



# MTConnect<sup>®</sup> Standard

## Part 3.0 – Streams Information Model

Version 1.4.0

Prepared for: MTConnect Institute

Prepared on: March 31, 2018

# MTConnect<sup>®</sup> Specification and Materials

AMT - The Association For Manufacturing Technology (“AMT”) owns the copyright in this MTConnect<sup>®</sup> 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](http://www.MTConnect.org) or by contacting [info@MTConnect.org](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 non-infringement, 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

<b>1</b>	<b>Purpose of This Document</b> .....	<b>1</b>
<b>2</b>	<b>Terminology</b> .....	<b>2</b>
<b>3</b>	<b><i>Streams Information Model</i></b> .....	<b>3</b>
<b>4</b>	<b><i>Structural Elements for MTConnectStreams</i></b> .....	<b>5</b>
4.1	Streams .....	8
4.2	DeviceStream.....	9
4.2.1	<i>XML Schema for DeviceStream</i> .....	9
4.2.2	<i>Attributes for DeviceStream</i> .....	10
4.2.3	<i>Elements for DeviceStream</i> .....	10
4.3	ComponentStream .....	11
4.3.1	<i>XML Schema for ComponentStream</i> .....	11
4.3.2	<i>Attributes for ComponentStream</i> .....	12
4.3.3	<i>Elements for ComponentStream</i> .....	14
<b>5</b>	<b><i>Data Entities</i></b> .....	<b>16</b>
5.1	<i>Element Names for Data Entities</i> .....	18
5.1.1	<i>Element Names when MTConnectDevices category is SAMPLE or EVENT</i> .....	18
5.1.2	<i>Changes to Element Names when representation attribute is used</i> .....	19
5.1.3	<i>Element Names when MTConnectDevices category is CONDITION</i> .....	20
5.2	Samples Container.....	20
5.3	Sample Data Entities.....	21
5.3.1	<i>XML Schema Structure for Sample</i> .....	22
5.3.2	<i>Attributes for Sample</i> .....	23
5.3.2.1	<i>duration Attribute for Sample</i> .....	25
5.3.2.2	<i>resetTriggered Attribute for Sample</i> .....	25
5.3.3	<i>Response for SAMPLE category DataItem Elements with a representation attribute of TIME_SERIES</i> .....	27
5.3.3.1	<i>XML Schema Structure for Sample when reporting Time Series data</i> .....	28
5.3.3.2	<i>Attributes for a Sample when reporting Time Series data</i> .....	29
5.3.4	<i>Valid Data Values for Sample</i> .....	29
5.3.5	<i>Unavailability of Valid Data Values for Sample</i> .....	31

5.4	Events Container .....	31
5.5	Event <i>Data Entities</i> .....	32
5.5.1	XML Schema Structure for <i>Event</i> .....	33
5.5.2	Attributes for <i>Event</i> .....	33
5.5.3	Response for <i>EVENT</i> category <i>Data Items</i> with a representation attribute of <i>DISCRETE</i> .....	34
5.5.4	Response for <i>EVENT</i> category <i>Data Items</i> with a type attribute of <i>MESSAGE</i> .....	35
5.5.5	Valid Data Values for <i>Event</i> .....	35
5.5.6	Unavailability of Valid Data Values for <i>Event</i> .....	36
5.6	Condition Container .....	36
5.7	Condition <i>Data Entities</i> .....	37
5.7.1	Element Names for <i>Condition</i> .....	38
5.7.2	XML Schema Structure for <i>Condition</i> .....	39
5.7.3	Attributes for <i>Condition</i> .....	39
5.7.3.1	qualifier Attribute for <i>Condition</i> .....	42
5.7.4	Valid Data Values for <i>Condition</i> .....	42
5.8	Unavailability of <i>Fault State</i> for <i>Condition</i> .....	43
<b>6</b>	<b>Listing of <i>Data Entities</i></b> .....	<b>44</b>
6.1	Sample <i>Element Names</i> .....	44
6.2	Event <i>Element Names</i> .....	52
6.3	Types of <i>Condition Elements</i> .....	76
	<b>Appendices</b> .....	<b>78</b>
<b>A.</b>	<b>Bibliography</b> .....	<b>78</b>

# Table of Figures

Figure 1: Streams Data Structure .....	6
Figure 2: Streams Schema Diagram.....	8
Figure 3: DeviceStream Schema Diagram.....	9
Figure 4: ComponentStream Schema Diagram .....	11
Figure 5: ComponentStream XML Tree Diagram.....	16
Figure 6: Sample Schema Diagram .....	22
Figure 7: AbsTimeSeries Schema Diagram .....	28
Figure 8: Event Schema Diagram .....	33
Figure 9: Condition Schema Diagram.....	39

# 1 Purpose of This Document

2 This document, *Part 3.0 - Streams Information Model* of the MTCConnect<sup>®</sup> Standard, establishes  
3 the rules and terminology that describes the information returned by an *MTCConnect Agent* from a  
4 piece of equipment. The *Streams Information Model* also defines, in *Section 3*, the structure for  
5 the XML documents that are returned from an *MTCConnect Agent* in response to a `Sample` or  
6 `Current` request.

7 *Part 3.0 - Streams Information Model* is not a stand-alone document. This document is used in  
8 conjunction with *Part 1.0 – Overview and Functionality* which defines the fundamentals of the  
9 operation of the MTCConnect Standard and *Part 2.0 – Devices Information Model* that defines the  
10 semantic model representing the information that may be returned from a piece of equipment.

11 Note: *Part 5 – Interfaces* provides details on extensions to the *Streams Information Model*  
12 required to describe the interactions between pieces of equipment.

13 In the MTCConnect Standard, *equipment* represents any tangible property that is used in the  
14 operation of a manufacturing facility. Examples of *equipment* are machine tools, ovens, sensor  
15 units, workstations, software applications, and bar feeders.

16

17

## 18 **2 Terminology**

19 Refer to *Section 5 of Part 1.0 – Overview and Functionality* for a dictionary of terms, reserved  
20 language, and document conventions used in the MTConnect<sup>®</sup> Standard.

### 21 **3 Streams Information Model**

22 The *Streams Information Model* provides a representation of the data reported by a piece of  
 23 equipment used for a manufacturing process, or used for any other purpose. Additional  
 24 descriptive information associated with the reported data is defined in the  
 25 `MTConnectDevices` document, which is described in *Part 2.0 – Devices Information Model*.

26 Information defined in the *Streams Information Model* allows a software application to (1)  
 27 determine the value for *Data Entities* returned from a piece of equipment and (2) interpret the  
 28 data associated with those *Data Entities* with the same meaning, value, and context that it had at  
 29 its original source. To do this, the software application issues one of two HTTP requests to an  
 30 *MTConnect Agent* associated with a piece of equipment. They are:

- 31 • `sample`: Returns a designated number of time stamped *Data Entities* from an  
 32 *MTConnect Agent* associated with a piece of equipment; subject to any HTTP filtering  
 33 associated with the request. See *Section 8.3.3 of Part 1.0 – Overview and Functionality*  
 34 of the MTConnect Standard for details on the `sample` HTTP request.
- 35 • `current`: Returns a snapshot of either the most recent values or the values at a given  
 36 sequence number for all *Data Entities* associated with a piece of equipment from an  
 37 *MTConnect Agent*; subject to any HTTP filtering associated with the request. See  
 38 *Section 8.3.2 of Part 1.0 – Overview and Functionality* of the MTConnect Standard for  
 39 details on the `current` HTTP request.

40 An *MTConnect Agent* responds to either the `sample` or `current` HTTP request with an  
 41 `MTConnectStreams` XML document. This document contains information describing *Data*  
 42 *Entities* reported by an *MTConnect Agent* associated with a piece of equipment. A client  
 43 software application may correlate the information provided in the `MTConnectStreams` XML  
 44 document with the physical and logical structure for that piece of equipment defined in the  
 45 `MTConnectDevices` document to form a clear and unambiguous understanding of the  
 46 information provided. (See details on the structure for a piece of equipment described in *Part*  
 47 *2.0 – Devices Information Model*).

48 The `MTConnectStreams` XML document is comprised of two sections: `Header` and  
 49 `Streams`.

50 The `Header` section contains protocol related information as defined in *Section 6.5 of Part 1.0*  
 51 *– Overview and Functionality* of the MTConnect Standard.

52 The `Streams` section of the `MTConnectStreams` document contains a `DeviceStream`  
 53 XML container for each piece of equipment represented in the document. Each  
 54 `DeviceStream` container is comprised of two primary types of XML elements – *Structural*  
 55 *Elements* and *Data Entities*. The contents of the `DeviceStream` container are described in  
 56 detail in this document, *Part 3.0* of the MTConnect Standard.

57



58 *Structural Elements* are defined for both the `MTConnectDevices` and the  
59 `MTConnectStreams` XML documents. These *Structural Elements* are used to provide a  
60 logical organization of the information provided in each document. While used for a similar  
61 purpose, the *Structural Elements* in the `MTConnectStreams` document are specifically  
62 designed to be distinctly different from those in the `MTConnectDevices` document:

63 • `MTConnectDevices` document: *Structural Elements* organize information that  
64 represents the physical and logical parts and sub-parts of a piece of equipment. (See *Part*  
65 *2.0 –Devices Information Model, Section 4* of the `MTConnect` Standard for more details  
66 on *Structural Elements* used in the `MTConnectDevices` document).

67 • `MTConnectStreams` document: *Structural Elements* provide the structure to organize  
68 the data returned from a piece of equipment and establishes the proper context for that  
69 data. The *Structural Elements* specifically defined for use in the `MTConnectStreams`  
70 document are `DeviceStream` (described in *Section 4.2* of this document) and  
71 `ComponentStream` (described in *Section 4.3* of this document).

72 `DeviceStream` and `ComponentStream` elements have a direct correlation to each  
73 of the *Structural Elements* defined in the `MTConnectDevices` document.

74 *Data Entities* that describe data reported by a piece of equipment are also defined for both the  
75 `MTConnectDevices` and the `MTConnectStreams` XML documents. The *Data Entities*  
76 provided in both documents directly relate to each other. However, *Data Entities* are used for  
77 different purposes in each document:

78 • `MTConnectDevices` document: *Data Entity* elements define the data that may be  
79 returned from a piece of equipment. *Part 2.0 – Devices Information Model, Sections 7*  
80 *and 8* lists the possible *Data Entity* XML elements that can be returned in a  
81 `MTConnectDevices` document.

82 • `MTConnectStreams` document: *Data Entity* elements provide the data reported by a  
83 piece of equipment. This data is organized in separate `ComponentStream` XML  
84 containers for each of the *Structural Elements* defined in the `MTConnectDevices`  
85 document associated with the data that is reported by a piece of equipment.

86 Within each `ComponentStream` XML container in the `MTConnectStreams` document,  
87 *Data Entities* are organized into three types of XML container elements - `Samples`, `Events`,  
88 and `Condition`. (Refer to *Sections 5 and 6* of this document for more information on these  
89 elements.)

## 90 4 *Structural Elements* for MTConnectStreams

91 *Structural Elements* are XML elements that form the logical structure for the  
92 MTConnectStreams XML document. These elements are used to organize the information  
93 and data that is reported by an *MTConnect Agent* for a piece of equipment. Refer to *Figure 1*  
94 below for an overview of the *Structural Elements* used in an MTConnectStreams document.

95 The first, or highest level, *Structural Element* in an MTConnectStreams XML document is  
96 Streams. Streams is a container type XML element used to group the data reported from  
97 one or more pieces of equipment into a single XML document. Streams **MUST** always appear  
98 in the MTConnectStreams document.

99 DeviceStream is the next *Structural Element* in the MTConnectStreams document.  
100 DeviceStream is also a XML container type element. A separate DeviceStream  
101 container is used to organize the information and data reported by each piece of equipment  
102 represented in the MTConnectStreams document. There **MUST** be at least one  
103 DeviceStream element in the Streams container.

104 A DeviceStream element provides the data reported by a piece of equipment. Each  
105 DeviceStream element **MUST** contain the attributes name and uuid to correlate the  
106 DeviceStream with a specific Device defined in the MTConnectDevices document.  
107 Once the DeviceStream element is associated with a specific piece of equipment based on  
108 this identity, all data reported by that piece of equipment is directly associated with that unique  
109 identity and that association does not need to be repeated for every piece of data reported. A  
110 client software application may then directly relate the information provided in the  
111 MTConnectDevices document with the data provided in the MTConnectStreams  
112 document based on this identity.

113 ComponentStream is the next level XML element in the MTConnectStreams document.  
114 ComponentStream is also a container type XML element. There **MUST** be a separate  
115 ComponentStream XML element for each of the *Structural Elements* (Device elements,  
116 *Top Level* Component elements, or *Lower Level* Component elements) defined for that piece  
117 of equipment in the associated MTConnectDevices XML document. A  
118 ComponentStream representing a *Structural Element* will only appear if there is data reported  
119 for that *Structural Element*. (Note: See *Part 2.0 – Devices Information Model* of the  
120 MTConnect Standard for a description of the *Structural Elements* for a piece of equipment).

121

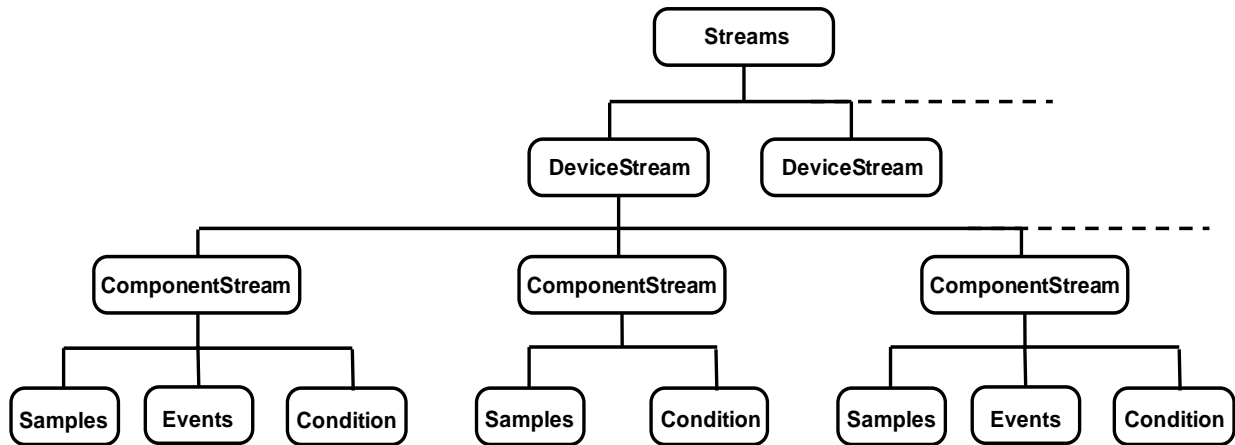
122 There are three (3) *Structural Elements* – Samples, Events, and Condition at the next  
 123 level of the MTConnectStreams document. Each one of these *Structural Elements* is a  
 124 container type XML element. These *Structural Elements* group the data reported for each  
 125 component of a piece of equipment according to the *Data Entity* categories defined in *Part 2.0 –*  
 126 *Devices Information Model, Sections 7 and 8*. Therefore,

- 127 • Samples contains SAMPLE category *Data Entities* defined in the  
 128 MTConnectDevices XML document (See *Part 2.0 – Devices Information Model,*  
 129 *Section 8.1*)
- 130 • Events contains EVENT category *Data Entities* defined in the MTConnectDevices  
 131 XML document (See *Part 2.0 – Devices Information Model, Section 8.2*)
- 132 • Condition contains CONDITION category *Data Entities* defined in the  
 133 MTConnectDevices XML document (See *Part 2.0 – Devices Information Model,*  
 134 *Section 8.3*)

135 There **MUST** be at least one of Samples, Events, or Condition elements in each  
 136 ComponentStream container.

137 The following XML tree structure illustrates the various *Structural Elements* used to organize the data  
 138 reported by a piece of equipment and the relationship between these elements.

139



140

141

142 **Figure 1: Streams Data Structure**

143

144

145 Below is a sample from an `MTConnectStreams` XML document that contains the response  
 146 from an *MTConnect Agent* representing two pieces of equipment, *mill-1* and *mill-2*. The data  
 147 from each piece of equipment is reported in a separate `DeviceStream` container.

```

148 1. <MTConnectStreams ...>
149 2.   <Header ... />
150 3.   <Streams>
151 4.     <DeviceStream name="mill-1" uuid="1">
152 5.       <ComponentStream component="Device" name="mill-1"
153 6.         componentId="d1">
154 7.         <Events>
155 8.           <Availability dataItemId="avail1" name="avail" sequence="5"
156 9.             timestamp="2010-04-06T06:19:35.153141">
157 10.            AVAILABLE</Availability>
158 11.          </Events>
159 12.        </ComponentStream>
160 13.      </DeviceStream>
161 14.      <DeviceStream name="mill-2" uuid="2">
162 15.        <ComponentStream component="Device" name="mill-2"
163 16.          componentId="d2">
164 17.          <Events>
165 18.            <Availability dataItemId="avail2" name="avail" sequence="15"
166 19.              timestamp="2010-04-06T06:19:35.153141">
167 20.              AVAILABLE</Availability>
168 21.            </Events>
169 22.          </ComponentStream>
170 23.        </DeviceStream>
171 24.      </Streams>
172 25. </MTConnectStreams>

```

173

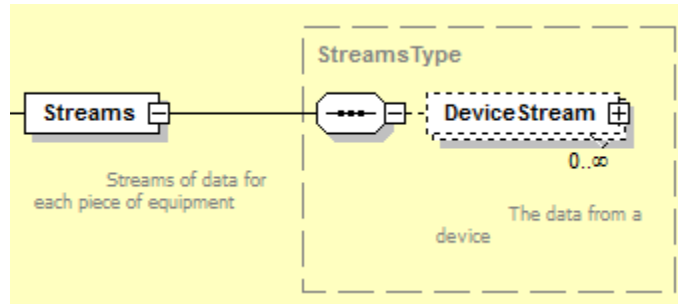
174 In the example above, it should be noted that the *sequence numbers* are unique across the two  
 175 pieces of equipment. Client software applications **MUST NOT** assume that the `Events` and  
 176 `Samples` sequence numbers are strictly in sequence. All sequence numbers **MAY NOT** be  
 177 included. For instance, such a case would occur when HTTP filtering is applied to the request  
 178 and the `SAMPLE`, `EVENT`, and `CONDITION` data types for other components are not returned.  
 179 Another case would occur when an *MTConnect Agent* is supporting more than one piece of  
 180 equipment and data from only one piece of equipment is requested. Refer to *MTConnect*  
 181 *Standard Part 1.0 – Overview and Functionality, Section 5: MTConnect Fundamentals* for more  
 182 information on *sequence numbers*.

183

184 **4.1 Streams**

185 Streams is a container type XML element that **MUST** contain only DeviceStream  
 186 elements. Streams **MAY** contain any number of DeviceStream elements. If there is no  
 187 data to be reported for a request for data, an MTConnectStreams document **MUST** be  
 188 returned with an empty Streams container. *Data Entities* **MAY NOT** be directly associated  
 189 with the Streams container.

190 The following XML schema represents the structure of the Streams XML element.



191  
 192 **Figure 2: Streams Schema Diagram**

193

Element	Description	Occurrence
Streams	<p>The first, or highest, level XML container element in an <i>MTConnectStreams Response Document</i> provided by an <i>MTConnect Agent</i> in response to a sample or current <i>HTTP Request</i>.</p> <p>There <b>MAY</b> be only one Streams element in an <i>MTConnectStreams Response Document</i> for each piece of equipment represented in the document.</p> <p>An empty Streams container <b>MAY</b> be provided to indicate that no data is available for the given <i>Request</i>.</p> <p>The Streams element <b>MAY</b> contain any number of DeviceStream elements, one for each piece of equipment represented in the <i>MTConnectStreams</i> document.</p>	1

194

195

196 **4.2 DeviceStream**

197 DeviceStream is a XML container that organizes data reported from a single piece of  
 198 equipment. A DeviceStream element **MUST** be provided for *each* piece of equipment  
 199 reporting data in an MTConnectStreams document.

200 A DeviceStream **MAY** contain any number of ComponentStream elements; limited to  
 201 one for each component element represented in the MTConnectDevices document. If the  
 202 response to the request for data from an *MTConnect Agent* does not contain any data for a  
 203 specific piece of equipment, an empty DeviceStream element **MAY** be created to indicate  
 204 that the piece of equipment exists, but there was no data available. In this case, there will be no  
 205 ComponentStream elements provided.

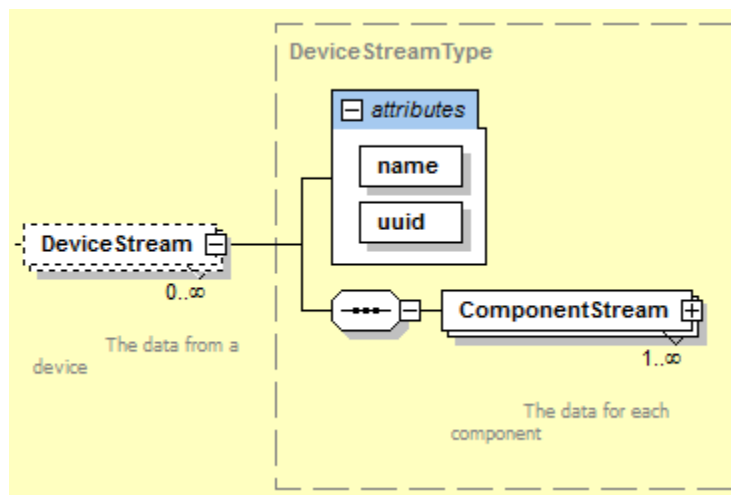
Element	Description	Occurrence
DeviceStream	A XML container element provided in the Streams container in the MTConnectStreams document.  There <b>MAY</b> be one or more DeviceStream elements in a Streams container; one for each piece of equipment represented in the MTConnectStreams document.	0..INF

206

207 **4.2.1 XML Schema for DeviceStream**

208 The following XML schema represents the structure of the DeviceStream XML element  
 209 showing the attributes defined for DeviceStream and the elements that **MAY** be associated  
 210 with DeviceStream.

211



212

213

**Figure 3: DeviceStream Schema Diagram**

214

215 **4.2.2 Attributes for DeviceStream**

216 The following table defines the attributes that **MUST** be provided to uniquely identify each  
 217 specific piece of equipment associated with the information provided in each DeviceStream.

218

Attribute	Description	Occurrence
name	<p>The name associated with the piece of equipment reporting the data contained in this DeviceStream container.</p> <p>name is a required attribute.</p> <p>The value reported for name <b>MUST</b> be the same as the value defined for the name attribute of the same piece of equipment in the MTConnectDevices document.</p> <p>An NMTOKEN XML type.</p> <p><b>WARNING:</b> name may become an optional attribute in future versions of the MTConnect Standard.</p>	1
uuid	<p>The uuid associated with the piece of equipment reporting the data contained in this DeviceStream container.</p> <p>uuid is a required attribute.</p> <p>The value reported for uuid <b>MUST</b> be the same as the value defined for the uuid attribute of the same piece of equipment in the MTConnectDevices document.</p>	1

219

220 **4.2.3 Elements for DeviceStream**

221 The following table lists the XML element(s) that **MAY** be provided in the DeviceStream  
 222 XML element.

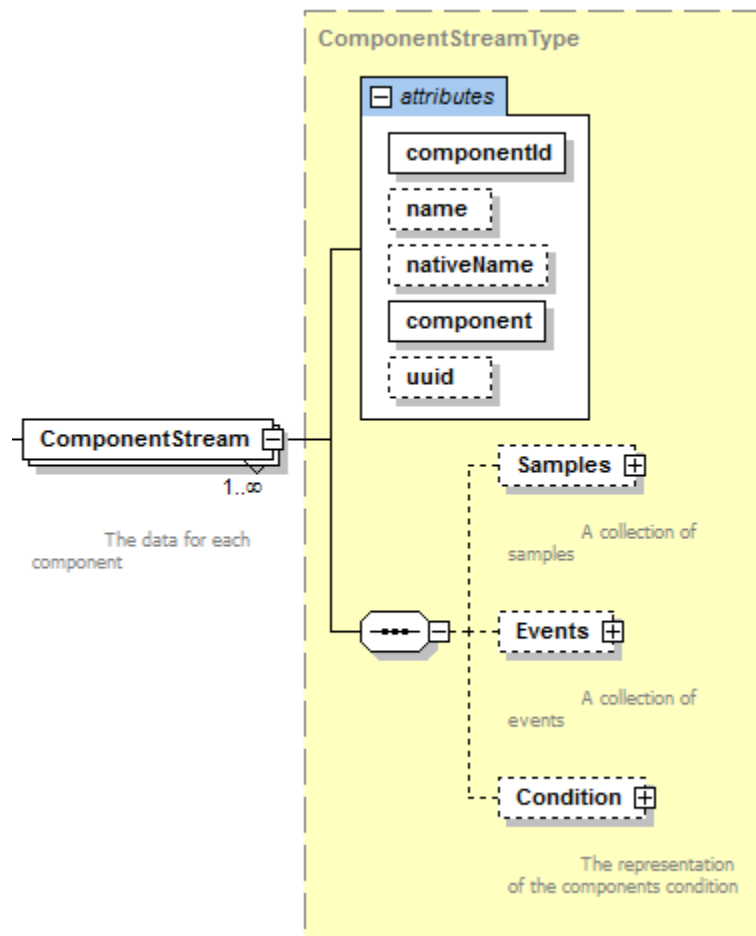
Element	Description	Occurrence
ComponentStream	<p>A XML container type element that organizes data returned from an <i>MTConnect Agent</i> in response to a current or sample HTTP request.</p> <p>Any number of ComponentStream elements <b>MAY</b> be provided in a DeviceStream container.</p> <p>There <b>MUST</b> be a separate ComponentStream XML element for each of the <i>Structural Elements</i> (Device elements, <i>Top Level Component</i> elements, or <i>Lower Level Component</i> elements) defined for that piece of equipment in the associated MTConnectDevices XML document. A ComponentStream representing a <i>Structural Element</i> will only appear if there is data reported for that <i>Structural Element</i>.</p>	0..INF

223 **4.3 ComponentStream**

224 ComponentStream is a XML container that organizes the data associated with each *Structural*  
 225 *Element* (Device element, *Top Level* Component, or *Lower Level* Component element)  
 226 defined for that piece of equipment in the associated MTConnectDevices XML document.  
 227 The data reported in each ComponentStream element **MUST** be grouped into individual  
 228 XML containers based on the value of the category attribute (SAMPLE, EVENT, or  
 229 CONDITION) defined for each *Data Entity* in the MTConnectDevices XML document.  
 230 These containers are Samples, Events, and Condition.

231 **4.3.1 XML Schema for ComponentStream**

232 The following XML schema represents the structure of a ComponentStream XML element  
 233 showing the attributes defined for ComponentStream and the elements that **MAY** be  
 234 associated with ComponentStream.



235  
 236 **Figure 4: ComponentStream Schema Diagram**  
 237



238 ComponentStream is similar to DeviceStream in that the attributes uniquely identify the  
 239 *Structural Element* with which the data reported is directly associated. This information does not  
 240 have to be repeated for each *Data Entity*. In the case of the DeviceStream, the attributes  
 241 uniquely identify the piece of equipment associated with the data. In the case of the  
 242 ComponentStream, the attributes identify the specific *Structural Element* within a piece of  
 243 equipment associated with each *Data Entity*.

#### 244 4.3.2 Attributes for ComponentStream

245 The following table defines the attributes used to uniquely identify the specific *Structural*  
 246 *Element(s)* of a piece of equipment associated with the data reported in the  
 247 MTConnectStreams document.

Attribute	Description	Occurrence
componentId	<p>The identifier of the <i>Structural Element</i> (Device element, <i>Top Level Component</i> element, or <i>Lower Level Component</i> element) as defined by the <i>id</i> attribute of the corresponding <i>Structural Element</i> in the MTConnectDevices XML document.</p> <p>componentId is a required attribute.</p> <p>The identifier <b>MUST</b> be the same as that defined in the MTConnectDevices document to associate the data reported in the ComponentStream container with the <i>Structural Element</i> identified in the MTConnectDevices document.</p>	1
name	<p>The name of the ComponentStream element.</p> <p>name is an optional attribute.</p> <p>If name is not defined for a specific <i>Structural Element</i> in the MTConnectDevices document, it <b>MUST NOT</b> be provided for the corresponding ComponentStream element in the MTConnectStreams document.</p> <p>If name is defined for a specific <i>Structural Element</i> in the MTConnectDevices document, it <b>MAY</b> be provided for the corresponding ComponentStream element in the MTConnectStreams document.</p> <p>If provided, the value reported for name <b>MUST</b> be the same as the value defined for the name attribute of the corresponding <i>Structural Element</i> (Device element, <i>Top Level Component</i> element, or <i>Lower Level Component</i> element) defined in the MTConnectDevices XML document.</p> <p>An NMTOKEN XML type.</p>	0..1

Attribute	Description	Occurrence
nativeName	<p>nativeName identifies the common name normally associated with the ComponentStream element.</p> <p>nativeName is an optional attribute.</p> <p>If nativeName is not defined for a specific <i>Structural Element</i> in the MTConnectDevices document, it <b>MUST NOT</b> be provided for the corresponding ComponentStream element in the MTConnectStreams document.</p> <p>If nativeName is defined for a specific <i>Structural Element</i> in the MTConnectDevices document, it <b>MAY</b> be provided for the corresponding ComponentStream element in the MTConnectStreams document.</p> <p>If provided, the value reported for nativeName <b>MUST</b> be the same as the value defined for the nativeName attribute of the corresponding <i>Structural Element</i> (Device element, <i>Top Level</i> Component element, or <i>Lower Level</i> Component element) defined in the MTConnectDevices XML document.</p>	0..1
component	<p>component identifies the <i>Structural Element</i> (Device, <i>Top Level</i> Component, or <i>Lower Level</i> Component) associated with the ComponentStream element.</p> <p>component is a required attribute.</p> <p>The value reported for component <b>MUST</b> be the same as the value defined for the <i>Element Name</i> of the XML container representing the corresponding <i>Structural Element</i> (Device element, <i>Top Level</i> Component element, or <i>Lower Level</i> Component element) defined in the MTConnectDevices XML document.</p> <p>Examples of component are Device, Axes, Controller, Linear, Electrical, User, and Loader.</p>	1

Attribute	Description	Occurrence
uuid	<p>uuid of the ComponentStream element.</p> <p>uuid is an optional attribute.</p> <p>If uuid is not defined for a specific <i>Structural Element</i> in the MTConnectDevices document, it <b>MUST NOT</b> be provided for the corresponding ComponentStream element in the MTConnectStreams document.</p> <p>If uuid is defined for a specific <i>Structural Element</i> in the MTConnectDevices document, it <b>MAY</b> be provided for the corresponding ComponentStream element in the MTConnectStreams document, but it is not required.</p> <p>If provided, the value reported for uuid <b>MUST</b> be the same as the value defined for the uuid attribute of the corresponding <i>Structural Element</i> (Device element, <i>Top Level</i> Component element, or <i>Lower Level</i> Component element) defined in the MTConnectDevices XML document.</p>	0..1

248

249 **4.3.3 Elements for ComponentStream**

250 In the ComponentStream container, an *MTConnect Agent* **MUST** organize the data reported  
 251 in each ComponentStream into individual Samples, Events, or Condition XML  
 252 containers based on the value of the category attribute (i.e., SAMPLE, EVENT, or CONDITION)  
 253 defined for each *Data Entity* defined in the MTConnectDevices XML document.

254 Each ComponentStream element **MUST** include at least one Events, Samples, or  
 255 Condition XML container element. *Data Entities* returned in each of the  
 256 ComponentStream container elements are defined in the table below.

Element	Description	Occurrence
Samples	<p>A XML container type element.</p> <p>Samples organizes the SAMPLE type <i>Data Entities</i> defined in the MTConnectDevices document that are reported in each ComponentStream XML element.</p>	0..1 *
Events	<p>A XML container type element.</p> <p>Events organizes the EVENT type <i>Data Entities</i> defined in the MTConnectDevices document that are reported in each ComponentStream XML element.</p>	0..1 *

Element	Description	Occurrence
Condition	<p>A XML container type element.</p> <p>Condition organizes the CONDITION type <i>Data Entities</i> defined in the MTConnectDevices document that are reported in each ComponentStream XML element.</p>	0..1 *

257

258

259

260

Note: \* The ComponentStream element **MUST** contain at least one of these element types.

261 **5 Data Entities**

262 When a piece of equipment reports values associated with `DataItem` elements defined in the  
 263 `MTConnectDevices` document, that information is organized as *Data Entities* in the  
 264 `MTConnectStreams` document. These *Data Entities* are organized in containers within each  
 265 `ComponentStream` element based on the `category` attribute defined for the corresponding  
 266 `DataItem` in the `MTConnectDevices` document:

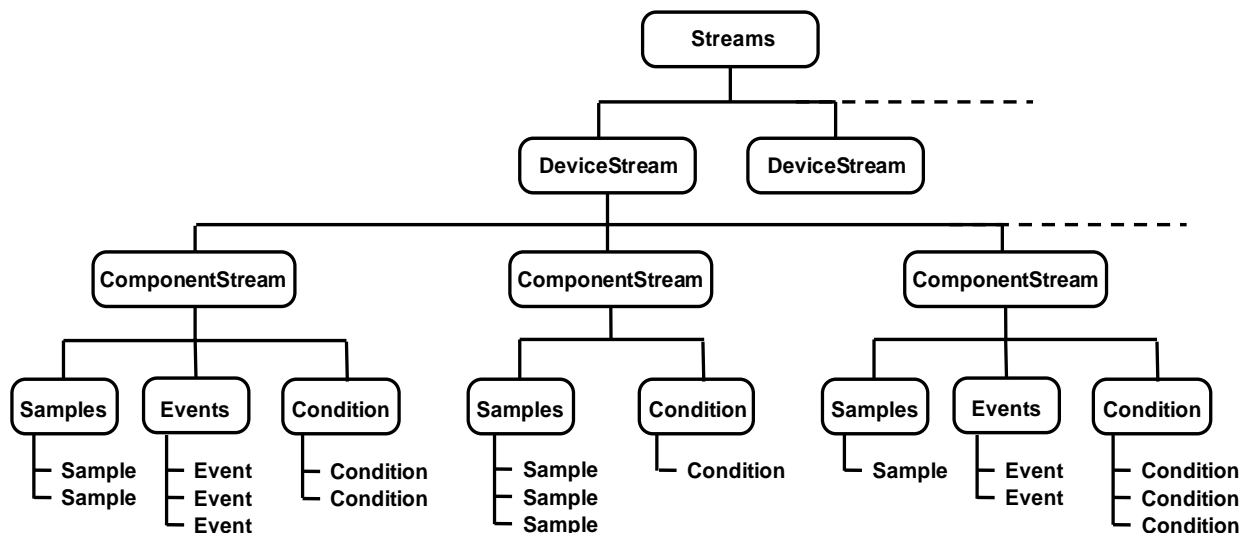
267 `DataItem` elements defined with a `category` attribute of `SAMPLE` in the  
 268 `MTConnectDevices` document are mapped to the `Samples` XML container in the  
 269 associated `ComponentStream` element.

270 `DataItem` elements defined with a `category` attribute of `EVENT` in the  
 271 `MTConnectDevices` document are mapped to the `Events` XML container in the  
 272 associated `ComponentStream` element.

273 `DataItem` elements defined with a `category` attribute of `CONDITION` in the  
 274 `MTConnectDevices` document are mapped to the `Condition` XML container in the  
 275 associated `ComponentStream` element.

276 The XML tree below demonstrates how *Data Entities* are organized in these containers.

277



278

279 **Figure 5: ComponentStream XML Tree Diagram**

280

281

282 The following is an illustration of the structure of an XML document demonstrating how *Data*  
 283 *Entities* are reported in a *MTConnectStreams* document:

```

284 1. <MTConnectStreams>
285 2.   <Header/>
286 3.   <Streams>
287 4.     <DeviceStream>
288 5.       <ComponentStream>
289 6.         <Samples>
290 7.           <Sample>
291 8.           <Sample>
292 9.           <Sample>
293 10.        </Samples>
294 11.        <Events>
295 12.          <Event>
296 13.          <Event>
297 14.        </Events>
298 15.        </Condition>
299 16.        <Condition>
300 17.        <Condition>
301 18.        </Condition>
302 19.      </ComponentStream>
303 20.      <ComponentStream>
304 21.        <Samples>
305 22.          <Sample>
306 23.          <Sample>
307 24.        </Samples>
308 25.        <Events>
309 26.          <Event>
310 27.          <Event>
311 28.          <Event>
312 29.        </Events>
313 30.        <Condition>
314 31.        <Condition>
315 32.        </Condition>
316 33.      </ComponentStream>
317 34.    </DeviceStream>
318 35.  </Streams>
319 36. </MTConnectStreams>

```

320

321 **Note:** There are no specific requirements defining the sequence in which the  
 322 *ComponentStream* XML elements are organized in the *MTConnectStreams*  
 323 document. They **MAY** be organized in any sequence based on the implementation of an  
 324 *MTConnect Agent*. The sequence in which the *ComponentStream* XML elements  
 325 appear does not impact the ability for a client software application to interpret the  
 326 information that it receives in the document.

327

328 When an *MTConnect Agent* responds to a `current` HTTP request, the information returned in  
329 the `MTConnectStreams` document **MUST** include the most current value for every *Data*  
330 *Entity* defined in the `MTConnectDevices` document subject to any filtering included within  
331 the request.

332 When an *MTConnect Agent* responds to a `sample` HTTP request, the information returned in  
333 the `MTConnectStreams` document **MUST** include the occurrences for each *Data Entity* that  
334 are available to an *MTConnect Agent* subject to filtering and the count parameter included within  
335 the request (see *Part 1 - Overview and Functionality* for a full definition of the protocol).

## 336 **5.1 *Element Names for Data Entities***

337 In the `MTConnectDevices` document, *Data Entities* are grouped as `DataItem XML`  
338 elements within each `Device`, *Top Level Component*, and *Lower Level Component*  
339 *Structural Element*. The *Data Entities* reported in the `MTConnectStreams` document  
340 associated with each of these *Structural Elements* are represented with an *Element Name* based  
341 on the category and type defined for each of the `DataItem` elements in the  
342 `MTConnectDevices` document.

### 343 **5.1.1 *Element Names when MTConnectDevices category is SAMPLE or*** 344 ***EVENT***

345 The *Data Entities* reported in the `MTConnectStreams` document associated with each  
346 `DataItem` element defined in the `MTConnectDevices` document with a category  
347 attribute of `SAMPLE` or `EVENT` **MUST** be identified in the `MTConnectStreams` document  
348 with an *Element Name* derived from the type attribute defined for that `DataItem` element in  
349 the `MTConnectDevices` document.

350

351 The example below describes the most common method used to derive the *Element Name* for a  
 352 *Data Entity* reported in the MTConnectStreams document from the information describing  
 353 that DataItem element in the MTConnectDevices document:

#### 354 **DataItem Represented in the MTConnectDevices Document**

```
355 1. <DataItem type="AXIS_FEEDRATE" id="xf" name="Xfirt"  
356 2.   category="SAMPLE" units="MILLIMETER/SECOND"  
357 3.   nativeUnits="MILLIMETER/SECOND"/>
```

358 • DataItem: The XML *Element Name* for this *Data Entity*.

359 Note: *Element Name* must not be confused with the name attribute for the data  
 360 item element.

361 • type, category, units, and nativeUnits: Attributes that provide  
 362 additional information regarding each data item in the MTConnectDevices  
 363 document.

#### 364 **Response Format reported in the MTConnectStreams Document**

```
365 1. <AxisFeedrate name="Xfirt" sequence="61315517" timestamp="2016-07-  
366 2.   28T02:06:01.364428Z" dataItemId="xf">10.83333</AxisFeedrate>
```

367 • AxisFeedrate: The *Element Name* provided in the MTConnectStreams  
 368 response format for the data item. The *Element Name* for a data item is defined by  
 369 the type attribute of AXIS\_FEEDRATE in the MTConnectDevices  
 370 document. The *Element Name* **MUST** be provided in Pascal case format (first  
 371 letter of each word is capitalized).

### 372 **5.1.2 Changes to *Element Names* when representation attribute is used**

373 The *Element Name* for a *Data Entity* reported in the MTConnectStreams document is  
 374 extended when the representation attribute is used to further describe that DataItem  
 375 element in the MTConnectDevices document.

376 When a DataItem element is defined in the MTConnectDevices document with a  
 377 representation attribute of TIME\_SERIES or DISCRETE, the XML *Element Name* for  
 378 the associated *Data Entity* reported in the MTConnectStreams document **MUST** be extended  
 379 by adding the value of the representation attribute to the *Element Name*.

380 For example, the DataItem element ANGULAR\_VELOCITY with a representation  
 381 attribute defined as TIME\_SERIES **MUST** be transformed to the *Element Name*  
 382 AngularVelocityTimeSeries.

383 Similarly, the DataItem element PART\_COUNT with a representation attribute defined  
 384 as DISCRETE **MUST** be transformed to the *Element Name* PartCountDiscrete.



385 **5.1.3 Element Names when MTConnectDevices category is CONDITION**

386 *Data Entities* defined in the MTConnectDevices document with a *category* attribute of  
 387 CONDITION are reported with an *Element Name* that is defined differently from other *Data*  
 388 *Entity* types. The *Element Name* for these *Data Entities* are defined based on the *Fault State*  
 389 (Normal, Warning, or Fault) associated with each *Data Entity* at the time that a value for  
 390 that *Data Entity* is reported. See *Sections 5.7.1 and 5.8* for details on how these *Data Entities* are  
 391 reported in the MTConnectStreams document.

392 **5.2 Samples Container**

393 *Samples* is a XML container type element. *Samples* organizes the *Data Entities* returned in  
 394 the MTConnectStreams XML document for those *DataItem* elements defined with a  
 395 *category* attribute of SAMPLE in the MTConnectDevices document.

396 A separate *Samples* container will be provided for the data returned for the *DataItem*  
 397 elements associated with each *Structural Element* of a piece of equipment defined in the  
 398 MTConnectDevices document.

399

Element	Description	Occurrence
Samples	<p>A XML container type element that organizes the data reported in the MTConnectStreams document for <i>DataItem</i> elements defined in the MTConnectDevices document with a <i>category</i> attribute of SAMPLE.</p> <p>A separate <i>Samples</i> container <b>MUST</b> be provided for each <i>ComponentStream</i> element for which data is returned for a <i>DataItem</i> element defined in the MTConnectDevices document with a <i>category</i> attribute of SAMPLE.</p> <p>If provided in the document, a <i>Samples</i> XML container <b>MUST</b> contain at least one <i>Sample</i> element.</p>	0..1

400

401

402 **5.3 Sample Data Entities**

403 A Sample XML element provides the information and data reported from a piece of equipment  
 404 for those DataItem elements defined with a category attribute of SAMPLE in the  
 405 MTConnectDevices document.

406 Sample is an abstract type XML element and will never appear directly in the  
 407 MTConnectStreams XML document. As an abstract type XML element, Sample will be  
 408 replaced in the XML document by a specific type of Sample specified by the *Element Name* for  
 409 that *Data Entity*. The different types of Sample elements are defined in *Section 6.1*. Examples  
 410 of XML elements representing Sample include PathPosition, Temperature, and  
 411 AxisVelocity.

Element	Description	Occurrence
Sample	<p>A XML element that provides the information and data reported from a piece of equipment for those DataItem elements defined with a category attribute of SAMPLE in the MTConnectDevices document.</p> <p>Sample is an abstract type XML element. It is replaced in the MTConnectStreams document by a specific type of Sample element.</p> <p>There <b>MAY</b> be multiple types of Sample elements in a Samples container.</p>	1..INF

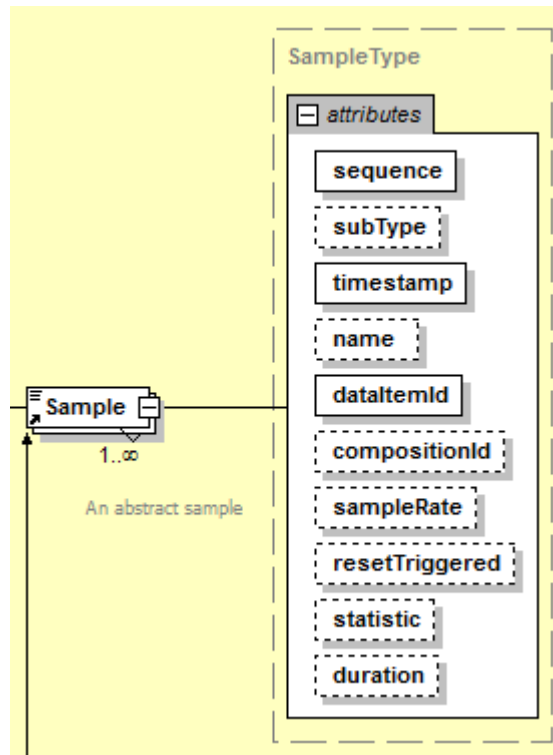
412

413

414 **5.3.1 XML Schema Structure for Sample**

415 The following XML schema represents the structure of a Sample XML element showing the  
 416 attributes defined for Sample elements.

417



418

419

**Figure 6: Sample Schema Diagram**

420

421 **5.3.2 Attributes for Sample**

422 The following table defines the attributes used to provide additional information for a Sample  
 423 XML element.

424

Attribute	Description	Occurrence
sequence	<p>A number representing the sequential position of an occurrence of the Sample in the data buffer of an <i>MTConnect Agent</i>.</p> <p>sequence is a required attribute.</p> <p>sequence <b>MUST</b> have a value represented as an unsigned 64-bit value from 1 to 2<sup>64</sup>-1.</p>	1
subType	<p>The subtype of the <i>Data Entity</i>.</p> <p>subType is an optional attribute.</p> <p>subType <b>MUST</b> match the subType attribute of the DataItem element as defined in the <i>MTConnectDevices</i> document that the Sample element represents.</p>	0..1
timestamp	<p>The most accurate time available to a piece of equipment that represents the point in time that the data reported for the Sample was measured.</p> <p>When the Sample element represents a DataItem element defined in the <i>MTConnectDevices</i> document with a representation or statistic attribute, timestamp <b>MUST</b> represent the time that the data collection was completed.</p> <p>timestamp is a required attribute.</p>	1
name	<p>The name of the Sample element.</p> <p>name is an optional attribute.</p> <p>name <b>MUST</b> match the name attribute of the DataItem element defined in the <i>MTConnectDevices</i> document that the Sample element represents.</p> <p>An NMTOKEN XML type.</p>	0..1
dataItemId	<p>The unique identifier for the Sample element.</p> <p>dataItemId is a required attribute.</p> <p>dataItemId <b>MUST</b> match the id attribute of the DataItem element defined in the <i>MTConnectDevices</i> document that the Sample element represents.</p>	1

Attribute	Description	Occurrence
sampleRate	<p>The rate at which successive samples of the value of a data item are recorded. <code>sampleRate</code> is expressed in terms of samples per second.</p> <p><code>sampleRate</code> is an optional attribute.</p> <p>If the <code>sampleRate</code> is smaller than one, the number can be represented as a decimal type floating-point number. For example, a rate of 1 per 10 seconds would be 0.1</p> <p><code>sampleRate</code> <b>MUST</b> be provided when the <code>representation</code> attribute of the <code>DataItem</code> element defined in the <code>MTConnectDevices</code> document that this <code>Sample</code> element represents is <code>TIME_SERIES</code>.</p> <p>For <code>DataItem</code> elements where the <code>representation</code> attribute defined in the <code>MTConnectDevices</code> document that this <code>Sample</code> element represents is not <code>TIME_SERIES</code>, it <b>MUST</b> be assumed that the data reported is represented by a single value and <code>sampleRate</code> <b>MUST NOT</b> be reported in the <code>MTConnectStreams</code> document.</p>	0..1
statistic	<p>The type of statistical calculation defined by the <code>statistic</code> attribute of the <code>DataItem</code> element defined in the <code>MTConnectDevices</code> document that this <code>Sample</code> element represents.</p> <p><code>statistic</code> is an optional attribute.</p>	0..1
duration	<p>The time-period over which the data was collected.</p> <p><code>duration</code> is an optional attribute.</p> <p><code>duration</code> <b>MUST</b> be provided when the <code>statistic</code> attribute of the <code>DataItem</code> element is defined in the <code>MTConnectDevices</code> document that this <code>Sample</code> element represents.</p>	0..1
resetTriggered	<p>For those <code>DataItem</code> elements that report data that may be periodically reset to an initial value, <code>resetTriggered</code> identifies when a reported value has been reset and what has caused that reset to occur.</p> <p><code>resetTriggered</code> is an optional attribute.</p> <p><code>resetTriggered</code> <b>MUST</b> only be provided for the specific occurrence of a <i>Data Entity</i> reported in the <code>MTConnectStreams</code> document when the reset occurred and <b>MUST NOT</b> be provided for any other occurrence of the <i>Data Entity</i> reported in a <code>MTConnectStreams</code> document.</p>	0..1
compositionId	<p>The identifier of the <code>Composition</code> element defined in the <code>MTConnectDevices</code> document associated with the data reported for the <code>Sample</code> element.</p> <p><code>compositionId</code> is an optional attribute.</p>	0..1

### 425 5.3.2.1 duration Attribute for Sample

426 Sample elements that represent the result of a computed value of a statistic **MUST** contain  
427 a duration attribute. For these *Data Entities*, the timestamp associated with the Sample  
428 **MUST** reference the time the data collection was completed. timestamp **MUST NOT**  
429 represent any other time associated with the data collection or the calculation of the statistic. The  
430 actual time the interval began can be computed by subtracting the duration from the  
431 timestamp.

432 Two Sample elements **MAY** have overlapping time periods when statistics are computed at  
433 different frequencies. For example, there may be two *Data Entities* reporting a statistic  
434 representing the average value for the readings of the same measured signal calculated over one  
435 and five minute intervals. These *Data Entities* can both have the same start time for their  
436 calculations (e.g., 05:10:00), but the timestamp and duration will be 05:11:00 and 60  
437 seconds, respectively, for the *Data Entity* reporting the one-minute average and 05:15:00 and  
438 300 seconds, respectively, for the *Data Entity* reporting the five-minute average. This allows for  
439 varying statistical methods to be applied with different interval lengths each having different  
440 values for the timestamp and duration attributes.

### 441 5.3.2.2 resetTriggered Attribute for Sample

442 Some *Data Entities* **MAY** have their reported value reset to an initial value. These reset actions  
443 may be based upon a specific elapsed time or may be triggered by a physical or logical reset  
444 action that causes the reset to occur. Examples of *Data Entities* that **MAY** have their reported  
445 value reset to an initial value are *Data Entities* representing a counter, a timer, or a statistic.

446 resetTriggered defines the type of reset action that caused the value of the reported data to  
447 be reset. The value reported for resetTriggered **MAY** be defined by the ResetTrigger  
448 element for the *Data Entity* in the MTConnectDevices document that this Sample element  
449 represents. If the ResetTrigger element is not defined in the MTConnectDevices  
450 document, a resetTriggered attribute **SHOULD** be reported in the MTConnectStreams  
451 document if the type of reset action can be determined and reported by the piece of equipment.

452 resetTriggered **MUST** only be reported for the first occurrence of a *Data Entity* after a  
453 reset action has occurred and **MUST NOT** be provided for any other occurrence of the *Data*  
454 *Entity* reported in a MTConnectStreams document. When a reset occurs, the piece of  
455 equipment **MUST** report an occurrence of the *Data Entity* that was reset even if that occurrence  
456 of the *Data Entity* would normally be suppressed based on the filtering criteria established in the  
457 MTConnectDevices document that this Sample element represents.

458

459 The following table provides the values that **MAY** be reported for `resetTriggered`:

Value for <code>resetTriggered</code>	Description
ACTION_COMPLETE	The value of the <i>Data Entity</i> that is measuring an action or operation was reset upon completion of that action or operation.
ANNUAL	The value of the <i>Data Entity</i> was reset at the end of a 12-month period.
DAY	The value of the <i>Data Entity</i> was reset at the end of a 24-hour period.
MAINTENANCE	The value of the <i>Data Entity</i> was reset upon completion of a maintenance event.
MANUAL	The value of the <i>Data Entity</i> was reset based on a physical reset action.
MONTH	The value of the <i>Data Entity</i> was reset at the end of a monthly period.
POWER_ON	The value of the <i>Data Entity</i> was reset when power was applied to the piece of equipment after a planned or unplanned interruption of power has occurred.
SHIFT	The value of the <i>Data Entity</i> was reset at the end of a work shift.
WEEK	The value of the <i>Data Entity</i> was reset at the end of a 7-day period.

460

461

462 **5.3.3 Response for SAMPLE category DataItem Elements with a**  
463 **representation attribute of TIME\_SERIES**

464 SAMPLE category DataItem elements defined in the MTConnectDevices document with a  
465 representation attribute of TIME\_SERIES **MUST** be represented in the  
466 MTConnectStreams document as Sample elements that report data that includes multiple  
467 values representing a series of readings of a measured value taken at a specific sample rate.  
468 Such a DataItem element can be defined for collecting high frequency readings of a measured  
469 value and then providing the entire series of values to a client software application as the data  
470 reported for a single *Data Entity*. In this case, the sampleCount and sampleRate attributes  
471 **MUST** be provided.

472 Note: sampleCount is an attribute **MUST** only be provided for Sample elements that  
473 represent SAMPLE category DataItem elements defined in the  
474 MTConnectDevices document with a representation attribute of  
475 TIME\_SERIES.

476 The CDATA provided for the *Data Entity* **MUST** be a series of space delimited floating-point  
477 numbers. The number of values **MUST** match the sampleCount.

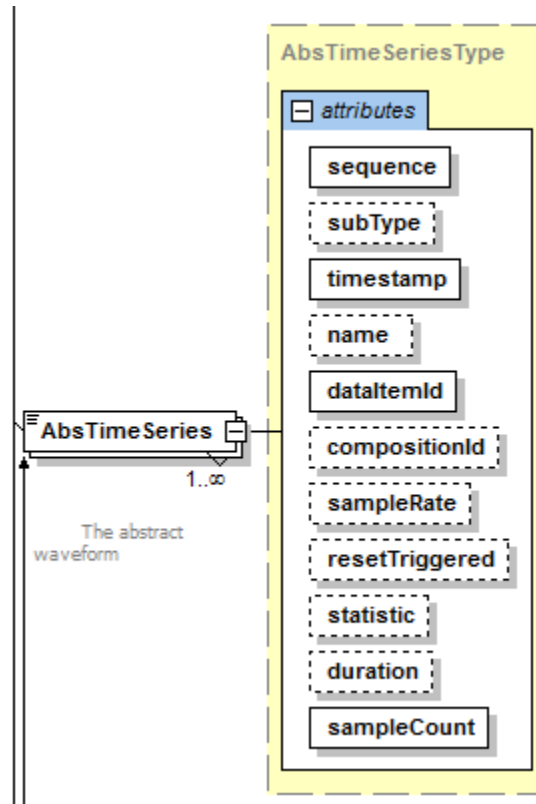
478



479 **5.3.3.1 XML Schema Structure for Sample when reporting Time Series data**

480 The following XML schema represents the extended structure of a Sample XML element that  
 481 represents a SAMPLE category DataItem element defined in the MTConnectDevices  
 482 document with a representation attribute of TIME\_SERIES.

483



484

485 **Figure 7: AbsTimeSeries Schema Diagram**

486

487 Note: The AbsTimeSeries element shown in the XML schema is an abstract type element  
 488 and will be replaced in the MTConnectStreams document by the element name  
 489 derived from the type attribute defined for the associated DataItem element defined  
 490 in the MTConnectDevices document.

491

492 **5.3.3.2 Attributes for a Sample when reporting Time Series data**

493 The following table defines the additional attribute provided for a Sample XML element that  
 494 represents a SAMPLE category DataItem element defined in the MTConnectDevices  
 495 document with a representation attribute of TIME\_SERIES.

496

Attribute	Description	Occurrence
sampleCount	The number of readings reported in the data returned for the DataItem element defined in the MTConnectDevices document that this Sample element represents.  sampleCount is an optional attribute.  sampleCount <b>MUST</b> be provided when the representation attribute of the DataItem element is TIME_SERIES.  sampleCount <b>MUST NOT</b> be provided when the representation attribute is defined as DISCRETE or VALUE, or when it is not defined.	0..1

497

498 **5.3.4 Valid Data Values for Sample**

499 All Sample elements reported in an MTConnectStreams XML document **MUST** provide a  
 500 value in the CDATA of the *Data Entity*.

501 The value returned in the CDATA **MUST** be reported as either a *Valid Data Value* representing  
 502 the information reported from a piece of equipment or UNAVAILABLE when a *Valid Data Value*  
 503 cannot be determined.

504 The *Valid Data Value* reported for a Sample represents the reading of the value of a  
 505 continuously variable or analog data source.

506 The representation attribute for a SAMPLE category DataItem element defined in the  
 507 MTConnectDevices document specifies how an *MTConnect Agent* **MUST** record instances  
 508 of the data associated with that data item and how often that data **MUST** be reported as a  
 509 Sample element in the MTConnectStreams document.

510

511 The data reported for a `Sample` element associated with a `SAMPLE` category `DataItem`  
 512 element with a representation of `VALUE` can be measured at any point-in-time and **MUST**  
 513 always produce a result with a single data value.

514 Note: If a `representation` attribute is not specified in the `MTConnectDevices`  
 515 document for a `DataItem` element, it **MUST** be assumed that the data reported in the  
 516 `MTConnectStreams` document for the *Data Entity* has a representation type  
 517 of `VALUE`.

518 In the case of a `Sample` element associated with a `SAMPLE` category `Data`  
 519 `Item` element with a `representation` attribute of `TIME_SERIES`, the data provided  
 520 **MUST** be a series of data values representing multiple sequential samples of the measured value  
 521 that will be provided only at the end of the completion of a sampling period. (See *Section 5.3.3*  
 522 of this document for more information on `TIME_SERIES` type data).

523 Data values provided for a `Sample` **MUST** always be a floating-point number. In the  
 524 `MTConnect` Standard, floating-point numbers are defined as XML `xs:float` type numbers as  
 525 defined by W3C. Any of the following number formats are valid XML floating type numbers:  
 526 1267.43233E12, -1E4, 12.78e-2, 12, 137.2847, 0, and INF.

527 Note: For some `Sample` elements, the *Valid Data Value* **MAY** be restricted to specific  
 528 formats. See *Section 6.1* of this document for a description of any restrictions of the  
 529 acceptable format for *Valid Data Values*.

530 For `Sample` elements, a client software application can determine the appropriate accuracy of  
 531 the value reported for the *Data Entity* by applying the `significantDigits` attribute defined  
 532 for the corresponding `DataItem` element defined in the `MTConnectDevices` document.

533 The *Valid Data Value* reported as `CDATA` for a `Sample` element **MUST** be formatted as part of  
 534 the content between the element tags in the XML element representing that *Data Entity*. As an  
 535 example, a `Position` is formatted as follows in the XML document:

```
536 1. <Position sequence="112" timestamp="2007-08-09T12:32:45.1232"  
537 2.   name="Xabs" dataItemId="10">123.3333</Position>
```

538 Note: The **BOLDED** item is identified for emphasis only.

539 In this example, the 123.3333 is the `CDATA` for `Position`. All `CDATA` in a `Sample` element is  
 540 *typed*, which means that the value reported for the *Data Entity* **MUST** be formatted as defined in *Section*  
 541 *6.1* for each *Data Entity* so that it can be validated.

542

543 **5.3.5 Unavailability of *Valid Data Values* for Sample**

544 If an *MTConnect Agent* cannot determine a *Valid Data Value* for a *Sample* element, the value  
 545 returned for the CDATA for the *Data Entity* **MUST** be reported as UNAVAILABLE .

546 The example below demonstrates how an *MTConnect Agent* reports the value for a *Sample* in  
 547 the CDATA when it is unable to determine a *Valid Data Value*:

```

548 1. <Samples>
549 2.   <PathPosition dataItemId="p2" timestamp="2009-03-04T19:45:50.458305"
550 3.     subType="ACTUAL" name="Zact"
551 4.     sequence="15065113">UNAVAILABLE</PathPosition>
552 5.   <Temperature dataItemId="t6"
553 6.     timestamp="2009-03-04T19:45:50.458305"
554 7.     name="temp" sequence="150651134">UNAVAILABLE</Temperature>
555 8. </Samples>
    
```

556

557 Note: The **BOLDED** items are identified for emphasis only.

558 **5.4 Events Container**

559 Events is a XML container type element. Events organizes the *Data Entities* returned in the  
 560 *MTConnectStreams* XML document for those *DataItem* elements defined with a  
 561 category attribute of EVENT in the *MTConnectDevices* document.

562 A separate Events container will be provided for the data returned for the *DataItem* elements  
 563 associated with each *Structural Element* of a piece of equipment defined in the  
 564 *MTConnectDevices* document.

565

Element	Description	Occurrence
Events	<p>A XML container type element that organizes the data reported in the <i>MTConnectStreams</i> document for <i>DataItem</i> elements defined in the <i>MTConnectDevices</i> document with a category attribute of EVENT.</p> <p>A separate Events container <b>MUST</b> be provided for each <i>ComponentStream</i> element for which data is returned for a <i>DataItem</i> element defined in the <i>MTConnectDevices</i> document with a category attribute of EVENT.</p> <p>If provided in the document, an Events XML container <b>MUST</b> contain at least one Event element.</p>	0..1

566

567

568 **5.5 Event Data Entities**

569 An Event XML element provides the information and data provided from a piece of equipment  
 570 for those DataItem elements defined with a category attribute of EVENT in the  
 571 MTConnectDevices document.

572 Event is an abstract type XML element and will never appear directly in the  
 573 MTConnectStreams XML document. As an abstract type XML element, Event will be  
 574 replaced in the XML document by a specific type of Event specified by the *Element Name* for  
 575 that *Data Entity*. The different types of Event elements are defined in *Section 6.2*. Examples  
 576 of XML elements representing Event include Block, Execution, and Line.

577 Event is similar to Sample, but its value can change with unpredictable frequency. Events  
 578 do not report intermediate values. As an example, when Availability transitions from  
 579 UNAVAILABLE to AVAILABLE, there is no intermediate state that can be inferred.

580 Event elements **MAY** report data values defined by a controlled vocabulary as specified in *Section 6.2*  
 581 of this document, by numeric values, or by a character string representing text or a message provided by  
 582 the piece of equipment.

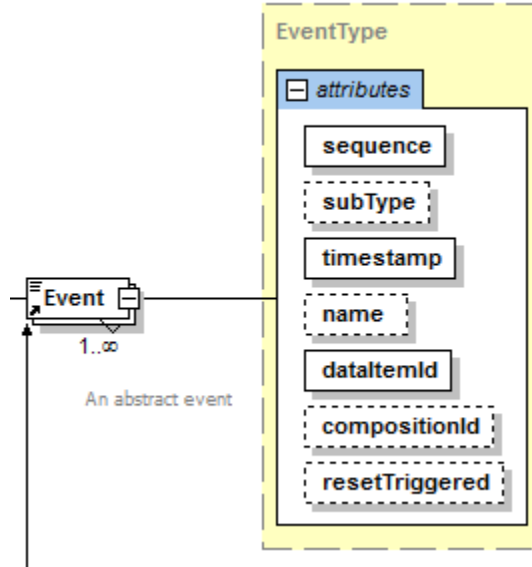
Element	Description	Occurrence
Event	<p>A XML element which provides the information and data reported from a piece of equipment for those DataItem elements defined with a category attribute of EVENT in the MTConnectDevices document.</p> <p>Event is an abstract type XML element. It is replaced in the MTConnectStreams document by a specific type of Event element.</p> <p>There <b>MAY</b> be multiple types of Event elements in an Events container.</p>	1..INF

583

584

585 **5.5.1 XML Schema Structure for Event**

586 The following XML schema represents the structure of an Event XML element showing the  
 587 attributes defined for Event elements.



588

589

**Figure 8: Event Schema Diagram**

590 **5.5.2 Attributes for Event**

591 The following table defines the attributes that **MAY** be used to provide additional information  
 592 for an Event XML element.

Attribute	Description	Occurrence
sequence	A number representing the sequential position of an occurrence of the Event in the data buffer of an <i>MTCConnect Agent</i> . sequence is a required attribute. sequence <b>MUST</b> have a value represented as an unsigned 64-bit value from 1 to 2 <sup>64</sup> -1.	1
subType	The subtype of the <i>Data Entity</i> . subType is an optional attribute. subType <b>MUST</b> match the subType attribute of the DataItem element as defined in the <i>MTCConnectDevices</i> document that the Event element represents.	0..1
timestamp	The most accurate time available to a piece of equipment that represents the point in time that the data reported for the Event was measured. timestamp is a required attribute.	1

Attribute	Description	Occurrence
name	<p>The name of the Event element.</p> <p>name is an optional attribute.</p> <p>name <b>MUST</b> match the name attribute of the DataItem element as defined in the MTConnectDevices document that the Event element represents.</p> <p>An NMTOKEN XML type.</p>	0..1
dataItemId	<p>The unique identifier for the Event element.</p> <p>dataItemId is a required attribute.</p> <p>dataItemId <b>MUST</b> match the id attribute of the DataItem element as defined in the MTConnectDevices document that the Event element represents.</p>	1
resetTriggered	<p>For those DataItem elements that report data that <b>MAY</b> be periodically reset to an initial value, resetTriggered identifies when a reported value has been reset and what that has caused that reset to occur.</p> <p>resetTriggered is an optional attribute.</p> <p>resetTriggered <b>MUST</b> only be provided for the specific occurrence of a <i>Data Entity</i> reported in the MTConnectStreams document when the reset occurred and <b>MUST NOT</b> be provided for any other occurrence of the <i>Data Entity</i> reported in a MTConnectStreams document.</p>	0..1
compositionId	<p>The identifier of the Composition element defined in the MTConnectDevices document that the data reported for the Event element is associated.</p> <p>compositionId is an optional attribute.</p>	0..1

593

### 594 **5.5.3 Response for EVENT category Data Items with a representation** 595 **attribute of DISCRETE**

596 EVENT category DataItem elements defined in an MTConnectDevices document with a  
597 representation attribute of DISCRETE indicate that the value of successive occurrences of  
598 the data reported in the associated Event type *Data Entity* in an MTConnectStreams  
599 document **MAY** be identical. Duplicate values **MUST NOT** be suppressed by an *MTConnect*  
600 *Agent* since each occurrence of the data item represents a different and unique Event.

601

602 An example of an EVENT category DataItem element with a representation attribute of  
603 DISCRETE would be a parts counter that reports the completion of each part produced, versus  
604 reporting the accumulation of parts produced over time. In this case, the associated Event  
605 element would be represented by a *Data Entity* with an *Element Name* of  
606 PartCountDiscrete. Each occurrence of this *Data Entity* in an MTConnectStreams  
607 document would indicate the completion of a fixed number of parts (typically 1).

#### 608 **5.5.4 Response for EVENT category Data Items with a type attribute of** 609 **MESSAGE**

610 EVENT category DataItem elements defined in the MTConnectDevices document with a  
611 type attribute of MESSAGE **MAY NOT** report a state change between successive occurrences  
612 of the associated *Data Entity* being reported by a piece of equipment in the  
613 MTConnectStreams document. If the *Data Entity* representing a message does not have a  
614 reset state, it **SHOULD** be defined with a representation attribute of DISCRETE in the  
615 MTConnectDevices document. In this case, each occurrence of this *Data Entity* in an  
616 MTConnectStreams document represents a different and unique Event. The *Element Name*  
617 for this Event element **MUST** be MessageDiscrete and each occurrence of this *Data*  
618 *Entity* in an MTConnectStreams document would indicate a unique occurrence of the  
619 message.

#### 620 **5.5.5 Valid Data Values for Event**

621 Event elements reported in an MTConnectStreams XML document **MUST** provide a value  
622 in the CDATA of the *Data Entity*.

623 The value reported in the CDATA **MUST** be reported as either a *Valid Data Value* representing  
624 the information reported from a piece of equipment or UNAVAILABLE when a *Valid Data Value*  
625 cannot be determined.

626 The *Valid Data Value* reported for an Event represents a distinct piece of information provided  
627 from a piece of equipment. Unlike Sample, Event does not report intermediate values that  
628 vary over time. Event reports information that, when provided at any specific point in time,  
629 represents the current state of the piece of equipment.

630 The representation attribute for an EVENT category data item defined in the  
631 MTConnectDevices document specifies how an *MTConnect Agent* **MUST** record instances  
632 of data associated with that data item and how that data **MUST** be reported as an Event  
633 element in the MTConnectStreams document.

634 The data reported for an Event element associated with an EVENT category data item with a  
635 representation attribute of VALUE **MUST** be either an integer, a floating-point number, a  
636 descriptive value (text string) representing one of two or more state values defined for that data  
637 item, or a text string representing a message.



638 If a representation attribute is not specified for a data item in an MTConnectDevices  
 639 document, the designation for the representation attribute **MUST** be interpreted as  
 640 VALUE.

641 The data reported for an Event element associated with an EVENT category data item with a  
 642 representation attribute of DISCRETE **MUST** be a numeric value representing a repetitive  
 643 occurrence of a single data value or a message. An EVENT with a representation attribute  
 644 of DISCRETE is the only case where an *MTConnect Agent* **MAY** provide successive  
 645 occurrences of a data item with identical data values since each occurrence of the Event  
 646 element represents a different and unique occurrence of the *Data Entity*.

647 The *Valid Data Value* reported as CDATA for an Event element **MUST** be formatted as part of  
 648 the content between the element tags in the XML element representing that *Data Entity*. As an  
 649 example, Event elements are formatted as follows in the XML document:

```
650 1. <PartCount dataItemId="pc4" timestamp="2009-02-26T02:02:36.48303"
651 2.   name="pcount" sequence="185">238</PartCount>
652 3. <ControllerMode dataItemId="p3" timestamp="2009-02-26T02:02:35.716224"
653 4.   name="mode" sequence="192">AUTOMATIC</ControllerMode>
654 5.   <Block dataItemId="cn2" name="block" sequence="206"
655 6.     timestamp="2009-02-26T02:02:37.394055">G0Z1</Block>
```

656 Note: The **BOLDED** items are identified for emphasis only.

657 In these examples, 238 is the CDATA for PartCount and is a numeric value; AUTOMATIC is  
 658 the CDATA for the ControllerMode and is a descriptive value representing a state for the  
 659 *Data Entity*; and G0Z1 is a text string representing a message describing the program code  
 660 associated with the Block *Data Entity*.

### 661 **5.5.6 Unavailability of *Valid Data Values* for Event**

662 If an *MTConnect Agent* cannot determine a *Valid Data Value* for an Event element, the value  
 663 returned for the CDATA for the *Data Entity* **MUST** be reported as UNAVAILABLE.

664 The example below demonstrates how an *MTConnect Agent* reports the value for an Event in  
 665 the CDATA when it is unable to determine a *Valid Data Value*:

```
666 1. <Events>
667 2.   <ControllerMode dataItemId="p3" timestamp="2009-02-26T02:02:35.716224"
668 3.     name="mode" sequence="182">UNAVAILABLE</ControllerMode>
669 4. </Events>
```

670 Note: The **BOLDED** items are identified for emphasis only.

## 671 **5.6 Condition Container**

672 Condition is a XML container type element. Condition organizes the *Data Entities*  
 673 returned in the MTConnectStreams XML document for those DataItem elements defined  
 674 with a category attribute of CONDITION in the MTConnectDevices document.

675 A separate Condition container will be provided for the data returned for the DataItem  
 676 elements associated with each *Structural Element* of a piece of equipment defined in the  
 677 MTConnectDevices document.

678

Element	Description	Occurrence
Condition	<p>A XML container type element that organizes the data reported in the MTConnectStreams document for DataItem elements defined in the MTConnectDevices document with a category attribute of CONDITION.</p> <p>A separate Condition container <b>MUST</b> be provided for each ComponentStream element for which data is returned for a DataItem element defined in the MTConnectDevices document with a category attribute of CONDITION.</p> <p>If provided in the document, a Condition XML container <b>MUST</b> contain at least one Condition data element.</p>	0..1

679

680 **5.7 Condition Data Entities**

681 A Condition XML element provides the information and data provided from a piece of  
 682 equipment for those DataItem elements defined with a category attribute of CONDITION  
 683 in the MTConnectDevices document.

684 Condition provides information reported by a piece of equipment describing its health and  
 685 ability to function.

686 Condition is an abstract type XML element and will never appear directly in the  
 687 MTConnectStreams XML document. As an abstract type XML element, Condition will  
 688 be replaced in the XML document by a *Data Entity* representing the CONDITION category  
 689 DataItem element defined in the MTConnectDevices document that this Condition  
 690 element represents.

691

692 The *Data Entities* represented by *Condition* are structured differently than the *Data Entities*  
 693 representing *Sample* and *Event*. The *Element Name* for each *Condition* element reported  
 694 in the *MTConnectStreams* document defines the *Fault State* of the *Data Entity*. A  
 695 *Condition* element is identified by the *Structural Element* to which it is associated, along with  
 696 the *type* and *dataItemId* defined for the element. *Section 6.3* provides details on the  
 697 different types of *Condition* elements.

698

Element	Description	Occurrence
Condition	<p>A XML element that provides the information and data reported from a piece of equipment for those <i>DataItem</i> elements defined with a <i>category</i> attribute of <i>CONDITION</i> in the <i>MTConnectDevices</i> document.</p> <p><i>Condition</i> is an abstract type XML element. It is replaced in the <i>MTConnectStreams</i> document by a specific type of <i>Condition</i> element.</p> <p>There <b>MAY</b> be multiple types of <i>Condition</i> elements in a <i>Condition</i> container.</p>	1..INF

699

700 *CONDITION* type *DataItem* elements defined in the *MTConnectDevices* document **MAY**  
 701 report multiple simultaneous *Fault States* in the *MTConnectStreams* document. This is  
 702 unlike a *SAMPLE* or *EVENT* *DataItem* element that can only report a single occurrence of a  
 703 *Sample* or *Event* element in the *MTConnectStreams* document at any one point in time.

704 For example, a controller on a piece of equipment may detect and report multiple format errors  
 705 in a motion program. Each error represents a separate *Fault State* from the controller. Each  
 706 *Fault State* is represented as a separate *Condition* element in the *MTConnectStreams*  
 707 document since each *Fault State* **MUST** be identified and tracked individually in the document.

### 708 **5.7.1 Element Names for Condition**

709 *Condition* elements are reported differently from other *Data Entity* types. The *Element Name*  
 710 reported for a *Condition* element represents the *Fault State* (*Normal*, *Warning*, or *Fault*)  
 711 associated with each *Condition*.

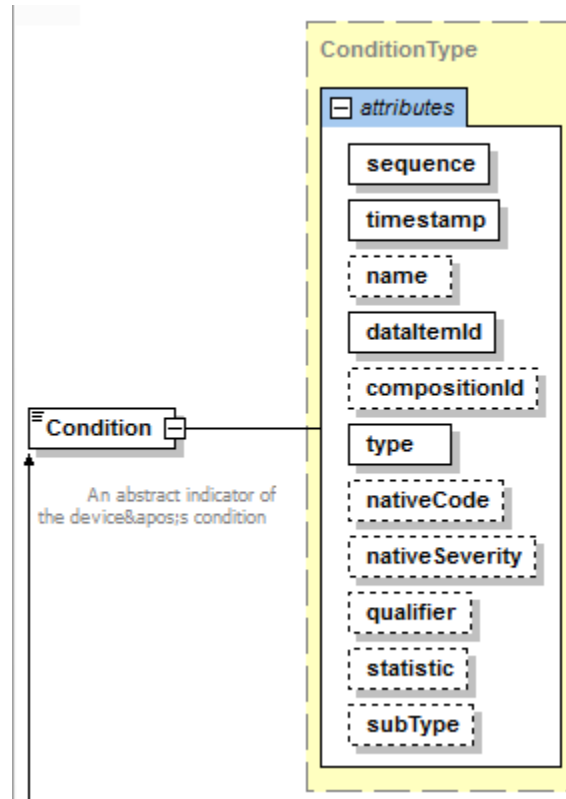
712 Examples of XML elements representing *Condition* elements for each of the possible *Fault*  
 713 *States* are:

- 714 1. `<Normal type="MOTION_PROGRAM" dataItemId="cc2" sequence="25"`
- 715 2. `timestamp="2010-04-06T06:19:35.153141">/Normal>`
- 716 3. `<Fault type="COMMUNICATIONS" dataItemId="cc1" sequence="26"`
- 717 4. `nativeCode="IO1231" timestamp="2010-04-`
- 718 5. `06T06:19:35.153141">Communications error</Fault>`
- 719 6. `<Warning type="LOGIC_PROGRAM" dataItemId="pm6" sequence="32"`
- 720 7. `timestamp="2010-04-06T06:19:35.153141">Warning/>`

721 Note: The **BOLDED** item is identified for emphasis only.

722 **5.7.2 XML Schema Structure for Condition**

723 The following XML schema represents the structure of a Condition XML element showing  
 724 the attributes defined for Condition elements.



725

726

**Figure 9: Condition Schema Diagram**

727

728 **5.7.3 Attributes for Condition**

729 The following table defines the attributes used to provide additional information for a  
 730 Condition XML element.

Attribute	Description	Occurrence
sequence	A number representing the sequential position of an occurrence of the Condition in the data buffer of an <i>MTCConnect Agent</i> . sequence is a required attribute. sequence <b>MUST</b> have a value represented as an unsigned 64-bit value from 1 to 2 <sup>64</sup> -1.	1

Attribute	Description	Occurrence
timestamp	<p>The most accurate time available to a piece of equipment that represents the point in time that the data reported for the Condition was measured or detected.</p> <p>timestamp is a required attribute.</p>	1
name	<p>The name of the Condition element.</p> <p>name is an optional attribute.</p> <p>name <b>MUST</b> match the name attribute of the DataItem element as defined in the MTConnectDevices document that this Condition element represents.</p> <p>An NMTOKEN XML type.</p>	0..1
dataItemId	<p>The unique identifier for the Condition element.</p> <p>dataItemId is a required attribute.</p> <p>dataItemId <b>MUST</b> match the id attribute of the DataItem element defined in the MTConnectDevices document that this Condition element represents.</p>	1
type	<p>An identifier of the type of fault represented by the Condition element.</p> <p>type is a required attribute.</p> <p>type <b>MUST</b> match the type attribute of the DataItem element defined in the MTConnectDevices document that this Condition element represents.</p>	1
nativeCode	<p>The native code (usually an alpha-numeric value) generated by the controller of a piece of equipment providing a reference identifier for a Condition.</p> <p>nativeCode is an optional attribute.</p> <p>This is the same information an operator or maintenance personnel may see as a reference code designating a specific fault code provided by the piece of equipment.</p>	0..1
nativeSeverity	<p>If the piece of equipment designates a severity level to a fault, nativeSeverity reports that severity information to a client software application.</p> <p>nativeSeverity is an optional attribute.</p>	0..1

Attribute	Description	Occurrence
qualifier	<p>qualifier provides additional information regarding a <i>Fault State</i> associated with the measured value of a process variable.</p> <p>qualifier is an optional attribute.</p> <p>qualifier defines whether the <i>Fault State</i> represented by the Condition indicates a measured value that is above or below an expected value of a process variable.</p> <p>If the <i>Fault State</i> represents a measured value that is greater than the expected value for the process variable, qualifier <b>MUST</b> report a value of HIGH.</p> <p>If the <i>Fault State</i> represents a measured value that is less than the expected value for the process variable, qualifier <b>MUST</b> report a value of LOW.</p>	0..1
statistic	<p>statistic provides additional information describing the meaning of the Condition element.</p> <p>statistic is an optional attribute.</p> <p>statistic <b>MUST</b> match the statistic attribute of the DataItem element defined in the MTConnectDevices document that this Condition element represents.</p>	0..1
subType	<p>subType provides additional information describing the meaning of the Condition element.</p> <p>subType is an optional attribute.</p> <p>subType <b>MUST</b> match the subType attribute of the DataItem element defined in the MTConnectDevices document that this Condition element represents.</p>	0..1
compositionId	<p>The identifier of the Composition element defined in the MTConnectDevices document that the data reported for this Condition element represents.</p> <p>compositionId is an optional attribute.</p>	0..1
xs:lang	<p>An optional attribute that specifies the language of the CDATA returned for the Condition.</p> <p>Refer to IETF RFC 4646 (<a href="http://www.ietf.org/rfc/rfc4646.txt">http://www.ietf.org/rfc/rfc4646.txt</a>) or successor for a full definition of the values for this attribute.</p> <p>xs:lang does not appear in the schema diagram.</p>	0..1

731

732

### 733 5.7.3.1 **qualifier Attribute for Condition**

734 Many Condition elements report the *Fault State* associated with the measured value of a  
735 process variable.

736 `qualifier` provides an indication whether the measured value is above or below an expected  
737 value of a process variable

738 As an example, a Condition element with a `type` attribute of AMPERAGE may differentiate  
739 between a higher than expected amperage and a lower than expected amperage by using the  
740 `qualifier` attribute.

741 When a `qualifier` of either HIGH or LOW is used with `Fault` and `Warning`, the *Fault*  
742 *States* can be differentiated as follows:

743        `Fault, LOW`  
744        `Warning, LOW`  
745        `Normal`  
746        `Warning, HIGH`  
747        `Fault, HIGH`

748 The following is an example of an XML element representing Condition using  
749 `qualifier`:

```
750 1. <Warning type="FILL_LEVEL" dataItemId="pm6" qualifier="HIGH"
751 2.   sequence="32" timestamp="2009-11-13T08:32:18">...</Warning>
```

752 Note: The `qualifier` attribute of “high” is **BOLDED** for emphasis only.

### 753 5.7.4 **Valid Data Values for Condition**

754 Condition elements reported in an `MTConnectStreams` XML document **MAY** provide a  
755 value in the CDATA of the *Data Entity* when additional information regarding the *Fault State* is  
756 available.

757 A *Valid Data Value* for the CDATA included in a Condition element **MAY** be any text  
758 string. A *Valid Data Value* is not required to be reported for a Condition category *Data*  
759 *Entity*. The *Fault State* and the attributes provided in a Condition element **MAY** be sufficient  
760 to fully describe the *Data Entity*.

761

762 The *Valid Data Value* reported as CDATA for a Condition element **MUST** be formatted as  
 763 part of the content between the element tags in the XML element representing that *Data Entity*.  
 764 As an example, Condition elements are formatted as follows in the XML document:

```
765 1. <Warning type="FILL_LEVEL" dataItemId="pm6" qualifier="HIGH"
766 2.     sequence="32" timestamp="2009-11-13T08:32:18">Fill Level on Tank
767 3.     #12 is reaching a high level</Warning>
```

768 Note: The **BOLDED** items are identified for emphasis only.

769 In this example, the “Fill Level on Tank #12 is reaching a high level” is the CDATA for the *Data*  
 770 *Entity*.

## 771 **5.8 Unavailability of Fault State for Condition**

772 When an *MTCConnect Agent* cannot determine a valid *Fault State* for a Condition element, it  
 773 **MUST** report the *Element Name* for the *Data Entity* as Unavailable.

774 The example below demonstrates how an *MTCConnect Agent* reports a Condition category  
 775 *Data Entity* when it is unable to determine a valid *Fault State*:

```
776 1. <Unavailable type="MOTION_PROGRAM" dataItemId="cc2" sequence="25"
777 2.     timestamp="2009-11-13T08:32:18">...</Unavailable>
778 3. <Unavailable type="COMMUNICATIONS" dataItemId="cc1" sequence="26"
779 4.     timestamp="2009-11-13T08:32:18">...</Unavailable>
780 5. <Unavailable type="LOGIC_PROGRAM" dataItemId="cc3" sequence="28"
781 6.     timestamp="2009-11-13T08:32:18">...</Unavailable>
782 7. <Unavailable type="LOGIC_PROGRAM" dataItemId="pm6" sequence="32"
783 8.     timestamp="2009-11-13T08:32:18">...</Unavailable>
```

784 Note: The **BOLDED** items are identified for emphasis only.

785



## 786 6 Listing of Data Entities

787 *Data Entities* that report data in MTConnectStreams documents are represented by Sample,  
 788 Event, or Condition elements based upon the category and type attributes defined for  
 789 the corresponding DataItem XML element in the MTConnectDevices document.

790 Each *Data Entity* in the MTConnectStreams document has an *Element Name*, as defined in  
 791 the following sections, based upon the corresponding category attribute defined for that  
 792 DataItem element in the MTConnectDevices document.

### 793 6.1 Sample Element Names

794 The following is a list of the XML elements that can be placed in the Samples container of the  
 795 ComponentStream element.

796 The table shows both the type attribute for each SAMPLE category DataItem element as  
 797 defined in the MTConnectDevices document and the corresponding *Element Name* for the  
 798 *Data Entity* that **MUST** be reported as a Sample element in the MTConnectStreams  
 799 document.

SAMPLE Data Item Type	Sample Element Name	Description
ACCELERATION	Acceleration	The measurement of the rate of change of velocity. Acceleration <b>MUST</b> be reported in units of MILLIMETER/SECOND^2.
ACCUMULATED_TIME	AccumulatedTime	The measurement of accumulated time for an activity or event. AccumulatedTime <b>MUST</b> be reported in units of SECOND. <b>DEPRECATION WARNING:</b> May be deprecated in the future. Recommend using ProcessTimer and MachineTimer.
ANGULAR_ACCELERATION	AngularAcceleration	The measurement of the rate of change of angular velocity. AngularAcceleration <b>MUST</b> be reported in units of DEGREE/SECOND^2.
ANGULAR_VELOCITY	AngularVelocity	The measurement of the rate of change of angular position. AngularVelocity <b>MUST</b> be reported in units of DEGREE/SECOND.

SAMPLE Data Item Type	Sample Element Name	Description
AMPERAGE	Amperage	<p>The measurement of electrical current.</p> <p>Subtypes of Amperage are ALTERNATING, DIRECT, ACTUAL, and TARGET.</p> <p>If a subType is not specified, the reported value for the data <b>MUST</b> default to the subtype of ACTUAL.</p> <p>Amperage <b>MUST</b> be reported in units of AMPERE.</p>
ANGLE	Angle	<p>The measurement of angular position.</p> <p>Subtypes of Angle are ACTUAL and COMMANDED.</p> <p>If a subType is not specified, the reported value for the data <b>MUST</b> default to the subtype of ACTUAL.</p> <p>Angle <b>MUST</b> be reported in units of DEGREE.</p>
AXIS_FEEDRATE	AxisFeedrate	<p>The measurement of the feedrate of a linear axis.</p> <p>Subtypes of AxisFeedrate are ACTUAL, COMMANDED, JOG, PROGRAMMED, and RAPID.</p> <p>If a subType is not specified, the reported value for the data <b>MUST</b> default to the subtype of PROGRAMMED.</p> <p>AxisFeedrate <b>MUST</b> be reported in units of MILLIMETER/SECOND.</p>
CLOCK_TIME	ClockTime	<p>The value provided by a timing device at a specific point in time.</p> <p>ClockTime <b>MUST</b> be reported in W3C ISO 8601 format of YYYY-MM-DDThh:mm:ss.ffff.</p>
CONCENTRATION	Concentration	<p>The measurement of the percentage of one component within a mixture of components.</p> <p>Concentration <b>MUST</b> be reported in units of PERCENT.</p>
CONDUCTIVITY	Conductivity	<p>The measurement of the ability of a material to conduct electricity.</p> <p>Conductivity <b>MUST</b> be reported in units of SIEMENS/METER.</p>

<b>SAMPLE Data Item Type</b>	<b>Sample Element Name</b>	<b>Description</b>
DISPLACEMENT	Displacement	<p>The measurement of the change in position of an object.</p> <p>Displacement <b>MUST</b> be reported in units of MILLIMETER.</p>
ELECTRICAL_ENERGY	ElectricalEnergy	<p>The measurement of electrical energy consumption by a component.</p> <p>ElectricalEnergy <b>MUST</b> be reported in units of WATT_SECOND.</p>
EQUIPMENT_TIMER	EquipmentTimer	<p>The measurement of the amount of time a piece of equipment or a sub-part of a piece of equipment has performed specific activities.</p> <p>Subtypes of EquipmentTimer are LOADED, WORKING, OPERATING, POWERED, and DELAY.</p> <p>A subType <b>MUST</b> always be specified.</p> <p>EquipmentTimer <b>MUST</b> be reported in units of SECOND.</p>
FILL_LEVEL	FillLevel	<p>The measurement of the amount of a substance remaining compared to the planned maximum amount of that substance.</p> <p>FillLevel <b>MUST</b> be reported in units of PERCENT.</p>
FLOW	Flow	<p>The measurement of the rate of flow of a fluid.</p> <p>Flow <b>MUST</b> be reported in units of LITER/SECOND.</p>
FREQUENCY	Frequency	<p>The measurement of the number of occurrences of a repeating event per unit time.</p> <p>Frequency <b>MUST</b> be reported in units of HERTZ.</p>
<del>GLOBAL_POSITION</del>	<del>GlobalPosition</del>	<b>DEPRECATED</b> in <i>Version 1.1.0</i> .
<del>LEVEL</del>	<del>Level</del>	<p><b>DEPRECATED</b> in <i>Version 1.2.0</i>.</p> <p>See FILL_LEVEL</p>

SAMPLE Data Item Type	Sample <i>Element Name</i>	Description
LENGTH	Length	<p>The measurement of the length of an object.</p> <p>Subtypes of Length are STANDARD, REMAINING, and USEABLE .</p> <p>If a subType is not specified, the reported value for the data <b>MUST</b> default to the subtype of REMAINING.</p> <p>Length <b>MUST</b> be reported in units of MILLIMETER.</p>
LINEAR_FORCE	LinearForce	<p>The measurement of the push or pull introduced by an actuator or exerted on an object.</p> <p>LinearForce <b>MUST</b> be reported in units of NEWTON.</p>
LOAD	Load	<p>The measurement of the actual versus the standard rating of a piece of equipment.</p> <p>Load <b>MUST</b> be reported in units of PERCENT.</p>
MASS	Mass	<p>The measurement of the mass of an object(s) or an amount of material.</p> <p>Mass <b>MUST</b> be reported in units of KILOGRAM.</p>
PATH_FEEDRATE	PathFeedrate	<p>The measurement of the feedrate for the axes, or a single axis, associated with a Path component– a vector.</p> <p>Subtypes of PathFeedrate are ACTUAL, COMMANDED, JOG, PROGRAMMED, and RAPID.</p> <p>If a subType is not specified, the reported value for the data <b>MUST</b> default to the subtype of PROGRAMMED.</p> <p>PathFeedrate <b>MUST</b> be reported in units of MILLIMETER/SECOND.</p>

<p><b>SAMPLE</b> <b>Data Item Type</b></p>	<p><b>Sample</b> <b>Element Name</b></p>	<p><b>Description</b></p>
<p>PATH_POSITION</p>	<p>PathPosition</p>	<p>A measured or calculated position of a control point reported by the CONTROLLER element of a piece of equipment expressed in WORK coordinates. The coordinate system will revert to MACHINE coordinates if WORK coordinates are not available.</p> <p>Subtypes of PathPosition are ACTUAL, PROGRAMMED, COMMANDED, TARGET, and PROBE .</p> <p>If a subType is not specified, the reported value for the data <b>MUST</b> default to the subtype of ACTUAL.</p> <p>PathPosition <b>MUST</b> be reported as a set of space-delimited floating-point numbers representing a point in 3-D space. The position of the control point <b>MUST</b> be reported in units of MILLIMETER and listed in order of X, Y, and Z referenced to the coordinate system of the piece of equipment.</p> <p>An example of the value reported for PathPosition would be:</p> <pre>&lt;PathPosition ...&gt;10.123 55.232 100.981 &lt;/PathPosition&gt;</pre> <p>Where X = 10.123, Y = 55.232, and Z=100.981.</p>
<p>PH</p>	<p>Ph</p>	<p>The measurement of acidity or alkalinity.</p> <p>PH <b>MUST</b> be reported in units of PH.</p>

SAMPLE Data Item Type	Sample Element Name	Description
POSITION	Position	<p>A measured or calculated position of a component element as reported by a piece of equipment.</p> <p>Subtypes of Position are ACTUAL, COMMANDED, PROGRAMMED, and TARGET.</p> <p>If a subType is not specified, the reported value for the data <b>MUST</b> default to the subtype of ACTUAL.</p> <p>When Position is provided representing a measured value for the physical axes of the piece of equipment, the data <b>MUST</b> be provided in MACHINE coordinates.</p> <p>When Position is provided representing a logical or calculated position, the data <b>MUST</b> be provided in WORK coordinates and is associated with a Path element of the equipment controller.</p> <p>Position <b>MUST</b> be reported in units of MILLIMETER.</p>
POWER_FACTOR	PowerFactor	<p>The measurement of the ratio of real power flowing to a load to the apparent power in that AC circuit.</p> <p>PowerFactor <b>MUST</b> be reported in units of PERCENT.</p>
PRESSURE	Pressure	<p>The measurement of the force per unit area exerted by a gas or liquid.</p> <p>Pressure <b>MUST</b> be reported in units of PASCAL.</p>
PROCESS_TIMER	ProcessTimer	<p>The measurement of the amount of time a piece of equipment has performed different types of activities associated with the process being performed at that piece of equipment.</p> <p>Subtypes of ProcessTimer are PROCESS and DELAY.</p> <p>A subType <b>MUST</b> always be specified.</p> <p>ProcessTimer <b>MUST</b> be reported in units of SECOND.</p>
RESISTANCE	Resistance	<p>The measurement of the degree to which a substance opposes the passage of an electric current.</p> <p>Resistance <b>MUST</b> be reported in units of OHM.</p>

SAMPLE Data Item Type	Sample Element Name	Description
ROTARY_VELOCITY	RotaryVelocity	<p>The measurement of the rotational speed of a rotary axis.</p> <p>Subtypes of RotaryVelocity are ACTUAL, COMMANDED, and PROGRAMMED.</p> <p>If a subType is not specified, the reported value for the data <b>MUST</b> default to the subtype of ACTUAL.</p> <p>RotaryVelocity <b>MUST</b> be reported in units of REVOLUTION/MINUTE.</p>
SOUND_LEVEL	SoundLevel	<p>The measurement of a sound level or sound pressure level relative to atmospheric pressure.</p> <p>Subtypes of SoundLevel are NO_SCALE, A_SCALE, B_SCALE, C_SCALE, and D_SCALE.</p> <p>If a subType is not specified, the reported value for the data <b>MUST</b> default to the subtype of NO_SCALE.</p> <p>SoundLevel <b>MUST</b> be provided in DECIBEL.</p>
SPINDLE_SPEED	SpindleSpeed	<p><b>DEPRECATED</b> in <i>Version 1.2.0</i>.</p> <p>Replaced by ROTARY_VELOCITY</p>
STRAIN	Strain	<p>The measurement of the amount of deformation per unit length of an object when a load is applied.</p> <p>Strain <b>MUST</b> be reported in units of PERCENT.</p>
TEMPERATURE	Temperature	<p>The measurement of temperature.</p> <p>Temperature <b>MUST</b> be reported in units of degrees CELSIUS.</p>
TENSION	Tension	<p>The measurement of a force that stretches or elongates an object.</p> <p>Tension <b>MUST</b> be reported in units of NEWTON.</p>
TILT	Tilt	<p>A measurement of angular displacement.</p> <p>Tilt <b>MUST</b> be reported in units of MICRO_RADIAN.</p>
TORQUE	Torque	<p>The measurement of the turning force exerted on an object or by an object.</p> <p>Torque <b>MUST</b> be reported in units of NEWTON_METER.</p>

SAMPLE Data Item Type	Sample Element Name	Description
VOLT_AMPERE	VoltAmpere	<p>The measurement of the apparent power in an electrical circuit, equal to the product of root-mean-square (RMS) voltage and RMS current (commonly referred to as VA).</p> <p>VoltAmpere <b>MUST</b> be reported in units of VOLT_AMPERE.</p>
VOLT_AMPERE_REACTIVE	VoltAmpereReactive	<p>The measurement of reactive power in an AC electrical circuit (commonly referred to as VAR).</p> <p>VoltAmpereReactive <b>MUST</b> be reported in units of VOLT_AMPERE_REACTIVE.</p>
VELOCITY	Velocity	<p>The measurement of the rate of change of position of a component.</p> <p>When provided as the Velocity of the Axes component, it represents the value of the velocity vector for all given axes, similar to PathFeedrate.</p> <p>When provided as the Velocity of an individual axis component, it represents the value of the velocity for that specific axis with no influence of the relative velocity of any other axes.</p> <p>Velocity <b>MUST</b> be reported in units of MILLIMETER/SECOND.</p>
VISCOSITY	Viscosity	<p>A measurement of a fluid's resistance to flow.</p> <p>Viscosity <b>MUST</b> be reported in units of PASCAL_SECOND.</p>
VOLTAGE	Voltage	<p>The measurement of electrical potential between two points.</p> <p>Subtypes of Voltage are ALTERNATING, DIRECT, ACTUAL, and TARGET.</p> <p>If a subType is not specified, the reported value for the data <b>MUST</b> default to the subType of ACTUAL.</p> <p>Voltage <b>MUST</b> be reported in units of VOLT.</p>



SAMPLE Data Item Type	Sample <i>Element Name</i>	Description
WATTAGE	Wattage	<p>The measurement of power flowing through or dissipated by an electrical circuit or piece of equipment.</p> <p>Subtypes of Wattage are ACTUAL and TARGET.</p> <p>If a subType is not specified, the reported value for the data <b>MUST</b> default to the subtype of ACTUAL.</p> <p>Wattage <b>MUST</b> be reported in units of WATT.</p>

800

801 Note: The Sample response format **MUST** be extended when the representation  
 802 attribute for the data item is TIME\_SERIES. See Section 5.3.3 of this document for  
 803 details on extending the response format.

804 **6.2 Event Element Names**

805 The following is a list of the XML elements that can be placed in the Events container of the  
 806 ComponentStream element.

807 The table shows both the type for each EVENT category DataItem element defined in the  
 808 MTConnectDevices document and the corresponding *Element Name* for the *Data Entity* that  
 809 **MUST** be reported as an Event element in the MTConnectStreams document.

810 The table also defines the *Valid Data Values* for those Event type data items where the reported  
 811 values are restricted to a *Controlled Vocabulary*.

812

EVENT Data Item Type	Event <i>Element Name</i>	Description and <i>Valid Data Values</i>
ACTUATOR_STATE	ActuatorState	<p>ActuatorState represents the operational state of an apparatus for moving or controlling a mechanism or system.</p> <p><i>Valid Data Values:</i></p> <ul style="list-style-type: none"> <li>- ACTIVE: The actuator is operating</li> <li>- INACTIVE: The actuator is not operating</li> </ul>

<b>EVENT</b> <b>Data Item Type</b>	<b>Event</b> <b>Element Name</b>	<b>Description and</b> <b>Valid Data Values</b>
<del>ALARM</del>	<del>Alarm</del>	<b>DEPRECATED:</b> Replaced with CONDITION category data items in <i>Version 1.1.0</i> .
ACTIVE_AXES	ActiveAxes	<p>The set of axes currently associated with a Path or Controller <i>Structural Element</i>.</p> <p>The <i>Valid Data Value</i> reported <b>SHOULD</b> be a space-delimited set of axes names. The names returned <b>SHOULD</b> match the name attribute of the Linear or Rotary <i>Structural Elements</i> defined in the MTConnectDevices document that this Event element represents. If name is not available, nativeName <b>MUST</b> be returned to identify the Linear or Rotary <i>Structural Elements</i>.</p> <p>For example:</p> <p style="padding-left: 40px;">&lt;ActiveAxes ...&gt;X Y Z W S&lt;/ActiveAxes&gt;</p> <p>where X, Y, Z, W, and S are the nativeName attributes of the <i>Structural Elements</i>.</p> <p>If it is not specified elsewhere in the MTConnectDevices document, it <b>MUST</b> be assumed that all of the axes are associated with the Path component.</p>
AVAILABILITY	Availability	<p>Represents an <i>MTConnect Agent's</i> ability to communicate with the data source.</p> <p>Availability <b>MUST</b> be provided for each Device <i>Structural Element</i> and <b>MAY</b> be provided for any other <i>Structural Element</i>.</p> <p><i>Valid Data Values:</i></p> <ul style="list-style-type: none"> <li>- AVAILABLE: The <i>Structural Element</i> is active and capable of providing data.</li> <li>- UNAVAILABLE: The <i>Structural Element</i> is either inactive or not capable of providing data.</li> </ul>

<b>EVENT</b> <b>Data Item Type</b>	<b>Event</b> <i>Element Name</i>	<b>Description and</b> <i>Valid Data Values</i>
AXIS_COUPLING	AxisCoupling	<p>Describes the way axes are associated to each other.</p> <p>This is used in conjunction with COUPLED_AXES to indicate the interaction between axes.</p> <p>The coupling of the axes <b>MUST</b> be viewed from the perspective of a specified axis. Therefore, a MASTER coupling indicates that this axis is the master for the COUPLED_AXES.</p> <p>AxisCoupling <b>MUST</b> be provided for each axis element associated with a set of axes defined by the COUPLED_AXES data item element defined in the MTConnectDevices document.</p> <p><i>Valid Data Values:</i></p> <ul style="list-style-type: none"> <li>- TANDEM: The axes are physically connected to each other and operate as a single unit.</li> <li>- SYNCHRONOUS: The axes are not physically connected to each other but are operating together in lockstep.</li> <li>- MASTER: The axis is the master of the CoupledAxes</li> <li>- SLAVE: The axis is a slave to the CoupledAxes</li> </ul>

<b>EVENT</b> <b>Data Item Type</b>	<b>Event</b> <b>Element Name</b>	<b>Description and</b> <b>Valid Data Values</b>
AXIS_FEEDRATE_OVERRIDE	AxisFeedrateOverride	<p>The value of a signal or calculation issued to adjust the feedrate of an individual linear type axis.</p> <p>The value provided for AxisFeedrateOverride is expressed as a percentage of the designated feedrate for the axis.</p> <p>Subtypes of AxisFeedrateOverride are JOG, PROGRAMMED, and RAPID.</p> <p>If a subType is not specified, the reported value for the data <b>MUST</b> default to the subtype of PROGRAMMED.</p> <p>The <i>Valid Data Value</i> <b>MUST</b> be a floating-point number.</p>
AXIS_INTERLOCK	AxisInterlock	<p>An indicator of the state of the axis lockout function when power has been removed and the axis is allowed to move freely.</p> <p><i>Valid Data Values:</i></p> <ul style="list-style-type: none"> <li>- ACTIVE : The axis lockout function is activated, power has been removed from the axis, and the axis is allowed to move freely.</li> <li>- INACTIVE: The axis lockout function has not been activated, the axis may be powered, and the axis is capable of being controlled by another component.</li> </ul>
AXIS_STATE	AxisState	<p>An indicator of the controlled state of a LINEAR or ROTARY component representing an axis.</p> <p><i>Valid Data Values:</i></p> <ul style="list-style-type: none"> <li>- HOME: The axis is in its home position.</li> <li>- TRAVEL: The axis is in motion</li> <li>- PARKED: The axis has been moved to a fixed position and is being maintained in that position either electrically or mechanically. Action is required to release the axis from this position.</li> <li>- STOPPED: The axis is stopped</li> </ul>

EVENT Data Item Type	Event <i>Element Name</i>	Description and <i>Valid Data Values</i>
BLOCK	Block	<p>The line of code or command being executed by a Controller <i>Structural Element</i>.</p> <p>Block <b>MUST</b> include the entire expression for a line of program code, including all parameters</p> <p>The <i>Valid Data Value</i> <b>MUST</b> be any text string.</p>
BLOCK_COUNT	BlockCount	<p>The total count of the number of blocks of program code that have been executed since execution started.</p> <p>The <i>Valid Data Value</i> <b>MUST</b> be an integer.</p>
CHUCK_INTERLOCK	ChuckInterlock	<p>An indication of the state of an interlock function or control logic state intended to prevent the associated CHUCK component or composition element from being operated.</p> <p>A CHUCK component or composition element may be controlled by more than one type of ChuckInterlock function. When the ChuckInterlock function is provided by an operator controlled interlock that can inhibit the ability to initiate an unclamp action of an electronically controlled chuck, this ChuckInterlock function <b>SHOULD</b> be further characterized by specifying a subType of MANUAL_UNCLAMP.</p> <p><i>Valid Data Values:</i></p> <ul style="list-style-type: none"> <li>- ACTIVE: The chuck cannot be unclamped</li> <li>- INACTIVE: The chuck can be unclamped.</li> </ul>

<b>EVENT</b> <b>Data Item Type</b>	<b>Event</b> <i>Element Name</i>	<b>Description and</b> <i>Valid Data Values</i>
CHUCK_STATE	ChuckState	<p>An indication of the operating state of a mechanism that holds a part or stock material during a manufacturing process. It may also represent a mechanism that holds any other item in place within a piece of equipment.</p> <p><i>Valid Data Values:</i></p> <ul style="list-style-type: none"> <li>- OPEN: The CHUCK component or composition element is open to the point of a positive confirmation</li> <li>- CLOSED: The CHUCK component or composition element is closed to the point of a positive confirmation</li> <li>- UNLATCHED: The CHUCK component or composition element is not closed to the point of a positive confirmation and not open to the point of a positive confirmation. It is in an intermediate position.</li> </ul>
CODE	Code	<p><b>DEPRECATED</b> in <i>Version 1.1.0</i>.</p>

<p><b>EVENT</b> <b>Data Item Type</b></p>	<p><b>Event</b> <b>Element Name</b></p>	<p><b>Description and</b> <b>Valid Data Values</b></p>
<p>COMPOSITION_STATE</p>	<p>CompositionState</p>	<p>An indication of the operating condition of a mechanism represented by a Composition type element.</p> <p>Subtypes of CompositionState are ACTION, LATERAL, MOTION, SWITCHED, and VERTICAL.</p> <p>A subType <b>MUST</b> be provided.</p> <p><i>Valid Data Values</i> for subtype ACTION are:</p> <ul style="list-style-type: none"> <li>- ACTIVE: The Composition element is operating</li> <li>- INACTIVE: The Composition element is not operating</li> </ul> <p><i>Valid Data Values</i> for subtype LATERAL are:</p> <ul style="list-style-type: none"> <li>- RIGHT: The position of the Composition element is oriented to the right to the point of a positive confirmation</li> <li>- LEFT: The position of the Composition element is oriented to the left to the point of a positive confirmation</li> <li>- TRANSITIONING: The position of the Composition element is not oriented to the right to the point of a positive confirmation and is not oriented to the left to the point of a positive confirmation. It is in an intermediate position.</li> </ul> <p><i>Valid Data Values</i> for subtype MOTION are:</p> <ul style="list-style-type: none"> <li>- OPEN: The position of the Composition element is open to the point of a positive confirmation</li> <li>- CLOSED: The position of the Composition element is closed to the point of a positive confirmation</li> <li>- UNLATCHED: The position of the Composition element is not open to the point of a positive confirmation and is not closed to the point of a positive confirmation. It is in an intermediate position.</li> </ul>

<b>EVENT</b> <b>Data Item Type</b>	<b>Event</b> <b>Element Name</b>	<b>Description and</b> <b>Valid Data Values</b>
COMPOSITION_STATE (Continued)	CompositionState (Continued)	<p><i>Valid Data Values</i> for subtype SWITCHED are:</p> <ul style="list-style-type: none"> <li>- ON: The activation state of the Composition element is in an ON condition, it is operating, or it is powered.</li> <li>- OFF: The activation state of the Composition element is in an OFF condition, it is not operating, or it is not powered.</li> </ul> <p><i>Valid Data Values</i> for subtype VERTICAL are:</p> <ul style="list-style-type: none"> <li>- UP: The position of the Composition element is oriented in an upward direction to the point of a positive confirmation</li> <li>- DOWN: The position of the Composition element is oriented in a downward direction to the point of a positive confirmation</li> <li>- TRANSITIONING: The position of the Composition element is not oriented in an upward direction to the point of a positive confirmation and is not oriented in a downward direction to the point of a positive confirmation. It is in an intermediate position.</li> </ul>



<b>EVENT</b> <b>Data Item Type</b>	<b>Event</b> <i>Element Name</i>	<b>Description and</b> <i>Valid Data Values</i>
CONTROLLER_MODE	ControllerMode	<p>The current operating mode of the Controller component.</p> <p><i>Valid Data Values:</i></p> <ul style="list-style-type: none"> <li>- AUTOMATIC: The controller is configured to automatically execute a program.</li> <li>- MANUAL: The controller is not executing an active program. It is capable of receiving instructions from an external source – typically an operator. The controller executes operations based on the instructions received from the external source.</li> <li>- MANUAL_DATA_INPUT: The operator can enter a series of operations for the controller to perform. The controller will execute this specific series of operations and then stop.</li> <li>- SEMI_AUTOMATIC: The controller is operating in a single cycle mode. It executes a single set of instructions from an active program and then stops until given a command to execute the next set of instructions.</li> <li>- EDIT: The controller is currently functioning as a programming device and is not capable of executing an active program.</li> </ul>
CONTROLLER_MODE_OVERRIDE	ControllerModeOverride	<p>A setting or operator selection that changes the behavior of a piece of equipment.</p> <p>Subtypes of CompositionState are DRY_RUN, SINGLE_BLOCK, MACHINE_AXIS_LOCK, OPTIONAL_STOP, and TOOL_CHANGE_STOP.</p> <p>A subType <b>MUST</b> always be specified.</p> <p><i>Valid Data Values:</i></p> <ul style="list-style-type: none"> <li>- ON: The indicator of the ControllerModeOverride is in the ON state and the mode override is active.</li> <li>- OFF: The indicator of the ControllerModeOverride is in the OFF state and the mode override is inactive</li> </ul>

<b>EVENT</b> <b>Data Item Type</b>	<b>Event</b> <i>Element Name</i>	<b>Description and</b> <i>Valid Data Values</i>
COUPLED_AXES	CoupledAxes	<p>Refers to a set of associated axes.</p> <p>Used in conjunction with <i>AxisCoupling</i> to describe how the <i>CoupledAxes</i> relate to each other.</p> <p>The <i>Valid Data Value</i> reported <b>SHOULD</b> be a space-delimited set of axes names. The names returned <b>SHOULD</b> match the <i>name</i> attribute of the <i>Linear</i> or <i>Rotary Structural Elements</i> defined in the <i>MTConnectDevices</i> document that this Event element represents. If name is not available, <i>nativeName</i> <b>MUST</b> be returned to identify the <i>Linear</i> or <i>Rotary Structural Elements</i>.</p> <p>Example:  <code>&lt;CoupledAxes ...&gt;Y1 Y2&lt;/CoupledAxes&gt;</code></p>
DIRECTION	Direction	<p>The direction of motion.</p> <p>Subtypes of <i>Direction</i> are <i>ROTARY</i> and <i>LINEAR</i>.</p> <p>A <i>subType</i> <b>MUST</b> always be specified.</p> <p><i>Valid Data Values</i> for subtype <i>ROTARY</i> are:</p> <ul style="list-style-type: none"> <li>- <i>CLOCKWISE</i>: A <i>ROTARY</i> type component is rotating in a clockwise fashion using the right-hand rule.</li> <li>- <i>COUNTER_CLOCKWISE</i>: A <i>ROTARY</i> type component is rotating in a counter clockwise fashion using the right-hand rule.</li> </ul> <p><i>Valid Data Values</i> for subtype <i>LINEAR</i> are:</p> <ul style="list-style-type: none"> <li>- <i>POSITIVE</i>: A <i>LINEAR</i> type component is moving in the direction of increasing position value</li> <li>- <i>NEGATIVE</i>: A <i>LINEAR</i> type component is moving in the direction of decreasing position value</li> </ul>

<b>EVENT</b> <b>Data Item Type</b>	<b>Event</b> <b>Element Name</b>	<b>Description and</b> <b>Valid Data Values</b>
DOOR_STATE	DoorState	<p>The operational state of a DOOR type component or composition element.</p> <p><i>Valid Data Values:</i></p> <ul style="list-style-type: none"> <li>- OPEN: The Door is open to the point of a positive confirmation</li> <li>- CLOSED: The Door is closed to the point of a positive confirmation</li> <li>- UNLATCHED: The DOOR is not closed to the point of a positive confirmation and is not open to the point of a positive confirmation. It is in an intermediate position.</li> </ul>
END_OF_BAR	EndOfBar	<p>An indication of whether the end of a piece of bar stock being fed by a bar feeder has been reached.</p> <p>Subtypes of EndOfBar are PRIMARY and AUXILIARY .</p> <p>If a subType is not specified, the reported value for the data <b>MUST</b> default to the subType of PRIMARY.</p> <p><i>Valid Data Values:</i></p> <ul style="list-style-type: none"> <li>- YES: The EndOfBar has been reached.</li> <li>- NO: The EndOfBar has not been reached.</li> </ul>
EMERGENCY_STOP	EmergencyStop	<p>The current state of the emergency stop signal for a piece of equipment, controller path, or any other component or subsystem of a piece of equipment.</p> <p><i>Valid Data Values:</i></p> <ul style="list-style-type: none"> <li>- ARMED: The emergency stop circuit is complete and the piece of equipment, component, or composition element is allowed to operate.</li> <li>- TRIGGERED: The emergency stop circuit is open and the operation of the piece of equipment, component, or composition element is inhibited.</li> </ul>

<b>EVENT</b> <b>Data Item Type</b>	<b>Event</b> <i>Element Name</i>	<b>Description and</b> <i>Valid Data Values</i>
EQUIPMENT_MODE	EquipmentMode	<p>An indication that a piece of equipment, or a sub-part of a piece of equipment, is performing specific types of activities.</p> <p>Subtypes of EquipmentMode are LOADED, WORKING, OPERATING, and POWERED.</p> <p>A subType <b>MUST</b> always be specified.</p> <p><i>Valid Data Values:</i></p> <ul style="list-style-type: none"> <li>- ON: The equipment is functioning in the mode designated by the subType.</li> <li>- OFF: The equipment is not functioning in the mode designated by the subType.</li> </ul>

<b>EVENT</b> <b>Data Item Type</b>	<b>Event</b> <i>Element Name</i>	<b>Description and</b> <i>Valid Data Values</i>
EXECUTION	Execution	<p>The execution status of the Controller component.</p> <p><i>Valid Data Values:</i></p> <ul style="list-style-type: none"> <li>- READY: The controller is ready to execute instructions. It is currently idle.</li> <li>- ACTIVE: The controller is actively executing an instruction.</li> <li>- INTERRUPTED: The execution of the controller's program has been suspended due to an external signal. Action is required to resume execution.</li> <li>- FEED_HOLD: Motion of the device has been commanded to stop at its current position. The controller remains able to execute instructions but cannot complete the current set of instructions until after motion resumes. The command to stop the motion must be removed before execution can resume.</li> <li>- STOPPED: The execution of the controller's program has been stopped in an unplanned manner and execution of the program cannot be resumed without intervention by an operator or external signal.</li> <li>- OPTIONAL_STOP: The controller's program has been intentionally stopped using an M01 or similar command. The program may be stopped at the designated location based upon the state of a secondary indication provided to the controller indicating whether the program execution must be stopped at this location or program execution should continue.</li> <li>- PROGRAM_STOPPED: The execution of the controller's program has been stopped by a command from within the program. Action is required to resume execution.</li> <li>- PROGRAM_COMPLETED: The program has completed execution.</li> </ul>

<b>EVENT</b> <b>Data Item Type</b>	<b>Event</b> <i>Element Name</i>	<b>Description and</b> <i>Valid Data Values</i>
FUNCTIONAL_MODE	FunctionalMode	<p>The current intended production status or intended use of a piece of equipment or component.</p> <p>Typically, the FunctionalMode <b>SHOULD</b> be associated with the <i>Device Structural Element</i>, but it <b>MAY</b> be associated with any <i>Structural Element</i> in the XML document.</p> <p><i>Valid Data Values:</i></p> <ul style="list-style-type: none"> <li>- PRODUCTION: The <i>Device</i> element or another <i>Structural Element</i> is currently producing product, ready to produce product, or its current intended use is to be producing product.</li> <li>- SETUP: The <i>Device</i> element or another <i>Structural Element</i> is not currently producing product. It is being prepared or modified to begin production of product.</li> <li>- TEARDOWN: The <i>Device</i> element or another <i>Structural Element</i> is not currently producing product. Typically, it has completed the production of a product and is being modified or returned to a neutral state such that it may then be prepared to begin production of a different product.</li> <li>- MAINTENANCE: The <i>Device</i> element or another <i>Structural Element</i> is not currently producing product. It is currently being repaired, waiting to be repaired, or has not yet been returned to a normal production status after maintenance has been performed.</li> <li>- PROCESS_DEVELOPMENT: The <i>Device</i> element or another <i>Structural Element</i> is being used to prove-out a new process, testing of equipment or processes, or any other active use that does not result in the production of product.</li> </ul>
HARDNESS	Hardness	<p>The measurement of the hardness of a material.</p> <p>Subtypes of Hardness are ROCKWELL, VICKERS, SHORE, BRINELL, LEEB, and MOHS.</p> <p>A subType <b>MUST</b> always be specified.</p> <p>The <i>Valid Data Value</i> <b>MUST</b> be a floating-point number.</p>

<b>EVENT</b> <b>Data Item Type</b>	<b>Event</b> <b>Element Name</b>	<b>Description and</b> <b>Valid Data Values</b>
<del>LINE</del>	<del>Line</del>	<b>DEPRECATED</b> in Version 1.4.0.
LINE_LABEL	LineLabel	An optional identifier for a BLOCK of code in a PROGRAM. The <i>Valid Data Value</i> <b>MUST</b> be any text string.
LINE_NUMBER	LineNumber	A reference to the position of a block of program code within a control program. Subtypes of LineNumber are ABSOLUTE and INCREMENTAL. A subType <b>MUST</b> always be specified. The <i>Valid Data Value</i> <b>MUST</b> be an integer.
MATERIAL	Material	The identifier of a material used or consumed in the manufacturing process. The <i>Valid Data Value</i> <b>MUST</b> be any text string.
MESSAGE	Message	Any text string of information to be transferred from a piece of equipment to a client software application. The <i>Valid Data Value</i> <b>MUST</b> be any text string.
OPERATOR_ID	OperatorId	The identifier of the person currently responsible for operating the piece of equipment. The <i>Valid Data Value</i> <b>MAY</b> be any text string. <b>DEPRECATION WARNING:</b> May be deprecated in the future. See USER below.

EVENT Data Item Type	Event <i>Element Name</i>	Description and <i>Valid Data Values</i>
PALLET_ID	PalletId	<p>The identifier for a pallet.</p> <p>The <i>Valid Data Value</i> <b>MAY</b> be any text string.</p>
PART_COUNT	PartCount	<p>The current count of parts produced as represented by the Controller component.</p> <p>Subtypes of PartCount are ALL, GOOD, BAD, TARGET, and REMAINING .</p> <p>PartCount will not be accumulated by an <i>MTConnect Agent</i> and <b>MUST</b> only be supplied if the Controller provides the count.</p> <p>PartCount <b>MAY</b> have a representation of DISCRETE. In this case, each occurrence of PartCount in an <i>MTConnectStreams</i> document represents a unique count of parts or product produced – it is not an accumulated count of parts or product produced.</p> <p>The <i>Valid Data Value</i> <b>MUST</b> be a floating-point number, usually an integer.</p>



<b>EVENT</b> <b>Data Item Type</b>	<b>Event</b> <i>Element Name</i>	<b>Description and</b> <i>Valid Data Values</i>
PART_ID	PartId	An identifier of a part in a manufacturing operation. The <i>Valid Data Value</i> <b>MAY</b> be any text string.
PATH_FEEDRATE_ OVERRIDE	PathFeedrateOverride	The value of a signal or calculation issued to adjust the feedrate for the axes associated with a Path component that may represent a single axis or the coordinated movement of multiple axes.  The value provided for PathFeedrateOverride is expressed as a percentage of the designated feedrate for the path.  Sub-types of PathFeedrateOverride are JOG, PROGRAMMED, and RAPID.  If a subType is not specified, the reported value for the data <b>MUST</b> default to the subtype of PROGRAMMED.  The <i>Valid Data Value</i> <b>MUST</b> be a floating-point number.

<p><b>EVENT</b> <b>Data Item Type</b></p>	<p><b>Event</b> <b><i>Element Name</i></b></p>	<p><b>Description and</b> <b><i>Valid Data Values</i></b></p>
<p>PATH_MODE</p>	<p>PathMode</p>	<p>Describes the operational relationship between a PATH <i>Structural Element</i> and another PATH <i>Structural Element</i> for pieces of equipment comprised of multiple logical groupings of controlled axes or other logical operations.</p> <p><i>Valid Data Values:</i></p> <ul style="list-style-type: none"> <li>- INDEPENDENT: The path is operating independently and without the influence of another path.</li> <li>- MASTER: The path provides the reference motion for a SYNCHRONOUS or MIRROR type path to follow. For non-motion type paths, the MASTER provides information or state values that influences the operation of other paths</li> <li>- SYNCHRONOUS: The axes associated with the path are following the motion of the MASTER type path.</li> <li>- MIRROR: The axes associated with the path are mirroring the motion of the MASTER path.</li> </ul> <p>When PathMode is not specified, the operational mode of the path <b>MUST</b> be interpreted as INDEPENDENT.</p>

<p><b>EVENT</b> <b>Data Item Type</b></p>	<p><b>Event</b> <b><i>Element Name</i></b></p>	<p><b>Description and</b> <b><i>Valid Data Values</i></b></p>
<p>POWER_STATE</p>	<p>PowerState</p>	<p>The indication of the status of the source of energy for a <i>Structural Element</i> to allow it to perform its intended function or the state of an enabling signal providing permission for the <i>Structural Element</i> to perform its functions.</p> <p>Subtypes of PowerState are LINE and CONTROL.</p> <p>When the subType is LINE, PowerState represents the primary source of energy for a <i>Structural Element</i>.</p> <p>When the subType is CONTROL, PowerState represents an enabling signal providing permission for the <i>Structural Element</i> to perform its function(s).</p> <p>If a subType is not specified, the reported value for the data <b>MUST</b> default to the subtype of LINE.</p> <p><i>Valid Data Values:</i></p> <ul style="list-style-type: none"> <li>- ON: The source of energy for a <i>Structural Element</i> or the enabling signal providing permission for the <i>Structural Element</i> to perform its function(s) is present and active.</li> <li>- OFF: The source of energy for a <i>Structural Element</i> or the enabling signal providing permission for the <i>Structural Element</i> to perform its function(s) is not present or is disconnected.</li> </ul> <p><b>DEPRECATION WARNING:</b> PowerState may be deprecated in the future.</p>

<b>EVENT</b> <b>Data Item Type</b>	<b>Event</b> <i>Element Name</i>	<b>Description and</b> <i>Valid Data Values</i>
<del>POWER_STATUS</del>	<del>PowerStatus</del>	<b>DEPRECATED</b> in <i>Version 1.1.0</i> .
PROGRAM	Program	The name of the logic or motion program being executed by the <i>Controller</i> component. This is usually the name of the file containing the program instructions. The <i>Valid Data Value</i> <b>MUST</b> be any text string.

<b>EVENT</b> <b>Data Item Type</b>	<b>Event</b> <i>Element Name</i>	<b>Description and</b> <i>Valid Data Values</i>
PROGRAM_EDIT	ProgramEdit	<p>An indication of the status of the Controller component's program editing mode.</p> <p>On many controls, a program can be edited while another program is currently being executed.</p> <p>ProgramEdit provides an indication of whether the controller is being used to edit programs in either case.</p> <p><i>Valid Data Values:</i></p> <ul style="list-style-type: none"> <li>- ACTIVE: The controller is in the program edit mode.</li> <li>- READY: The controller is capable of entering the program edit mode and no function is inhibiting a change to that mode.</li> <li>- NOT_READY: A function is inhibiting the controller from entering the program edit mode.</li> </ul>
PROGRAM_EDIT_NAME	ProgramEditName	<p>The name of the program being edited.</p> <p>This is used in conjunction with PROGRAM_EDIT when it is in an ACTIVE state.</p> <p>The <i>Valid Data Value</i> <b>MUST</b> be any text string.</p>
PROGRAM_COMMENT	ProgramComment	<p>A comment or non-executable statement in the control program.</p> <p>The <i>Valid Data Value</i> <b>MUST</b> be any text string.</p>
PROGRAM_HEADER	ProgramHeader	<p>The non-executable header section of the control program.</p> <p>The content <b>SHOULD</b> be limited to 512 bytes.</p> <p>The <i>Valid Data Value</i> <b>MUST</b> be any text string.</p>

<b>EVENT</b> <b>Data Item Type</b>	<b>Event</b> <b>Element Name</b>	<b>Description and</b> <b>Valid Data Values</b>
ROTARY_MODE	RotaryMode	<p>The current operating mode for a Rotary type axis.</p> <p><i>Valid Data Values:</i></p> <ul style="list-style-type: none"> <li>- SPINDLE: The axis is functioning as a spindle. Generally, it is configured to rotate at a defined speed.</li> <li>- INDEX: The axis is configured to index to a set of fixed positions or to incrementally index by a fixed amount.</li> <li>- CONTOUR: The position of the axis is being interpolated as part of the PathPosition defined by the Controller Structural Element.</li> </ul>
ROTARY_VELOCITY_OVERRIDE	RotaryVelocityOverride	<p>The value of a command issued to adjust the programmed velocity for a Rotary type axis.</p> <p>This command represents a percentage change to the velocity calculated by a logic or motion program or set by a switch for a Rotary type axis.</p> <p>RotaryVelocityOverride is expressed as a percentage of the programmed RotaryVelocity.</p> <p>The <i>Valid Data Value</i> <b>MUST</b> be a floating-point number.</p>
SERIAL_NUMBER	SerialNumber	<p>The serial number associated with a Component, Asset, or Device.</p> <p>The <i>Valid Data Value</i> <b>MUST</b> be any text string.</p>
SPINDLE_INTERLOCK	SpindleInterlock	<p>An indication of the status of the spindle for a piece of equipment when power has been removed and it is free to rotate.</p> <p><i>Valid Data Values:</i></p> <ul style="list-style-type: none"> <li>- ACTIVE: Power has been removed and the spindle cannot be operated.</li> <li>- INACTIVE: Spindle has not been deactivated.</li> </ul>

EVENT Data Item Type	Event <i>Element Name</i>	Description and <i>Valid Data Values</i>
TOOL_ID	<del>ToolID</del>	<b>DEPRECATED</b> in Version 1.2.0. See Tool_ASSET_ID. <del>The identifier of the tool currently in use for a given Path</del>
TOOL_ASSET_ID	ToolAssetId	The unique identifier of an individual tool asset. The <i>Valid Data Value</i> <b>MUST</b> be any text string.
TOOL_NUMBER	ToolNumber	The identifier assigned by the Controller component to a cutting tool when in use by a piece of equipment. The <i>Valid Data Value</i> <b>MUST</b> be any text string.
TOOL_OFFSET	ToolOffset	A reference to the tool offset variables applied to the active cutting tool associated with a Path in a Controller type component. Subtypes of ToolOffset are RADIAL and LENGTH. A subType <b>MUST</b> always be specified. The <i>Valid Data Value</i> <b>MUST</b> be a floating-point number.
USER	User	The identifier of the person currently responsible for operating the piece of equipment. Subtypes of User are OPERATOR, MAINTENANCE, and SET_UP. A subType <b>MUST</b> always be specified. The <i>Valid Data Value</i> <b>MUST</b> be any text string.
WIRE	Wire	The identifier for the type of wire used as the cutting mechanism in Electrical Discharge Machining or similar processes. The <i>Valid Data Value</i> <b>MUST</b> be any text string.

EVENT Data Item Type	Event <i>Element Name</i>	Description and <i>Valid Data Values</i>
WORKHOLDING_ID	WorkholdingId	<p>The identifier for the current workholding or part clamp in use by a piece of equipment.</p> <p>The <i>Valid Data Value</i> <b>MUST</b> be any text string.</p>
WORK_OFFSET	WorkOffset	<p>A reference to the offset variables for a work piece or part associated with a Path in a Controller type component.</p> <p>The Valid Data Value <b>MUST</b> be a floating-point number.</p>

813

814

Note: The Event response format **MUST** be extended to represent those data items where the representation attribute is DISCRETE. See *Section 5.5.3* of this document for details on extending the response format.

815

816



817 **6.3 Types of Condition Elements**

818 As described above in Section 5.7, *Condition Data Entities* are reported differently from  
 819 other data item types. They are reported based on the *Fault State* for each *Condition*.  
 820 Unlike *Sample* and *Event* data items that are identified by their *Element Name*, *Condition*  
 821 data items are defined by the *type* and *subType* (where applicable) attributes defined for each  
 822 *Condition*.

823 The *type* and *subType* (where applicable) attributes for a *Condition* element **MAY** be any  
 824 of the *type* and *subType* attributes defined for *SAMPLE* category or *EVENT* category data  
 825 item listed in the *Device Information Model*.

826 The following table lists additional *Condition Data Entities* that have been defined to  
 827 represent the health and fault status of *Structural Elements*. The table defines the *type* attribute  
 828 for each of these additional *Condition* category elements that **MAY** be reported in the  
 829 *MTConnectStreams* document.

830

CONDITION Data Item Type	Description
ACTUATOR	An indication of a fault associated with an actuator.
CHUCK_INTERLOCK	An indication of the operational condition of the interlock function for an electronically controller chuck.
COMMUNICATIONS	An indication that the piece of equipment has experienced a communications failure.
DATA_RANGE	An indication that the value of the data associated with a measured value or a calculation is outside of an expected range.
DIRECTION	An indication of a fault associated with the direction of motion of a <i>Structural Element</i> .
END_OF_BAR	An indication that the end of a piece of bar stock has been reached.
HARDWARE	An indication of a fault associated with the hardware subsystem of the <i>Structural Element</i> .
INTERFACE_STATE	An indication of the operational condition of an <i>Interface</i> component.
LOGIC_PROGRAM	An indication that an error occurred in the logic program or programmable logic controller (PLC) associated with a piece of equipment.
MOTION_PROGRAM	An indication that an error occurred in the motion program associated with a piece of equipment

<b>CONDITION</b> <b>Data Item Type</b>	<b>Description</b>
SYSTEM	A general purpose indication associated with an electronic component of a piece of equipment or a controller that represents a fault that is not associated with the operator, program, or hardware.

831

## Appendices

832

### A. Bibliography

- 833 1. Engineering Industries Association. *EIA Standard - EIA-274-D*, Interchangeable Variable,  
834 Block Data Format for Positioning, Contouring, and Contouring/Positioning Numerically  
835 Controlled Machines. Washington, D.C. 1979.
- 836 2. ISO TC 184/SC4/WG3 N1089. *ISO/DIS 10303-238*: Industrial automation systems and  
837 integration Product data representation and exchange Part 238: Application Protocols:  
838 Application interpreted model for computerized numerical controllers. Geneva,  
839 Switzerland, 2004.
- 840 3. International Organization for Standardization. *ISO 14649*: Industrial automation systems  
841 and integration – Physical device control – Data model for computerized numerical  
842 controllers – Part 10: General process data. Geneva, Switzerland, 2004.
- 843 4. International Organization for Standardization. *ISO 14649*: Industrial automation systems  
844 and integration – Physical device control – Data model for computerized numerical  
845 controllers – Part 11: Process data for milling. Geneva, Switzerland, 2000.
- 846 5. International Organization for Standardization. *ISO 6983/1* – Numerical Control of  
847 machines – Program format and definition of address words – Part 1: Data format for  
848 positioning, line and contouring control systems. Geneva, Switzerland, 1982.
- 849 6. Electronic Industries Association. *ANSI/EIA-494-B-1992*, 32 Bit Binary CL (BCL) and 7  
850 Bit ASCII CL (ACL) Exchange Input Format for Numerically Controlled Machines.  
851 Washington, D.C. 1992.
- 852 7. National Aerospace Standard. *Uniform Cutting Tests* - NAS Series: Metal Cutting  
853 Equipment Specifications. Washington, D.C. 1969.
- 854 8. International Organization for Standardization. *ISO 10303-11*: 1994, Industrial  
855 automation systems and integration product data representation and exchange Part 11:  
856 Description methods: The EXPRESS language reference manual. Geneva, Switzerland,  
857 1994.
- 858 9. International Organization for Standardization. *ISO 10303-21*: 1996, Industrial  
859 automation systems and integration -- Product data representation and exchange -- Part  
860 21: Implementation methods: Clear text encoding of the exchange structure. Geneva,  
861 Switzerland, 1996.
- 862 10. H.L. Horton, F.D. Jones, and E. Oberg. *Machinery's handbook*. Industrial Press, Inc. New  
863 York, 1984.
- 864 11. International Organization for Standardization. *ISO 841-2001: Industrial automation  
865 systems and integration - Numerical control of machines - Coordinate systems and  
866 motion nomenclature*. Geneva, Switzerland, 2001.

867 12. ASME B5.57: *Methods for Performance Evaluation of Computer Numerically Controlled*  
868 *Lathes and Turning Centers*, 1998

869 13. ASME/ANSI B5.54: *Methods for Performance Evaluation of Computer Numerically*  
870 *Controlled Machining Centers*. 2005.

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

873 15. IEEE STD 1451.0-2007, *Standard for a Smart Transducer Interface for Sensors and*  
874 *Actuators – Common Functions, Communication Protocols, and Transducer Electronic*  
875 *Data Sheet (TEDS) Formats*, IEEE Instrumentation and Measurement Society, TC-9, The  
876 *Institute of Electrical and Electronics Engineers, Inc., New York, N.Y. 10016, SH99684,*  
877 *October 5, 2007.*

878 16. IEEE STD 1451.4-1994, *Standard for a Smart Transducer Interface for Sensors and*  
879 *Actuators – Mixed-Mode Communication Protocols and Transducer Electronic Data*  
880 *Sheet (TEDS) Formats*, IEEE Instrumentation and Measurement Society, TC-9, The  
881 *Institute of Electrical and Electronics Engineers, Inc., New York, N.Y. 10016, SH95225,*  
882 *December 15, 2004.*

883