the remaining document, we will be discussing Cutting Tools as the first asset type covered by the standard. The cutting tool must have an assetId that differs from all the other assets tracked by this agent. There **MUST** never be more than one asset provided by MTConnect with the same asset Id in the same agent.

Appendices

A. Bibliography

1. Engineering Industries Association. *EIA Standard - EIA-274-D*, Interchangeable Variable, Block Data Format for Positioning, Contouring, and Contouring/Positioning Numerically Controlled Machines. Washington, D.C. 1979.

2. ISO TC 184/SC4/WG3 N1089. *ISO/DIS 10303-238*: Industrial automation systems and integration Product data representation and exchange Part 238: Application Protocols: Application interpreted model for computerized numerical controllers. Geneva, Switzerland, 2004.

3. International Organization for Standardization. *ISO 14649*: Industrial automation systems and integration – Physical device control – Data model for computerized numerical controllers – Part 10: General process data. Geneva, Switzerland, 2004.

4. International Organization for Standardization. *ISO 14649*: Industrial automation systems and integration – Physical device control – Data model for computerized numerical controllers – Part 11: Process data for milling. Geneva, Switzerland, 2000.

5. International Organization for Standardization. *ISO 6983/1* – Numerical Control of machines – Program format and definition of address words – Part 1: Data format for positioning, line and contouring control systems. Geneva, Switzerland, 1982.

6. Electronic Industries Association. *ANSI/EIA-494-B-1992*, 32 Bit Binary CL (BCL) and 7 Bit ASCII CL (ACL) Exchange Input Format for Numerically Controlled Machines. Washington, D.C. 1992.

7. National Aerospace Standard. *Uniform Cutting Tests* - NAS Series: Metal Cutting Equipment Specifications. Washington, D.C. 1969.

8. International Organization for Standardization. *ISO 10303-11*: 1994, Industrial automation systems and integration Product data representation and exchange Part 11: Description methods: The EXPRESS language reference manual. Geneva, Switzerland, 1994.

9. International Organization for Standardization. *ISO 10303-21*: 1996, Industrial automation systems and integration -- Product data representation and exchange -- Part 21: Implementation methods: Clear text encoding of the exchange structure. Geneva, Switzerland, 1996.

10. H.L. Horton, F.D. Jones, and E. Oberg. *Machinery's handbook*. Industrial Press, Inc. New York, 1984.

11. International Organization for Standardization. *ISO 841-2001: Industrial automation systems and integration - Numerical control of machines - Coordinate systems and motion nomenclature*. Geneva, Switzerland, 2001.

12. *ASME B5.59-2 Version 9c: Data Specification for Properties of Machine Tools for Milling and Turning. 2005.*

13. *ASME/ANSI B5.54: Methods for Performance Evaluation of Computer Numerically Controlled Lathes and Turning Centers. 2005.*

14. OPC Foundation. *OPC Unified Architecture Specification, Part 1: Concepts Version 1.00. July 28, 2006.*

15. International Organization for Standardization. *ISO 13399*: *Cutting tool data representation and exchange*. Geneva, Switzerland, 2000.