MTConnect® Standard
Part 4.0 – Assets Information Model
Version 1.7.0

Prepared for: MTConnect Institute
Prepared on: February 25, 2021

MTConnect Specification and Materials

The Association for Manufacturing Technology (AMT) owns the copyright in this *MTConnect* Specification or Material. AMT grants to you a non-exclusive, non-transferable, revocable, non-sublicensable, fully-paid-up copyright license to reproduce, copy and redistribute this *MTConnect* Specification or Material, provided that you may only copy or redistribute the *MTConnect* Specification or Material in the form in which you received it, without modifications, and with all copyright notices and other notices and disclaimers contained in the *MTConnect* Specification or Material.

If you intend to adopt or implement an *MTConnect* Specification or Material in a product, whether hardware, software or firmware, which complies with an *MTConnect* Specification, you shall agree to the *MTConnect* Specification Implementer License Agreement ("Implementer License") or to the *MTConnect* Intellectual Property Policy and Agreement ("IP Policy"). The Implementer License and IP Policy each sets forth the license terms and other terms of use for *MTConnect* Implementers to adopt or implement the *MTConnect* Specifications, including certain license rights covering necessary patent claims for that purpose. These materials can be found at `www.MTConnect.org`, or or by contacting `mailto:info@MTConnect.org`.

MTConnect Institute and AMT have no responsibility to identify patents, patent claims or patent applications which may relate to or be required to implement a Specification, or to determine the legal validity or scope of any such patent claims brought to their attention. Each *MTConnect* Implementer is responsible for securing its own licenses or rights to any patent or other intellectual property rights that may be necessary for such use, and neither AMT nor *MTConnect* Institute have any obligation to secure any such rights.

This Material and all *MTConnect* Specifications and Materials are provided "as is" and *MTConnect* Institute and AMT, and each of their respective members, officers, affiliates, sponsors and agents, make no representation or warranty of any kind relating to these materials or to any implementation of the *MTConnect* Specifications or Materials in any product, including, without limitation, any expressed or implied warranty of noninfringement, merchantability, or fitness for particular purpose, or of the accuracy, reliability, or completeness of information contained herein. In no event shall *MTConnect* Institute or AMT be liable to any user or implementer of *MTConnect* Specifications or Materials for the cost of procuring substitute goods or services, lost profits, loss of use, loss of data or any incidental, consequential, indirect, special or punitive damages or other direct damages, whether under contract, tort, warranty or otherwise, arising in any way out of access, use or inability to use the *MTConnect* Specification or other *MTConnect* Materials, whether or not they had advance notice of the possibility of such damage.

Table of Contents

Table of Figures

List of Tables

1   1 Purpose of This Document

2   This document, *MTConnect Standard: Part 4.0 - Assets Information Model* of the MTCon-
3   nect Standard, details information that is common to all types of *MTConnect Assets*. Part
4   4.0 and its sub-parts of the MTConnect Standard provide semantic models for entities that
5   are used in the manufacturing process, but are not considered to be a piece of equipment.
6   These entities are defined as *MTConnect Assets*. These *Assets* may be removed from a
7   piece of equipment without detriment to the function of the equipment and can be associ-
8   ated with other pieces of equipment during their lifecycle. The data associated with these
9   *Assets* may be retrieved from multiple sources that are each responsible for providing their
10 knowledge of the *Asset*.

2 Terminology and Conventions

Refer to Section 2 of *MTConnect Standard Part 1.0 - Overview and Fundamentals* for a dictionary of terms, reserved language, and document conventions used in the MTConnect Standard.

2.1 Glossary

CDATA

> General meaning:

> An abbreviation for Character Data.

> CDATA is used to describe a value (text or data) published as part of an XML element.

> For example, `"This is some text"` is the CDATA in the XML element:

> `<Message ...>This is some text</Message>`

> Appears in the documents in the following form: CDATA

NMTOKEN

> The data type for XML identifiers.

> Note: The identifier must start with a letter, an underscore "_" or a colon. The next character must be a letter, a number, or one of the following ".", "-", "_", ":". The identifier must not have any spaces or special characters.

> Appears in the documents in the following form: NMTOKEN.

XML

> Stands for eXtensible Markup Language.

> XML defines a set of rules for encoding documents that both a human-readable and machine-readable.

> XML is the language used for all code examples in the MTConnect Standard.

> Refer to http://www.w3.org/XML for more information about XML.

Agent

> Refers to an MTConnect Agent.

> Software that collects data published from one or more piece(s) of equipment, organizes that data in a structured manner, and responds to requests for data from client

40 software systems by providing a structured response in the form of a *Response Doc-*
41 *ument* that is constructed using the *semantic data models* defined in the Standard.

42 Appears in the documents in the following form: *Agent*.

43 ***Asset***

44 item, thing or entity that has potential or actual value to an organization *Ref:ISO*
45 *55000:2014(en)*

46 Note 1 to entry: Value can be tangible or intangible, financial or non-financial,
47 and includes consideration of risks and liabilities. It can be positive or negative
48 at different stages of the asset life.

49 Note 2 to entry: Physical assets usually refer to equipment, inventory and prop-
50 erties owned by the organization. Physical assets are the opposite of intangible
51 assets, which are non-physical assets such as leases, brands, digital assets, use
52 rights, licences, intellectual property rights, reputation or agreements.

53 Note 3 to entry: A grouping of assets referred to as an asset system could also
54 be considered as an asset.

55

56 ***Child Element***

57 A portion of a data modeling structure that illustrates the relationship between an
58 element and the higher-level *Parent Element* within which it is contained.

59 Appears in the documents in the following form: *Child Element*.

60 ***Component***

61 General meaning:

62 A *Structural Element* that represents a physical or logical part or subpart of a piece
63 of equipment.

64 Appears in the documents in the following form: *Component*.

65 Used in *Information Models*:

66 A data modeling element used to organize the data being retrieved from a piece of
67 equipment.

68 • When used as an XML container to organize *Lower Level* `Component` ele-
69 ments.
70 Appears in the documents in the following form: `Components`.

71 • When used as an abstract XML element. `Component` is replaced in a data
72 model by a type of *Component* element. `Component` is also an XML con-
73 tainer used to organize *Lower Level* `Component` elements, *Data Entities*, or
74 both.

75 Appears in the documents in the following form: `Component`.

Current Request
76

77 A *Current Request* is a *Request* to an *Agent* to produce an *MTConnectStreams Re-*
78 *sponse Document* containing the *Observations Information Model* for a snapshot of
79 the latest *observations* at the moment of the *Request* or at a given *sequence number*.

Data Entity
80

81 A primary data modeling element that represents all elements that either describe
82 data items that may be reported by an *Agent* or the data items that contain the actual
83 data published by an *Agent*.

84 Appears in the documents in the following form: *Data Entity*.

Devices Information Model
85

86 A set of rules and terms that describes the physical and logical configuration for a
87 piece of equipment and the data that may be reported by that equipment.

88 Appears in the documents in the following form: *Devices Information Model*.

Equipment Metadata
89

90 See *Metadata*

Information Model
91

92 The rules, relationships, and terminology that are used to define how information is
93 structured.

94 For example, an information model is used to define the structure for each *MTCon-*
95 *nect Response Document*; the definition of each piece of information within those
96 documents and the relationship between pieces of information.

97 Appears in the documents in the following form: *Information Model*.

Lower Level
98

99 A nested element that is below a higher level element.

100 *Metadata*

101 Data that provides information about other data.

102 For example, *Equipment Metadata* defines both the *Structural Elements* that rep-
103 resent the physical and logical parts and sub-parts of each piece of equipment, the
104 relationships between those parts and sub-parts, and the definitions of the *Data En-*
105 *tities* associated with that piece of equipment.

106 Appears in the documents in the following form: *Metadata* or *Equipment Metadata*.

107 *MTConnect Agent*

108 See definition for *Agent*.

109 *MTConnect Asset*

110 An *MTConnect Asset* is an *Asset* used by the manufacturing process to perform
111 tasks.

112   Note 1 to entry: An *MTConnect Asset* relies upon an *MTConnect Device* to
113 provide *observations* and information about itself and the *MTConnect Device*
114 revises the information to reflect changes to the *MTConnect Asset* during their
115 interaction. Examples of *MTConnect Assets* are Cutting Tools, Part Information,
116 Manufacturing Processes, Fixtures, and Files.

117 Note 2 to entry: A singular `assetId` uniquely identifies an *MTConnect Asset*
118 throughout its lifecycle and is used to track and relate the *MTConnect Asset* to
119 other *MTConnect Devices* and entities.

120 Note 3 to entry: *MTConnect Assets* are temporally associated with a device and
121 can be removed from the device without damage or alteration to its primary
122 functions.

123

124 *MTConnect Device*

125 An *MTConnect Device* is a piece of equipment or a manufacturing system that pro-
126 duces *observations* about itself and/or publishes data using the *MTConnect Infor-*
127 *mation Model*.

128 *MTConnect Information Model*

129 See *Information Model*

130 *MTConnectDevices Response Document*

131 A *Response Document* published by an *MTConnect Agent* in response to a *Probe*
132 *Request*.

133 **MTConnectStreams Response Document**

134 A *Response Document* published by an *MTConnect Agent* in response to a *Current*
135 *Request* or a *Sample Request*.

136 *observation*

137 The observed value of a property at a point in time.

138 **Observations Information Model**

139 An *Information Model* that describes the *Streaming Data* reported by a piece of
140 equipment.

141 **Parent Element**

142 An XML element used to organize *Lower Level* child elements that share a common
143 relationship to the *Parent Element*.

144 Appears in the documents in the following form: *Parent Element*.

145 **Probe Request**

146 A *Probe Request* is a *Request* to an *Agent* to produce an *MTConnectDevices Re-*
147 *sponse Document* containing the *Devices Information Model*.

148 **Request**

149 A communications method where a client software application transmits a message
150 to an *Agent*. That message instructs the *Agent* to respond with specific information.

151 Appears in the documents in the following form: *Request*.

152 **Response Document**

153 An electronic document published by an *MTConnect Agent* in response to a *Probe*
154 *Request*, *Current Request*, *Sample Request* or *Asset Request*.

155 **Sample Request**

156 A *Sample Request* is a *Request* to an *Agent* to produce an *MTConnectStreams Re-*
157 *sponse Document* containing the *Observations Information Model* for a set of time-
158 stamped *observations* made by *Components*.

159 **semantic data model**

160 A methodology for defining the structure and meaning for data in a specific logical
161 way.

162 It provides the rules for encoding electronic information such that it can be inter-
163 preted by a software system.

164 Appears in the documents in the following form: *semantic data model*.

165 *sequence number*

166 The primary key identifier used to manage and locate a specific piece of *Streaming*
167 *Data* in an *Agent*.

168 *sequence number* is a monotonically increasing number within an instance of an
169 *Agent*.

170 Appears in the documents in the following form: *sequence number*.

171 *Streaming Data*

172 The values published by a piece of equipment for the *Data Entities* defined by the
173 *Equipment Metadata*.

174 Appears in the documents in the following form: *Streaming Data*.

175 *Structural Element*

176 General meaning:

177 An XML element that organizes information that represents the physical and logical
178 parts and sub-parts of a piece of equipment.

179 Appears in the documents in the following form: *Structural Element*.

180 Used to indicate hierarchy of Components:

181 When used to describe a primary physical or logical construct within a piece of
182 equipment.

183 Appears in the documents in the following form: *Top Level Structural Element*.

184 When used to indicate a *Child Element* which provides additional detail describing
185 the physical or logical structure of a *Top Level Structural Element*.

186 Appears in the documents in the following form: *Lower Level Structural Element*.

187 *Top Level*

188 *Structural Elements* that represent the most significant physical or logical functions
189 of a piece of equipment.

190 *Valid Data Value*

191 One or more acceptable values or constrained values that can be reported for a *Data*
192 *Entity*.

193 Appears in the documents in the following form: *Valid Data Value*(s).

194 *XML Schema*

195 In the MTConnect Standard, an instantiation of a schema defining a specific docu-
196 ment encoded in XML.

2.2 Acronyms

AMT

The Association for Manufacturing Technology

2.3 MTConnect References

[MTConnect Part 1.0] *MTConnect Standard Part 1.0 - Overview and Fundamentals*. Version 1.7.0.

[MTConnect Part 3.0] *MTConnect Standard: Part 3.0 - Streams Information Model*. Version 1.7.0.

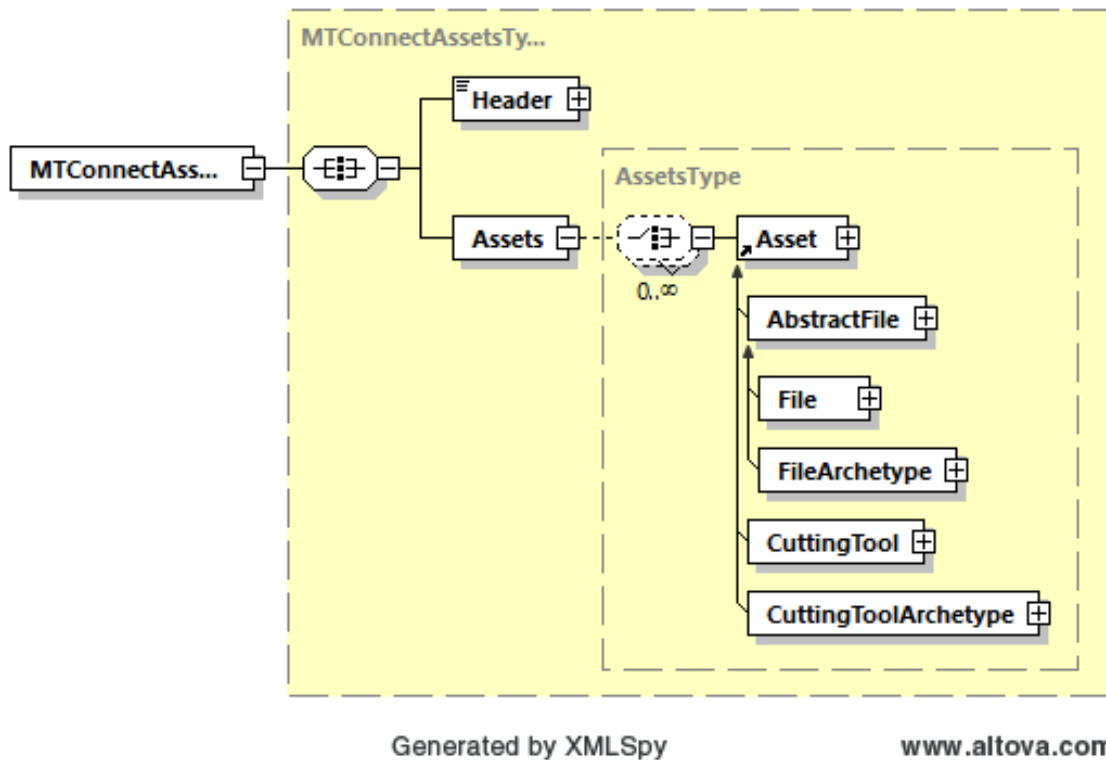[MTConnect Part 4.0] *MTConnect Standard: Part 4.0 - Assets Information Model*. Version 1.7.0.

207 3 MTConnect Assets

208 3.1 Overview

209 The MTConnect Standard supports a simple distributed storage mechanism that allows ap-
210 plications and equipment to share and exchange complex information models in a similar
211 way to a distributed data store. The *Asset Information Model* associates each electronic
212 `MTConnectAssets` document with a unique identifier and allows for some predefined
213 mechanisms to find, create, request, updated, and delete these electronic documents in a
214 way that provides for consistency across multiple pieces of equipment.

215 The protocol provides a limited mechanism of accessing *MTConnect Assets* using the fol-
216 lowing properties: `assetId`, *Asset* type (element name of *Asset* root), and the piece of
217 equipment associated with the *Asset*. These access strategies will provide the following
218 services and answer the following questions: What *Assets* are from a particular piece of
219 equipment? What are the *Assets* of a particular type? What *Assets* is stored for a given
220 `assetId`?

221 Although these mechanisms are provided, an *Agent* should not be considered a data store
222 or a system of reference. The *Agent* is providing an ephemeral storage capability that will
223 temporarily manage the data for applications wishing to communicate and manage data as
224 need-ed by the various processes. An application cannot rely on an *Agent* for long term
225 persistence or durability since the *Agent* is only required to temporarily store the *Asset*
226 data and may require an-other system to provide the source data upon initialization. An
227 *Agent* is always providing the best-known equipment centric view of the data given the
228 limitations of that piece of equipment.

3.2 MTConnectAssets

Figure 1: MTConnectAssets Schema

At the top level of the `MTConnectAssets` document is a standard header, as stated in *MTConnect Standard Part 1.0 - Overview and Fundamentals*, and one or more *MTConnect Assets*. Each *Asset* is required to have an `assetId` that serves as a unique identifier of that *Asset*. `assetId` allows an application to request the *Asset* data from an *Agent*.

In the remaining Part 4.x sub-part documents of *MTConnect Assets*, various types of *Assets* will be introduced such as cutting tools and other *Asset* types.

3.2.1 MTConnectAssets Header

The `MTConnectAssets` header is where the protocol sequence information **MUST** be provided. The *XML Schema* in *Figure 2* represents the structure of the `MTConnectAssets` header showing the attributes defined for `MTConnectAssets`.

Refer to *MTConnect Standard Part 1.0 - Overview and Fundamentals* for more information on headers.
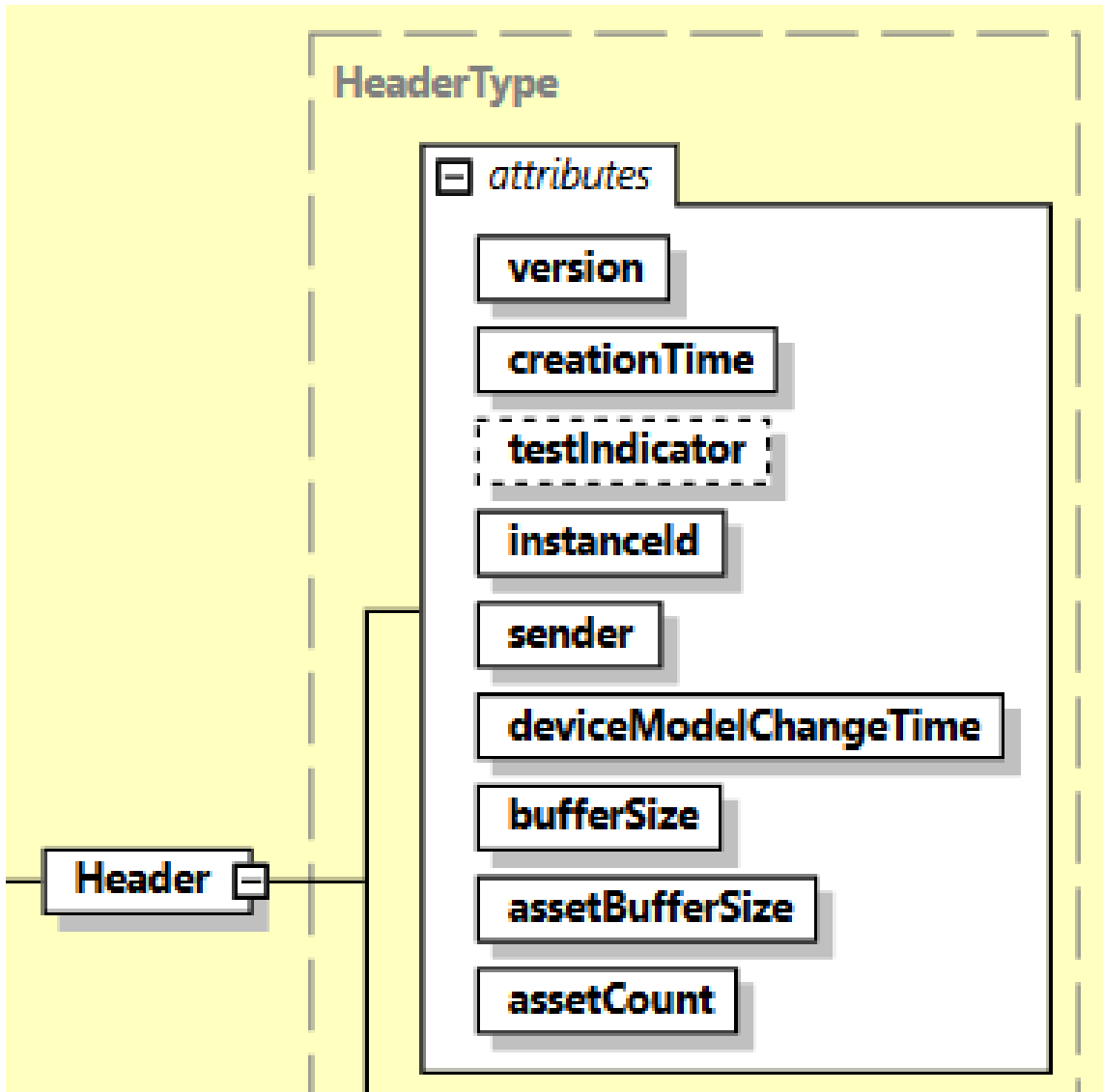
Figure 2: MTConnectAssets Header

242 **3.2.1.1 Header Attributes**

243 *Table 1* defines the attributes used to provide information for an `MTConnectAssets`
244 header.

Table 1: MTConnectAssets Header

Attribute	Description	Occurrence
version	The protocol version number. This is the *major* and *minor* version number of the MTConnect Standard being used. For example, if the version number of the Standard used is `10.21.33`, the `version` will be `10.21`. `version` is a required attribute.	1
creationTime	The time the response was created. `creationTime` is a required attribute.	1
testIndicator	Optional flag that indicates the system is operating in test mode. This data is only for testing and indicates that the data is simulated. `testIndicator` is an optional attribute.	0..1
instanceId	A number indicating which invocation of the *Agent*. This is used to differentiate between separate instances of the *Agent*. This value **MUST** have a maximum value of $2^{64} - 1$ and **MUST** be stored in an unsigned 64-bit integer. `instanceId` is a required attribute.	1
sender	The *Agent* identification information. `sender` is a required attribute.	1

Continuation of Table 1		
Attribute	Description	Occurrence
`assetBufferSize`	The maximum number of *MTConnect Assets* that will be retained by the *Agent*. The `assetBufferSize` **MUST** be an unsigned positive integer value with a maximum value of $2^{32} - 1$. `assetBufferSize` is a required attribute.	1
`assetCount`	The total number of *MTConnect Assets* in an *Agent*. This **MUST** be an unsigned positive integer value with a maximum value of $2^{32} - 1$. This value **MUST NOT** be greater than `assetBufferSize`. `assetCount` is a required attribute.	1
`deviceModelChangeTime`	A timestamp in 8601 format of the last update of the `Device` information for any device.	1

Example 1: MTConnectAssets Header Example

```
245  1  <Header creationTime="2010-03-13T07:59:11+00:00"
246  2       sender="localhost" instanceId="1268463594"
247  3       assetBufferSize="1024" version="1.1"
248  4       assetCount="12" />
```

249 ### 3.2.2 Assets

250 `Assets` is an XML container used to group information about various *MTConnect Asset*
251 types. `Assets` contains one or more `Asset` XML elements.

Table 2: MTConnect Assets Element

Element	Description	Occurrence
`Assets`	An XML container that consists of one or more types of `Asset` XML elements.	0..1

252 ### 3.2.3 Asset

253 An `Asset` XML element is a container type XML element used to organize information
254 describing an entity that is not a piece of equipment. `Asset` is an abstract type XML
255 element and will never appear directly in the MTConnect XML document. As an abstract
256 type XML element, `Asset` will be replaced in the XML document by specific *MTConnect*
257 *Asset* type.

Table 3: MTConnect Asset Element

Element	Description	Occurrence
Asset	An abstract XML element. Replaced in the XML document by types of `Asset` elements representing entities that are not pieces of equipment. There can be multiple types of `Asset` XML elements in the document.	1..*

258 There are various types of entities or *Asset* types. Each type of *Asset* is described in sub-
259 parts of *MTConnect Standard: Part 4.0 - Assets Information Model*. These sub-parts are
260 designated by a Part 4.x document number.

261 For all *MTConnect Asset* types there are some common attributes and elements that apply
262 to all of them. The following defines these common attributes and elements.

263 #### 3.2.3.1 Common Asset Attributes

264 The *XML Schema* in *Figure 3* represents the structure of `Asset` showing the attributes
265 defined for `Asset`.
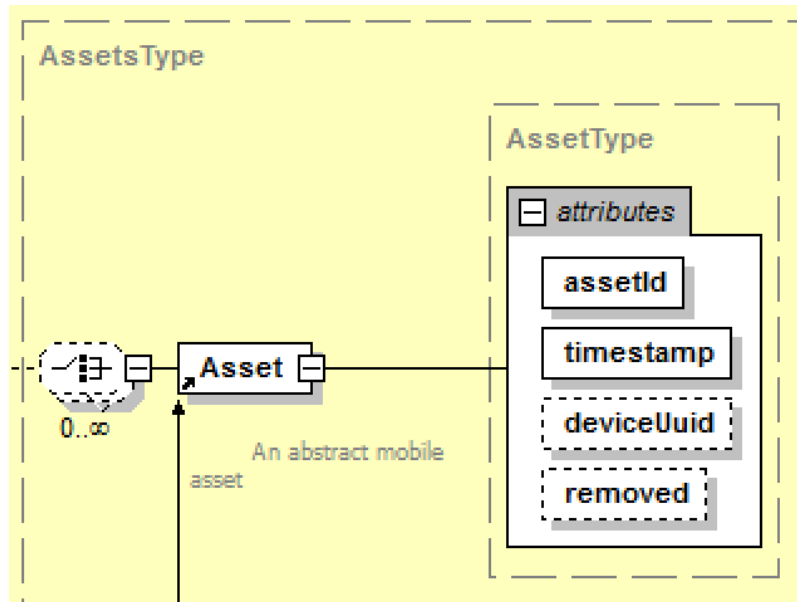
Figure 3: Asset Schema

266 *Table 4* defines the attributes that are used to provide information for the `Asset` element.

Table 4: Attributes for Asset

Attribute	Description	Occurrence
assetId	The unique identifier for the *MTConnect Asset*. The identifier **MUST** be unique with respect to all other *Assets* in an MTConnect installation. The identifier **SHOULD** be globally unique with respect to all other *Assets*. assetId is a required attribute.	1
timestamp	The time this *MTConnect Asset* was last modified. Always given in UTC. The timestamp **MUST** be provided in UTC (Universal Time Coordinate, also known as GMT). This is the time the *Asset* data was last modified. timestamp is a required attribute.	1

Continuation of Table 4		
Attribute	Description	Occurrence
deviceUuid	The piece of equipments UUID that supplied this data. This is an optional element references to the UUID attribute given in the Device element. This can be any series of numbers and letters as defined by the XML type NMTOKEN.	0..1
removed	This is an optional attribute that is an indicator that the *MTConnect Asset* has been removed from the piece of equipment. If the *Asset* is marked as removed, it will not be visible to the client application unless the=true parameter is provided in the URL. If this attribute is not present it MUST be assumed to be false. The value is an xsi:boolean type and **MUST** be true or false.	0..1

267 All *MTConnect Assets* **MUST** have a unique value for assetId and it **SHOULD** be
268 globally unique, such as a RFC 4122 UUID.

269 The following attributes **MUST** be provided and are common to all *MTConnect Asset*
270 types: the assetId attribute providing the unique identifier for the *Asset*, and the times-
271 tamp providing the time the *Asset* was inserted or updated. A removed flag that if true
272 indicates the *Asset* has been removed (deleted) from the equipment is optional, however
273 the *Asset* will still be available if requested directly or a request is made that includes
274 removed *Assets*.

275 An MTConnectAssets document contains information pertaining to something that is
276 not a direct component of the piece of equipment and can be relocated to another piece
277 of equipment or location during its lifecycle. The Asset will contain data that will be
278 changed as a unit, meaning that at any given point in time the latest version of the complete
279 state for this *Asset* will be provided.

280 Each piece of equipment or location may have a different view of this *Asset* and it is
281 the responsibility of an application to collect and determine the aggregate information
282 and keep a historical record if required. An *Agent* will allow any application or other
283 equipment to request this information. The piece of equipment **MUST** supply the latest
284 and most accurate information regarding a given *Asset*.

285 **3.2.3.2   Common Asset Elements**

286 The element `Description` is the only element common to all *Asset* types.

287 The *XML Schema* in *Figure 4* represents the structure of `Description`.
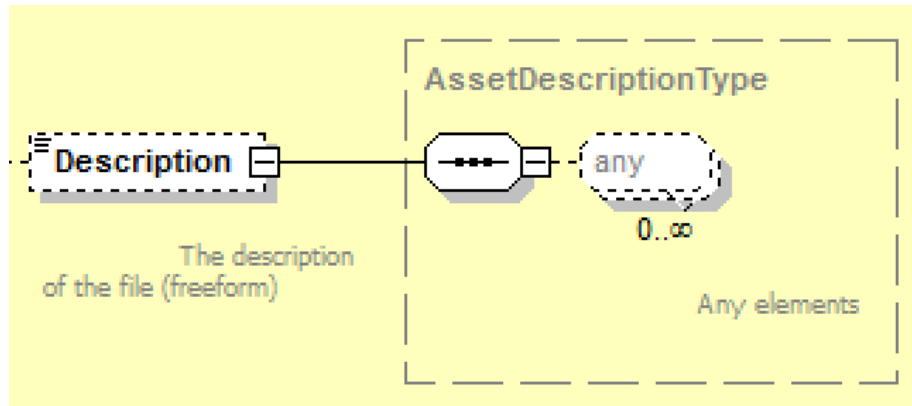
Figure 4: Description Schema

288 *Table 5* defines the elements that are used to provide information for `Asset`.

Table 5: Elements for Asset

| Elements	Description	Occurrence
Description	An optional element that can contain any descriptive content. This can contain configuration information and manufacturer specific details. This element is defined to contain mixed content and XML elements can be added to extend the descriptive semantics of MTConnect Standard.	0..1

289 4 MTConnect Assets Architecture

290 4.1 Agent Asset Storage

291 The *Agent* stores *MTConnect Assets* in a similar fashion as the *Agent* data storage de-
292 scribed in *MTConnect Standard Part 1.0 - Overview and Fundamentals*. The storage of
293 information is contained in the *asset buffer*. The *Agent* provides a limited number of *As-*
294 *sets* that can be stored at one time and uses the same method of pushing out the oldest
295 *Asset* when the *asset buffer* is full. The *asset buffer* size for the *Asset* storage is maintained
296 separately from the `Sample`, `Event`, and `Condition` storage.
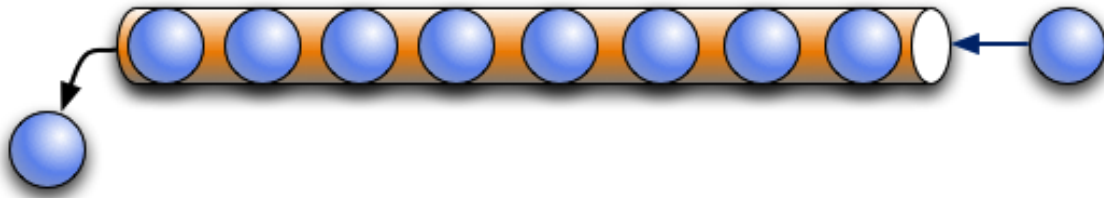
Figure 5: MTConnect Assets storage as First in First Out

297 *MTConnect Assets* also behave like a key/value in memory database. In the case of the
298 *Asset*, the key is the `assetId` and the value is the XML document describing the *Asset*.
299 The key can be any string of letters, punctuation or digits and represent the domain specific
300 coding scheme for their assets. Each *Asset* type will have a recommended way to construct
301 a unique `assetId`, for example, a cutting tool **SHOULD** be identified by the tool ID and
302 serial number as a composed synthetic identifier.
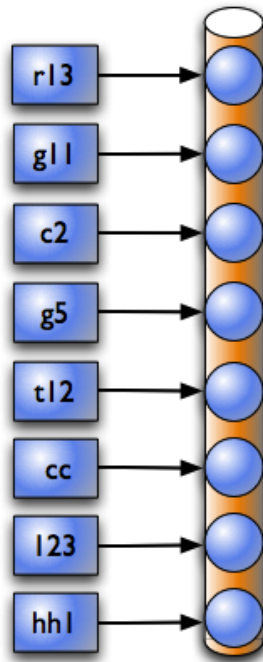
Figure 6: MTConnect Assets storage as Key/Value pairs

303 As in *Figure 6* , each of the *Assets* is referred to by their key. The key is independent of
304 the order in the *asset buffer* storage.

4.2   Asset Protocol

306 MTConnect Standard provides methods to retrieve an *MTConnect Asset* or a set of *Assets*
307 given various criteria. These criteria are as follows: The `assetId`, the *Asset* `type` as de-
308 fined by the name of the *Asset*'s topmost element, and the originating piece of equipment.

309 The URL format is similar to the `probe` and `sample` structure. Reference each `as-`
310 `setId` directly to request an *MTConnect Asset* by `assetId`.

4.2.1   Asset by assetId

Example 2: Asset by assetId Example

312 1 `url: http://example.com/asset/e39d23ba-ef2d-`
313 2   `11e6-b12c15028cfe91a82ef`

314 *Example 2* returns the `MTConnectAssets` document for *Asset* `e39d23ba-ef2d-`
315 `11e6-b12c-28cfe91a82ef`

316 Request multiple *Assets* by each `assetId`:

Example 3: Assets by assetId Example

```
317  1  url: http://example.com/asset/e39d23ba-ef2d-11e6-b12c155;
318  2     8cfe91a82ef;e46d5256-ef2d-11e6-96aa-28cfe91a82ef
```

319 *Example 3* returns the `MTConnectAssets` document for *Assets* `e39d23ba-ef2d-`
320 `11e6-b12c-28cfe91a82ef` and `e46d5256-ef2d-11e6-96aa-28cfe91a82ef`.

321 Request for all the *Assets* in the *Agent*:

Example 4: Get all Assets Example

```
322  1  url: http://example.com/assets
```

323 *Example 4* returns all available *MTConnect Assets* in the *Agent*. The *Agent* **MAY** return
324 a limited set if there are too many *Asset* records. The *Assets* **MUST** be added to the
325 beginning with the most recently modified *Asset*.

4.2.2   Asset for a Given Type

Example 5: Asset for a Given Type Example

```
327  1  url: http://example.com/assets?type="CuttingTool"
```

328 *Example 5* returns all available `CuttingTool` *Assets* from the *Agent* of the type `Cut-`
329 `tingTool`. The *Agent* **MAY** return a limited set if there are too many *Asset* records. The
330 *Assets* **MUST** be added to the beginning with the most recently modified assets.

331 Request for all *Assets* of a given type in the *Agent* up to a maximum count:

Example 6: Asset for a Given Type with Maximum count Example

```
332  1  url: http://example.com/assets?type="CuttingTool"
```

333 *Example 6* returns all available `CuttingTool` *Assets* from the *Agent*. The *Agent* **MUST**
334 return up to 1000 *Assets* beginning with the most recently modified *Assets* if they exist.

4.2.3   Assets Including Removed Assets

Example 7: Assets Including Removed Assets Example

```
336  1  url: http://example.com/assets?type=CuttingTool&removed=true
```

337 *Example 7* returns all available `CuttingTool` *Assets* from the *Agent*. With the removed
338 flag, *Assets* that have been removed but are included in the result set.

4.2.4   Assets for a Piece of Equipment

340 If no `assetId` is provided with a general *Assets* request, it would be as shown in *Exam-*
341 *ple 8*:

Example 8: Assets For a Piece of Equipment Example

342   1   url: http://example.com/Mill123/assets

343 All *MTConnect Assets* will be provided for that piece of equipment (`Device`) up to the
344 *Agent*'s maximum count or as specified with the count parameter. These *Assets* will be
345 returned starting from the newest to oldest list.

346 Any of the previous constraints can also be applied to the request, for example, to get all
347 the `CuttingTool` instances for a given piece of equipment:

Example 9: Assets For a Piece of Equipment For a Given Type Example

348   1   url: http://example.com/Mill123/asset/
349   2   ?type=CuttingTool&count=100

350 The request in *Example 9* will get the newest 100 Cutting Tool Instance *Assets* from the
351 *Agent* for `Mill123`. Similarly:

Example 10: Assets For a Piece of Equipment For a Given Type Example 2

352   1   url: http://example.com/Mill123/asset/
353   2   ?type=CuttingToolArchetype

354 *Example 10* will provide all Cutting Tool Archetype *Assets* with the `deviceUuid` of
355 `Mill123`.

356 5 Extensions to Part 2.0 - Devices Information Model

357 This document will add the following data item types to support change notification when
358 an *MTConnect Asset* is added or updated. The data item **MUST** be placed in the `DataItems`
359 container associated with `Device`. The `Device` **MUST** be the piece of equipment that
360 is supplying the asset data.

361 5.1 Data Item Types added for EVENT Category

Table 6: DataItem Type for EVENT category

DataItem Type SubType	Description
ASSET_CHANGED	The event generated when an asset is added or changed. `AssetChanged` **MUST** be discrete and the value of the `DataItem`'s `discrete` attribute **MUST** be `true`.
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.

362 5.1.1 ASSET_CHANGED Data Item Type

363 When an *MTConnect Asset* is added or modified, an `AssetChanged` event **MUST** be
364 published to inform an application that new asset data is available. The application can
365 request the new asset data from the piece of equipment at that time. Every time the asset
366 data is modified an `AssetChanged` event will be published. Since the asset data is a
367 complete electronic document, the system will publish a single `AssetChanged` event
368 for the entire set of changes.

369 The asset data **MUST** remain constant until the `AssetChanged` event is published.
370 Once it is published the data **MUST** change to reflect the new content at that instant.
371 The timestamp of the asset will reflect the time the last change was made to the asset data.

372 5.1.2 ASSET_REMOVED Data Item Type

373 When an *MTConnect Asset* has been removed from an *Agent*, or marked as removed, an
374 `AssetRemoved` event **MUST** be generated in a similar way to the `AssetChanged`
375 event. The CDATA of the `AssetRemoved` event **MUST** contain the `assetId` that was
376 just removed.

377 Every time an *MTConnect Asset* is modified or added it will be moved to the beginning
378 of the *asset buffer* and become the newest *Asset*. As the *asset buffer* fills up, the oldest
379 *Asset* will be pushed out and its information will be removed. The MTConnect Standard
380 does not specify the maximum size of the *asset buffer*, and if the implementation desires,
381 permanent storage **MAY** be used to store the *Assets*. A value of 4,294,967,296 or 2^{32} can
382 be given to indicate unlimited storage.

383 There is no requirement for persistent *Asset* storage. If the *Agent* fails, all existing *MT-*
384 *Connect Assets* **MAY** be lost. It is the responsibility of the implementation to restore the
385 lost *Asset* data and it is the responsibility of the application to persist the *Asset* data. The
386 *Agent* **MAY** make no guarantees about availability of *Asset* data after the *Agent* stops.

6 Extensions to Part 3.0 - Streams Information Model

387

388 The associated modifications **MUST** be added to *MTConnect Standard: Part 3.0 - Streams*
389 *Information Model* to add the following event to the `Events` in the streams.

6.1 AssetChanged Extension to Events

390

391 The `AssetChanged` element extends the base `Event` type XML data element defined in
392 *MTConnect Standard: Part 3.0 - Streams Information Model* and adds the `assetType`
393 attribute to the base `Event`. This new `Event` will signal whenever a new *MTConnect*
394 *Asset* is added or the existing definition of an *Asset* is updated. The `assetId` is provided
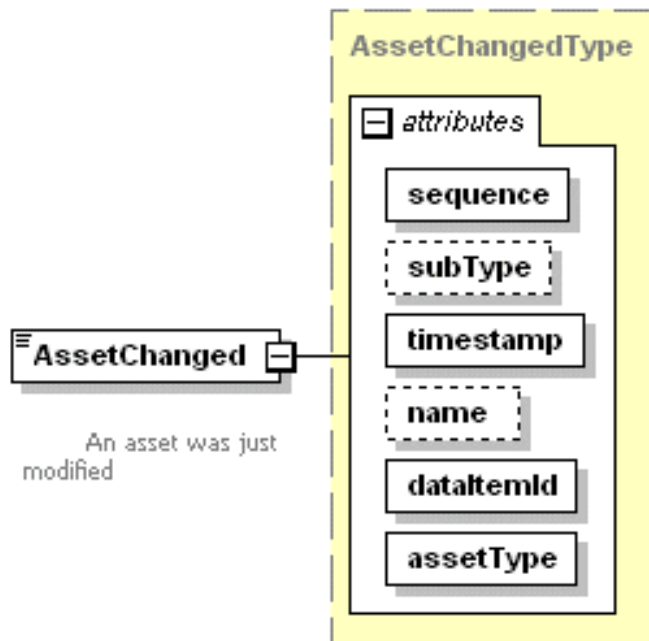395 as the CDATA value and can be used to request the *Asset* data from the *Agent*.

**Figure 7:** AssetChanged Schema

396 `AssetChanged`: An *MTConnect Asset* has been added or modified. The CDATA
397 for the `AssetChanged` element **MUST** be the `assetId` of the *Asset* that has been
398 modified.

399 6.1.1 AssetChanged event Attributes

Table 7: Attributes for AssetChanged

Attribute	Description	Occurrence
assetType	The type of asset changed. assetType is a required attribute. *Valid Data Values*: CuttingTool File	1

400 6.2 AssetRemoved Extension to Events
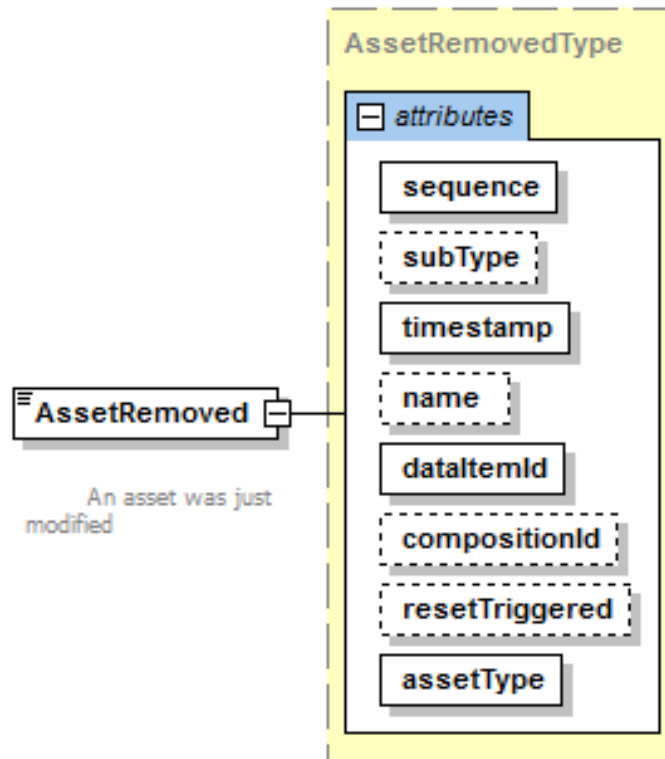
Figure 8: AssetRemoved Schema

401   AssetRemoved: An *MTConnect Asset* has been removed. The CDATA for the As-

402 `setRemoved` element **MUST** be the `assetId` of the *Asset* that has been removed.

6.2.1   **AssetRemoved Attributes**

Table 8: Attributes for AssetRemoved

Attribute	Description	Occurrence
assetType	The type of asset that was removed. assetType is a required attribute. *Valid Data Values*: CuttingTool File	1

404 The *MTConnect Asset* will still be available if requested if the removed=true argument is
405 supplied. The `assetId` is provide as the CDATA value and can be used to request the
406 *Asset* data from the *Agent*.

Appendices

A Bibliography

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.

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.

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.

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.

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.

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.

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

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.

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.

H.L. Horton, F.D. Jones, and E. Oberg. *Machinery's Handbook.* Industrial Press, Inc.

438   New York, 1984.

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

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

444   *ASME/ANSI B5.54: Methods for Performance Evaluation of Computer Numerically Con-*
445   *trolled Machining Centers. 2005.*

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

448   International Organization for Standardization. *ISO 13399: Cutting tool data representa-*
449   *tion and exchange*. Geneva, Switzerland, 2000.