



MTConnect[®] Standard

Part 3 – Streams Information Model

Version 1.3.1

Prepared for: MTConnect Institute
Prepared by: John Turner
Prepared on: June 8, 2015

MTConnect[®] Specification and Materials

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

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

1	Overview	1
2	Terminology	2
3	Streams Information Model	3
4	Streams Data Structure	4
4.1	Streams	6
4.2	DeviceStream.....	7
4.2.1	<i>attributes for DeviceStream</i>	8
4.2.2	<i>ComponentStream Element</i>	8
4.3	ComponentStream	9
4.3.1	<i>ComponentStream Attributes</i>	10
4.3.2	<i>ComponentStream Elements</i>	10
5	Data Items	12
5.1	Transformation of Data Item Names	13
5.2	Returned Data Values	15
5.3	Samples XML Element	18
5.4	Sample XML Element.....	19
5.4.1	<i>XML Schema Structure for Sample</i>	19
5.4.2	<i>Attributes for Sample</i>	20
5.4.3	<i>Sample XML Element Tag Names</i>	22
5.5	Events XML Elements	27
5.6	Event XML Elements.....	27
5.6.1	<i>XML Schema Structure for Event</i>	28
5.6.2	<i>Attributes for Event</i>	28
5.6.3	<i>Event XML Element Tag Names</i>	29
5.6.4	<i>EVENT Category DataItem Types Specific for Interface</i>	42
5.7	Condition XML Elements.....	43
5.8	Condition Type Data Elements	44
5.8.1	<i>XML Schema Structure for Condition</i>	45
5.8.2	<i>Condition Attributes</i>	46
5.8.3	<i>Condition Type Data Elements</i>	47
	Appendices	49
A.	Bibliography	49

Table of Figures

Figure 1: Streams Data Structure 4
Figure 2: Streams Schema Diagram..... 6
Figure 3: DeviceStream Schema Diagram..... 7
Figure 4: ComponentStream Schema Diagram 9
Figure 5: ComponentStream XML Tree Diagram..... 12
Figure 6: Sample Schema Diagram..... 19
Figure 7: Event Schema Diagram 28
Figure 8: Condition Schema Diagram..... 45

1 Overview

2 This document, *Part 3 Streams Information Model* of the MTConnect[®] Standard, defines the
3 rules and terminology for the information that is returned by an MTConnect Agent from a device
4 in response to a `Sample` or `Current` request.

5 In the MTConnect Standard, a device typically represents a single piece of equipment (i.e.
6 machine, robot, etc.). It can also represent any logical grouping of pieces of equipment that
7 operate together to perform a function.

8 **2 Terminology**

9 Refer to *Section 2 of Part 1, Overview and Protocol*, for a dictionary of terms used in the
10 MTConnect Standard.

11 **3 Streams Information Model**

12 A MTConnect Agent responds to a `Current` or `Sample` request with a
13 `MTConnectStreams` XML document.

14 The `MTConnectStreams` XML document is comprised of two sections – `Header` and
15 `Streams`.

16 `Header` is defined in *Section 4.2 of Part 1 Overview and Protocol* of the MTConnect Standard.

17 `Streams` is defined by the Streams Information Model. The Streams Information Model is a
18 XML data model that describes the data provided from a device and associated information that
19 allows a client software application to understand that data. A client software application
20 correlates the information provided in `Streams` with the data structure defined by the Device
21 Information Model (*Part 2 of the MTConnect Standard*) for a specific device to form a clear and
22 unambiguous understanding of the information provided.

23 In the Streams Information Model, data provided for a device is organized into three types of
24 XML container type elements - `Samples`, `Events`, and `Condition`.

25 `Samples` provides the value(s) of `SAMPLE` type data items (See *Part 2, Section 7.1*) at a
26 specific point in time.

27 `Events` provides the value(s) of `EVENT` type data items (See *Part 2, Section 7.2*) at a specific
28 point in time.

29 `Condition` provides the value(s) of `CONDITION` type data items (See *Part 2, Section 7.3*) at a
30 specific point in time.

31

32

33 Note: The MTConnect Standard also defines the information model for `Assets`. An `Asset` is
34 something that is associated with the manufacturing process that is not a component of a device,
35 can be removed without detriment to the function of the device, and can be associated with other
36 devices during their lifecycle. See *Part 4, Assets*, of the MTConnect Standard, for more details
37 on `Assets`.

38

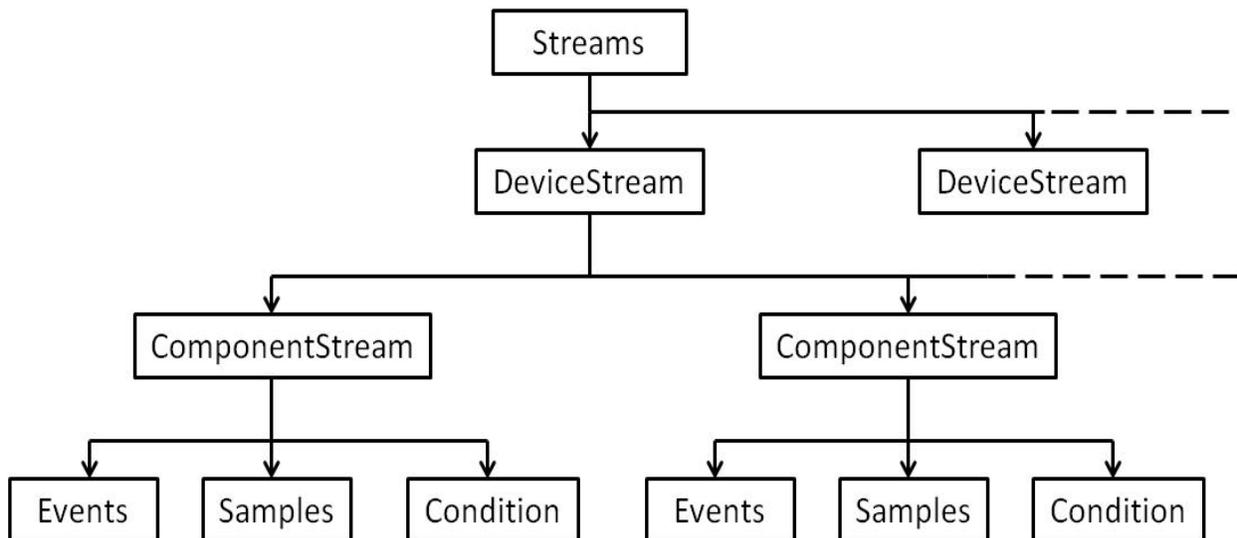
39 4 Streams Data Structure

40 Streams is one of the two top level XML elements in the MTConnectStreams XML
41 document. Streams is a container type XML element. Its function is to organize
42 DeviceStream elements. There **MUST** be one or more DeviceStream element(s), each
43 representing an individual device, in the Streams container. Streams **MUST NOT** contain
44 any other type XML elements or attributes.

45 DeviceStream is the next level XML element in the document. It is a XML container type
46 element. A DeviceStream contains all of the information for a specific device. Its attributes
47 uniquely identify the specific device by providing the name of the device and its UUID. Once
48 this information is defined, all data for a device can be associated with this identity and it does
49 not need to be repeated for every piece of data provided in the document. DeviceStream
50 **MAY** also contain one or more ComponentStream XML elements which provide the actual
51 data values provided from a device.

52 ComponentStream is the next level XML element in the document. It too is an XML
53 container type element. There **WILL** be a separate ComponentStream XML element for
54 each of a device's Structural Elements (Device, Component type or Subcomponent type)
55 as defined in the Device Information Model (*See Part 2*) for which data is provided. The
56 ComponentStream element groups the data provided for each type of data item into
57 individual container type XML elements – one each for Samples, Events, and Condition;
58 as required.

59 The tree structure below illustrates the data structure of the Streams Information Model.
60
61



62
63

64

65

Figure 1: Streams Data Structure

66 Below is a sample from an MTConnectStreams XML document that contains the response
67 from a MTConnect Agent representing two devices, mill-1 and mill-2. The data from each
68 device is reported in separate DeviceStream containers.

```
69 <MTConnectStreams ...>
70   <Header ... />
71   <Streams>
72     <DeviceStream name="mill-1" uuid="1">
73       <ComponentStream component="Device" name="mill-1" componentId="d1">
74         <Events>
75           <Availability dataItemId="avail1" name="avail" sequence="5"
76             timestamp="2010-04-06T06:19:35.153141">AVAILABLE</Availability>
77         </Events>
78       </ComponentStream>
79     </DeviceStream>
80     <DeviceStream name="mill-2" uuid="2">
81       <ComponentStream component="Device" name="mill-2" componentId="d2">
82         <Events>
83           <Availability dataItemId="avail2" name="avail" sequence="15"
84             timestamp="2010-04-06T06:19:35.153141">AVAILABLE</Availability>
85         </Events>
86       </ComponentStream>
87     </DeviceStream>
88   </Streams>
89 </MTConnectStreams>
90
```

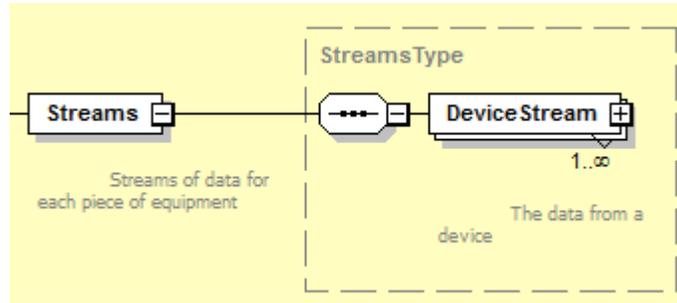
91 *In the example above, it should be noted that the sequence numbers are unique across the two*
92 *devices. Client software applications **MUST NOT** assume that the Events and Samples*
93 *sequence numbers are strictly in sequence. All sequence numbers **MAY NOT** be included. For*
94 *instance, such a case would occur when a Path argument is provided and the non-Path*
95 *SAMPLE, EVENT, and CONDITION data types are not returned, or when the Agent is*
96 *supporting more than one device and data from only one device is requested. Refer to*
97 *MTConnect[®] Part 1, Overview and Protocol, Section 5: Protocol for more information.*

98

99

100 **4.1 Streams**

101 The following XML tree represents the structure of Streams. Streams is a XML container
 102 that provides one or more DeviceStream XML elements.



103

104

105

106

Figure 2: Streams Schema Diagram

Elements	Description	Occurrence
Streams	A top level XML container element in a MTConnectStreams XML document provided by a MTConnect Agent in response to a Sample or Current request. Streams is contained within the top level MTConnectStreams container. There MAY be only one Streams element in a MTConnectStreams XML document. The Streams element contains one, or more, DeviceStream elements.	1

107

108 Streams **MUST** have at least one DeviceStream XML element. It **MAY** have more than
 109 one DeviceStream element – one for the data provided for each device described in the
 110 document.

111

112

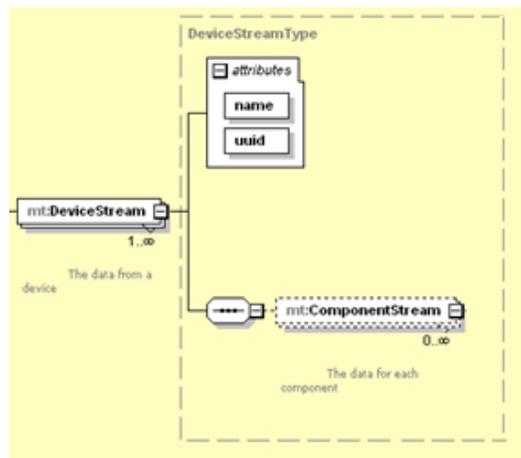
113 **4.2 DeviceStream**

114 DeviceStream is a XML container that provides data from a device and the information that
 115 identifies the specific device for which that data is associated. If data is provided for a device, it
 116 **MUST** be organized in a lower level ComponentStream XML element.

117 A DeviceStream **MAY** contain one or more ComponentStream element(s). However, if
 118 the request is valid and there are no SAMPLE, EVENT, or CONDITION data types that match
 119 the request criteria, an empty DeviceStream element **MUST** be created to indicate that the
 120 device exists, but there was no data available. In this case, there will be no
 121 ComponentStream element provided.

122 The following XML tree represents the structure of the DeviceStream XML element showing
 123 the attributes defined for the DeviceStream and the ComponentStream element(s) that
 124 may be associated with the DeviceStream.

125



126

127

Figure 3: DeviceStream Schema Diagram

128

Elements	Description	Occurrence
DeviceStream	A XML container element provided in each MTConnectStreams XML document provided by a MTConnect Agent in response to a Sample or Current request. DeviceStream is contained within the higher level Streams container. There MAY be one or more DeviceStream elements in a MTConnectStreams XML document – one for each device represented in the document. DeviceStream MAY contain one or more ComponentStream XML elements – one for each of a device’s Structural Elements (Device, Component type or Subcomponent type) as defined in the Device Information Model (See Part 2) for which data is provided.	1..INF

129

130 4.2.1 **attributes for DeviceStream**

131 The following table defines the **attributes** that **MUST** be provided to uniquely identify the
 132 specific device for which the data in the document applies.

133

Attributes	Description	Occurrence
name	name attribute of the Device defined in the Device Information Model (<i>See Part 2</i>) for which data is provided. An NMTOKEN XML type.	1
uuid	uuid attribute of the Device defined in the Device Information Model (<i>See Part 2</i>) for which data is provided.	1

134

135 4.2.2 **ComponentStream Element**

136 The following table defines the ComponentStream XML element that **MAY** be provided in
 137 the DeviceStream. The ComponentStream element contains the data associated with a
 138 specific Structural Element (Device, Component type or Subcomponent type) of a device
 139 that is provided by a MTConnect Agent in response to a Current or Sample request.

140

Element	Description	Occurrence
ComponentStream	A XML container type element that may be provided in a MTConnectStreams XML document returned from a MTConnect Agent in response to a Current or Sample request that organizes data for a specific Structural Element of a device. ComponentStream is contained within the higher level DeviceStream container. There MAY be one or more ComponentStream element(s) in a DeviceStream container – one for each Structural Element (Device, Component type or Subcomponent type) of a device for which data is provided. ComponentStream groups individual pieces of data into separate container type XML elements – one for each type of data item (SAMPLE, EVENT, and CONDITION) provided.	0..INF

141

142

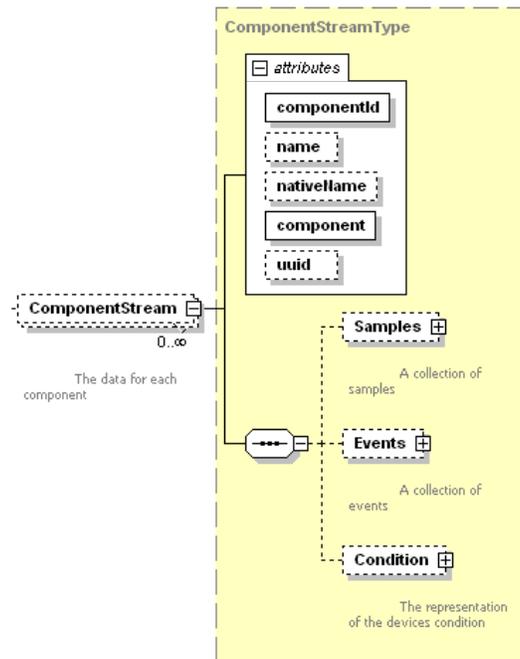
143

144 **4.3 ComponentStream**

145 ComponentStream is a XML container that organizes the data provided for a Structural
146 Element (Device, Component type or Subcomponent type) of a device in a
147 MTConnectStreams document. The MTConnect Agent organizes data provided for each
148 Structural Element of the device into individual ComponentStream container elements. The
149 data in each ComponentStream element is then organized into individual XML container
150 elements for each type of data item provided– one each for SAMPLE, EVENT, and CONDITION
151 type; as required.

152 There **MUST** be a separate ComponentStream element in a DeviceStream for every
153 Structural Element of the device for which data is provided.

154
155 The following XML tree represents the structure for the ComponentStream XML element
156 showing the attributes defined for the ComponentStream and the individual container type
157 XML elements provided for each type of data item contained in the document.



158
159 **Figure 4: ComponentStream Schema Diagram**
160

161 ComponentStream is similar to the DeviceStream in that the attributes uniquely
162 identify the data provided such that this information does not have to be repeated for each piece
163 of data. In the case of the DeviceStream, the attributes uniquely identify the device
164 associated with the data. In the case of the ComponentStream, the attributes identify
165 the specific Structural Element within a device associated with each piece of data.

166 4.3.1 ComponentStream Attributes

167 The following table defines the attributes that **MAY** be provided to uniquely identify the
168 specific Structural Element of a device (Device, Component type or Subcomponent type)
169 for which the data in the document applies.

170

Attribute	Description	Occurrence
name	name attribute of the Structural Element* (Device, Component type or Subcomponent type) of a device for which data is provided. An NMOKEN XML type.	0..1
nativeName	nativeName attribute of the Structural Element* (Device, Component type or Subcomponent type) of a device for which data is provided.	0..1
component	The type of Structural Element* (Device, Component type or Subcomponent type) for which data is provided. An NMOKEN XML type.	1
uuid	uuid attribute (unique identifier) of the Structural Element* (Device, Component type or Subcomponent type) of a device for which data is provided.	0..1
componentId	id attribute of a device's Structural Element* (Device, Component type or Subcomponent type) for which data is provided.	1

171

172 **Note: * Structural Elements for a device are defined in the Device Information**
173 **Model. See Part 2 for details on Structural Elements.**

174

175 4.3.2 ComponentStream Elements

176 In the ComponentStream container, a MTConnect Agent organizes the data returned in the
177 MTConnectStreams XML document into individual XML container type elements for each
178 type of data item provided – one each for SAMPLE, EVENT, and CONDITION type; as required.

179 The ComponentStream element **MUST NOT** be empty. It **MUST** include at least one
180 Events, Samples, or Condition XML container element.

181

182 The type of data provided in each of the `ComponentStream` sub-elements is defined in the
183 table below.

184

Element	Description	Occurrence
Samples	A XML container type element. <code>Samples</code> organizes the data returned in the <code>MTConnectStreams</code> XML document in response to a <code>Current</code> or <code>Sample</code> request for the <code>SAMPLE</code> type data items for a specific <code>StructuralElement</code> of a device.	0..1 *
Events	A XML container type element. <code>Events</code> organizes the data returned in the <code>MTConnectStreams</code> XML document in response to a <code>Current</code> or <code>Sample</code> request for the <code>EVENT</code> type data items for a specific <code>StructuralElement</code> of a device.	0..1 *
Condition	A XML container type element. <code>Condition</code> organizes the data returned in the <code>MTConnectStreams</code> XML document in response to a <code>Current</code> or <code>Sample</code> request for the <code>CONDITION</code> type data items for a specific <code>StructuralElement</code> of a device.	0..1 *

185 Note: * The `ComponentStream` element **MUST** contain at least one of these element types.
186

187 **5 Data Items**

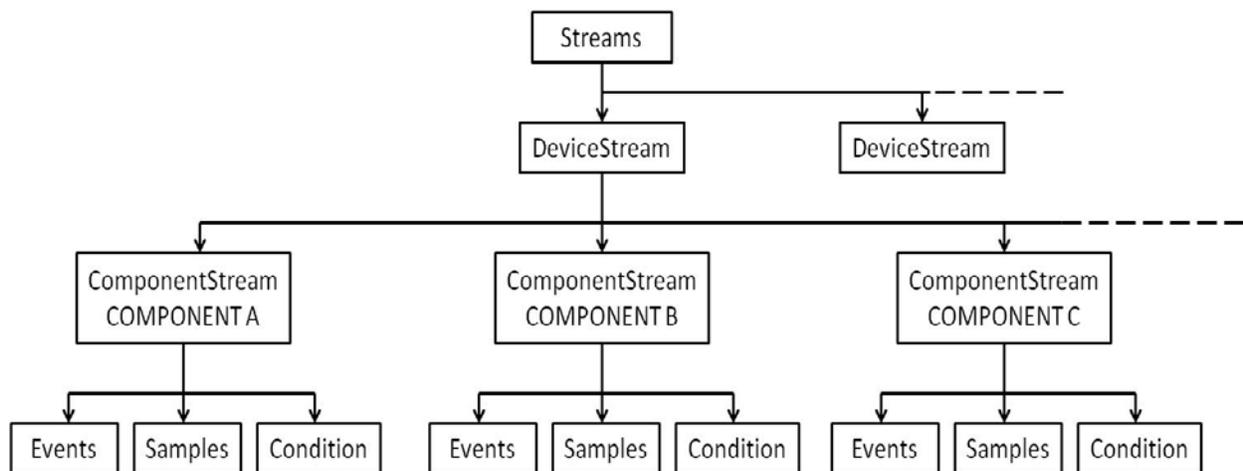
188 In the Device Information Model, `DataItem` is defined as a XML Element that describes data
189 that can be collected from a device and is associated with a `Device`, `Component`, or
190 `Subcomponent` Structural Element (See *Part 2, Section 6.2*).

191 The Streams Information Model defines how the data associated with a `DataItem` is structured
192 in a `MTConnectStreams` XML document.

193 As defined in the Device Information Model, there are three types of data items - `SAMPLE`,
194 `EVENT`, and `CONDITION` types. When a `MTConnect Agent` responds to a `Current` request,
195 the data returned in the `MTConnectStreams` document **MUST** include the most current value
196 for every `DataItem` defined in the Device Information Model; subject to any filtering included
197 within the request. When a `MTConnect Agent` responds to a `Sample` request, the data returned
198 in the `MTConnectStreams` document **MUST** include all of the occurrences of each
199 `DataItem` XML Element that are available to the Agent; subject to any filtering included
200 within the request. It is recommended that the request include filtering to control the amount of
201 data transmitted in the `MTConnectStreams` document (See *Part 1, Section 5.3* for more
202 details on setting filter criteria for data requests to a `MTConnect Agent`). `DataItem` XML
203 Elements are organized first by each Structural Element in the Device Information Model and
204 then into individual XML containers – one for each type of `DataItem` XML Element
205 (`SAMPLE`, `EVENT`, and `CONDITION`).

206 See the XML tree below for more details on how this data is organized.

207



208

209

210

Figure 5: ComponentStream XML Tree Diagram

211 The resulting Streams XML document will be structured as follows:

```
212     <DeviceStream>
213         <ComponentStream Component A>
214             <Samples>
215                 <Sample type 1>
216                 <Sample type 2>
217                 <Sample type 3>
218             </Samples>
219             <Events>
220                 <Event type 1>
221                 <Event type 2>
222             </Events>
223             </Condition>
224                 <Condition type 1>
225                 <Condition type 2>
226             </Condition>
227         </ComponentStream>
228         <ComponentStream Component B>
229             <Samples>
230                 <Sample type 1>
231                 <Sample type 2>
232             </Samples>
233             <Events>
234                 <Event type 1>
235                 <Event type 2>
236                 <Event type 3>
237             </Events>
238             </Condition>
239                 <Condition type 1>
240             </Condition>
241         </ComponentStream>
242     </DeviceStream>
```

244 (Note: The highlighted items are identified for emphasis only)

245
246 Note: There are no specific requirements defining the sequence in which the
247 ComponentStream XML Elements are organized in the MTConnectStreams
248 document. They **MAY** be organized in any sequence subject to the implementation of the
249 Agent and do not impact the ability for a client software application to receive and
250 interpret the information in the document.

252 5.1 Transformation of Data Item Names

253 MTConnect XML documents **MUST** adhere to industry standard XML Markup Conventions for
254 formatting all XML Elements and their associated attributes and data contained in those
255 documents.

256

257 For SAMPLE and EVENT type data items with a representation attribute defined in the
258 Device Information Model as VALUE or the representation attribute is not specified, the
259 tag names for these Data Elements **MUST** be transformed into Pascal case format (first letter of
260 each word is capitalized) in a MTConnectStreams document.

261 (Note: The highlighted items are identified for emphasis only)

262 For example: The Data Element CONTROLLER_MODE from the Device Information Model is
263 transformed into ControllerMode in the MTConnectStreams document.

264 Sections 5.4.3 and 6.4.3 of this document detail this transformation for each of the SAMPLE and
265 EVENT type Data Elements.

266 The following demonstrates how the SAMPLE type Data Element PATH_POSITION with
267 subType ACTUAL would be formatted when returning a data value of **7.02**.

```
268 <PathPosition dataItemId="p2" timestamp="2009-03-04T19:45:50.458305"  
269 subType="ACTUAL" name="Zact" sequence="15065113">7.02</PathPosition>  
270
```

271 and the Event type Data Element CONTROLLER_MODE would be formatted when returning a
272 data value of AUTOMATIC.

```
273 <ControllerMode dataItemId="p3" timestamp="2009-02-26T02:02:35.716224"  
274 name="mode" sequence="182">AUTOMATIC</ControllerMode>
```

275 For SAMPLE and EVENT type data items with a representation attribute defined in the
276 Device Information Model as TIME_SERIES or DISCRETE, the tag names for these Data
277 Elements **MUST** be transformed using the Pascal case format (first letter of each word is
278 capitalized) and then adding the representation type. For example, the data item
279 ANGULAR_VELOCITY with a representation defined as TIME_SERIES **MUST** be
280 AngularVelocityTimeSeries and the data item PARTS_COUNT with a
281 representation defined as DISCRETE **MUST** be PartsCountDiscrete.

282 The format for Condition type data items **MUST NOT** be transformed to Pascal case format.
283 Each Condition type data item provides a type attribute that has the same format as defined
284 for the data item in the Device Information Model. An example of the format for a Condition
285 type data item is as follows:

```
286 <Normal type="LOGIC_PROGRAM" id="cc3" sequence="28" timestamp="..."/>
```

287 (Note: The highlighted items are identified for emphasis only)

288

289 5.2 Returned Data Values

290 Every data item provided in a `MtConnectStreams` XML document **MUST** provide a valid
291 data value. A valid data value **MUST** be either a value representing the information provided
292 from the data source or it **MUST** be `UNAVAILABLE`.

293 The `MtConnect` Agent **MUST NOT** send two successive samples for the same data item with
294 the same data value to a client application. A client software application can always assume that
295 if a data item is not present in a `MtConnectStreams` XML document, its value has not
296 changed and it still has the previous value.

297 For `SAMPLE` and `EVENT` category data items, the Returned Data Value is provided in the `CDATA`
298 associated with each data item type XML element in the `MtConnectStreams` document.

299 A `SAMPLE` category data item provides the reading of the value of a continuously variable or
300 analog data value. A `SAMPLE` category data item with a `representation` of `VALUE` can be
301 measured at any point-in-time and **MUST** always produce a result with a single data value. In
302 the case of a `SAMPLE` category data item with a `representation` of `TIME_SERIES`, the
303 data provided **MUST** be a series of data values representing multiple sequential samples of the
304 measured value which will be provided only at the end of the completion of a sampling period.
305 (See *Section 5.4* of this document for more information on `TIME_SERIES` type data).

306 Data values provided for a `SAMPLE` category data item **MUST** always be a floating point
307 number. In `MtConnect`, floating point numbers are defined as XML `xs:float` type numbers as
308 defined by W3C. Any of the following number formats are valid XML floating type numbers: -
309 `1E4`, `1267.43233E12`, `12.78e-2`, `12`, `137.2847`, `-0`, `0` and `INF`. For some `SAMPLE` category data
310 items, the Returned Data Value may be restricted to specific formats. See *Section 5.4.3* of this
311 document for a description of any restrictions on the acceptable format for Returned Data Values.

312 For `SAMPLE` category data items, a client software application can determine the appropriate
313 accuracy of the value for the data item by applying the `significantDigits` attribute
314 defined for that data item in the Device Information Model.

315 The Returned Data Value provided as `CDATA` for a `SAMPLE` category data item **MUST** be
316 formatted as part of the content between the element tags in the XML element representing that
317 data item. As an example, a `Position` is formatted as follows in the XML document:

```
318     <Position sequence="112" timestamp="2007-08-09T12:32:45.1232"  
319     name="Xabs" dataItemId="10">123.3333</Position>
```

320 (Note: The highlighted items are identified for emphasis only)

321 In this example, the `123.3333` is the `CDATA` for the `Position`. All `CDATA` in a `Sample`
322 type element is typed, which means that the value for each type of data item **MUST** be formatted
323 in a specific pattern so that it can be validated using a XML parser.

324

325 An EVENT category data item represents a discrete piece of information from a device. EVENT
326 does not have intermediate values that vary over time, as does SAMPLE. An EVENT is
327 information that, when provided at any specific point in time, represents the current state of the
328 device.

329 The `representation` attribute for a EVENT type data item described in the Device
330 Information Model defines the type of data to be provided for each data item. When
331 `representation` is set to VALUE, the data provided **MUST** be either a floating point
332 number, a descriptive value (text string) representing one of two or more discrete state values
333 defined for that data item, or a text string representing a message.

334 When `representation` is set to DISCRETE, the data provided **MUST** be a numeric value
335 representing a repetitive occurrence of a data value. An EVENT with a `representation` of
336 DISCRETE is the only case where the MTConnect Agent **MAY** provide successive instances of
337 a data item with identical data values since each occurrence of the of the data item represents a
338 different and unique EVENT.

339 The Returned Data Value provided as CDATA for an EVENT category data item **MUST** be
340 formatted as part of the content between the element tags in the XML element representing that
341 data item. The XML elements representing EVENT type data items are formatted as follows in
342 the XML document:

```
343     <PartCount dataItemId="pc4" timestamp="2009-02-26T02:02:36.48303"  
344         name="pcount" sequence="185">238</PartCount>  
345     <ControllerMode dataItemId="p3" timestamp="2009-02-26T02:02:35.716224"  
346         name="mode" sequence="192">AUTOMATIC</ControllerMode>  
347     <Block dataItemId="cn2" name="block" sequence="206"  
348         timestamp="2009-02-26T02:02:37.394055">G0Z1</Block>
```

349 (Note: The highlighted items are identified for emphasis only)

350 In these examples, 238 is the CDATA for the PartCount and is a numeric value; AUTOMATIC
351 is the CDATA for the ControllerMode and is a descriptive value representing a discrete state
352 for the data item; and G0Z1 is a text string representing a message describing the program code
353 associated with the Block data item.

354 A CONDITION category data item communicates information about the health of a device and
355 its ability to function. Valid values reported for a CONDITION category data item can be one of
356 NORMAL, WARNING, or FAULT.

357

358 The Returned Data Value for a CONDITION category data item is provided differently than for a
359 SAMPLE or EVENT category data item. The Returned Data Value is used to characterize each
360 XML element representing a CONDITION category data item based on the state of the data item
361 – NORMAL, FAULT, or WARNING. Examples of XML elements representing CONDITION
362 category data items for each of the available states are:

```
363     <Normal type="MOTION_PROGRAM" id="cc2" sequence="25" timestamp="..." />  
364     <Fault type="COMMUNICATIONS" id="cc1" sequence="26" nativeCode="IO1231"  
365         timestamp="...">Communications error</Fault>  
366     <Normal type="LOGIC_PROGRAM" id="cc3" sequence="28" timestamp="..." />  
367     <Warning type="LOGIC_PROGRAM" id="pm6" sequence="32" timestamp="..." />
```

368 (Note: The highlighted items are identified for emphasis only)

369 Unlike a SAMPLE or EVENT category data item that can only have a single value at any one
370 point in time, some CONDITION type data items **MAY** report multiple simultaneous values
371 (CONDITION). For example, a device controller may detect and report multiple format errors
372 in a motion program. Each error represents a separate Fault (Condition). Each Fault
373 must be identified and tracked individually in the MTConnectStreams document.

374 The representation attribute **MUST** default to VALUE for all CONDITION category data
375 items.

376 The XML element representing a CONDITION category data item **MAY** contain CDATA as part
377 of the content between the element tags to provide additional information further defining the
378 meaning of the CONDITION. In this case, the CDATA will be a text string providing that
379 additional information. An example of a CONDITION category data item reporting its state as
380 WARNING and providing additional information defining the meaning of the CONDITION is
381 formatted as follows in the XML document:

```
382     <Warning type="FILL_LEVEL" id="pm6" qualifier="high" sequence="32"  
383         timestamp="2009-11-13T08:32:18">"Fill Level on Tank #12 is  
384         reaching a high level"/>
```

385 (Note: The highlighted items are identified for emphasis only)

386 In this example, the “Fill Level on Tank #12 is reaching a high level” is the CDATA for the
387 FILL_LEVEL data item.

388 All data items **MUST** report a value of UNAVAILABLE when the data source is not connected to
389 the Agent or the data source is unable to provide a valid data value. The UNAVAILABLE value
390 will persist until the connection is restored and a new valid data value can be retrieved.

391 When a data value of UNAVAILABLE is reported, it does not imply the device is no longer
392 operational. It only implies that a valid data value cannot be determined by the Agent.

393

394 SAMPLE or EVENT category data items report a value of UNAVAILABLE in the CDATA
 395 provided in the XML element returned for each data item. See the examples below:

```

396     <Samples>
397         <PathPosition dataItemId="p2" timestamp="2009-03-04T19:45:50.458305"
398             subType="ACTUAL" name="Zact"
399             sequence="15065113">UNAVAILABLE</PathPosition>
400         <Block dataItemId="p6" timestamp="2009-03-04T19:45:50.458305"
401             name="block" sequence="150651134">UNAVAILABLE</Block>
402     </Samples>
403     <Events>
404         <ControllerMode dataItemId="p3" timestamp="2009-02-26T02:02:35.716224"
405             name="mode" sequence="182">UNAVAILABLE</ControllerMode>
406     </Events>
  
```

407 (Note: The highlighted items are identified for emphasis only)

408 For a CONDITION category data item reporting a Returned Data Value of UNAVAILABLE, the
 409 state of the data item **MUST** be set to UNAVAILABLE. Examples of XML elements
 410 representing CONDITION category data items reporting a state of UNAVAILABLE available are:

```

411     <UNAVAILABLE type="MOTION_PROGRAM" id="cc2" sequence="25"
412         timestamp="..." />
413     <UNAVAILABLE type="COMMUNICATIONS" id="cc1" sequence="26"
414         timestamp="..." />
415     <UNAVAILABLE type="LOGIC_PROGRAM" id="cc3" sequence="28"
416         timestamp="..." />
417     <UNAVAILABLE type="LOGIC_PROGRAM" id="pm6" sequence="32"
418         timestamp="..." />
419
  
```

420 (Note: The highlighted items are identified for emphasis only)

421

422 5.3 samples XML Element

423 Samples is a XML container type element. Samples organizes the data returned in the
 424 MTConnectStreams XML document for the SAMPLE type data items for a specific Structural
 425 Element of a device.

426

Element	Description	Occurrence
Samples	A XML container type element which organizes the data returned in the MTConnectStreams XML document for the SAMPLE type data items for a specific Structural Element of a device. A Samples XML element MUST contain at least one Sample element.	0..1

427

428 **5.4 Sample XML Element**

429 A Sample XML Element provides the information and data provided from a data source for a
 430 SAMPLE type data item. Sample is an abstract type XML element and will never appear
 431 directly in the MTConnectStreams XML document. As an abstract type XML element, each
 432 Sample element will be replaced in the XML document by a specific Sample XML element
 433 representing the SAMPLE category data item defined in the Device Information Model. XML
 434 elements representing Sample are described in Section 5.4.3 of this document and include
 435 elements such as PathPosition, Block, AxisVelocity, etc.

436

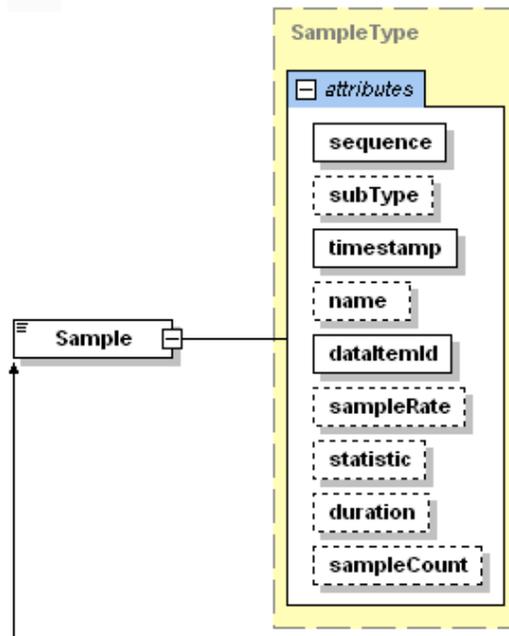
Element	Description	Occurrence
Sample	An abstract XML Element. Replaced in the XML document by type(s) of Sample XML elements representing SAMPLE category data items defined for a Device in the Device Information Model. There can be multiple types of Sample XML Elements in a Samples container.	1..INF

437

438 **5.4.1 XML Schema Structure for Sample**

439 The following XML tree represents the structure of a Sample XML element showing the
 440 attributes defined for Sample type XML elements.

441



442

443

Figure 6: Sample Schema Diagram

444 5.4.2 **Attributes for Sample**

445 The following table defines the attributes that may be used to provide additional information for
 446 a Sample type XML element.

447

Attribute	Description	Occurrence
name	<p>The name of the Sample. The name MUST match the name of the data item defined in the Device Information Model that this Sample element represents.</p> <p>An NMTOKEN XML type.</p>	0..1
sequence	<p>A number representing the sequential position of an occurrence of the Sample in the data buffer of the Agent.</p> <p>The value MUST be represented as an unsigned 64 bit with valid values from 1 to 2⁶⁴-1.</p>	1
timestamp	<p>The time the data for the Sample was reported or the statistics for the Sample was computed.</p> <p>The timestamp MUST always represent the end of the collection interval when a duration or a TIME_SERIES is provided.</p> <p>The most accurate time available to the device MUST be used for the timestamp.</p>	1
dataItemID	<p>The unique identifier for the Sample. The dataItemID MUST match the id attribute of the data item defined in the Device Information Model that this Sample element represents.</p>	1
subType	<p>The subtype of the DataItem defined in the Device Information Model that this Sample element represents.</p>	0..1
sampleRate	<p>The rate at which successive samples of the value of a DataItem are recorded. sampleRate is expressed in terms of samples per second.</p> <p>If the sampleRate 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>The sampleRate attribute MUST be provided when the representation attribute for the data item is TIME_SERIES.</p> <p>For data items where the representation attribute for the data item IS NOT TIME_SERIES, it may be assumed that the sampleRate is constant and sampleRate does not need to be reported in the MTConnectStreams document.</p>	0..1
statistic	<p>The type of statistical calculation specified in the Device Information Model that this Sample element represents.</p>	0..1

Attribute	Description	Occurrence
duration	<p>The time elapsed since the statistic calculation was last reset.</p> <p>The duration attribute MUST be provided when the value of the data returned for the data item is a <i>statistic</i>.</p>	0..1
sampleCount	<p>The number of readings of the value of a data item provided in the data returned when the <i>representation</i> attribute for the data item is <i>TIME_SERIES</i>.</p> <p>sampleCount is not provided for data items unless the <i>representation</i> attribute is <i>TIME_SERIES</i> and it MUST be specified when the attribute is <i>TIME_SERIES</i>.</p>	0..1

448

449 For Sample type data items containing an attribute for duration, the timestamp associated
 450 with the Sample references the time the data value was reported or the statistics for the data
 451 item were computed - NOT the time the interval began. The actual time the interval began can
 452 be computed by subtracting the duration from the timestamp.

453 Two Sample type data items can have overlapping time periods as in the case where statistics
 454 are computed at various frequencies. For example, a one minute average and a five minute
 455 average can both have the same start time (e.g. 05:10:00), but their timestamps will be 05:11:00
 456 for the data item with a duration of 60 seconds for the one minute average and the second
 457 data item with a duration of 300 seconds will have a timestamp of 05:15:00 for the five minute
 458 average. This allows for varying statistical methods to be applied with different interval lengths
 459 without having duplicate timestamps and durations. If a statistical data item does not report a
 460 value for a period greater than the previous duration, it can be assumed the computed value has
 461 not changed since the last value was reported.

462 When the representation attribute for a SAMPLE category data item is specified as
 463 TIME_SERIES, the data reported for the data item **MUST** include multiple readings of the
 464 value for the data item taken at a specified sample rate. A TIME_SERIES type data item can be
 465 used for collecting high frequency samples of the value for a data item and then providing the
 466 entire series of data values to an application as a single data item. For a TIME_SERIES type
 467 data item, the sampleCount and sampleRate attributes **MUST** be provided. For a
 468 TIME_SERIES type data item, sampleRate defines the time period (frequency) for the
 469 collection of each reading of the value of the data item and sampleCount defines the total
 470 number of readings being transmitted. The CDATA provided for the data item **MUST** be a series
 471 of floating point numbers. The number of readings **MUST** match the sampleCount. Also,
 472 the units for a data item with the representation attribute of TIME_SERIES **MUST** be the
 473 same as the units specified for that type of data item.

474

475

476 5.4.3 **Sample XML Element Tag Names**

477 The following is a list of all of the XML elements that can be placed in the Samples section of
 478 the ComponentStream. The table shows both the type for each SAMPLE category data
 479 item defined in the Device Information Model and the corresponding transformed designation for
 480 that same data item that **MUST** be provided as a Sample XML element in the
 481 MTConnectStreams XML document.

482 (Note: The Sample response format **MUST** be extended to represent those data items
 483 where the representation attribute is TIME_SERIES. See Section 5.1 of this
 484 document for details on extending the response format.)

485

Data Element SAMPLE	Response Format Sample	Description
ACCELERATION	Acceleration	Rate of change of velocity. Acceleration MUST be reported in units of MILLIMETER/SECOND ² .
ACCUMULATED_TIME	AccumulatedTime	The measurement of accumulated time for an activity or event. AccumulatedTime MUST be reported in units of SECOND.
ANGULAR_ACCELERATION	AngularAcceleration	Rate of change of angular velocity. AngularAcceleration MUST be reported in units of DEGREE/SECOND ² .
ANGULAR_VELOCITY	AngularVelocity	Rate of change of angular position. AngularVelocity MUST be reported in units of DEGREE/SECOND.
AMPERAGE	Amperage	The measurement of Electrical Current. Sub-types of Amperage are ALTERNATING and DIRECT. Amperage MUST be reported in units of AMPERE.
ANGLE	Angle	The measurement of angular position. Sub-types of Angle are ACTUAL and COMMANDED. Angle MUST be reported in units of DEGREE.
AXIS_FEEDRATE	AxisFeedrate	The feedrate of a linear axis. Sub-types of AxisFeedrate are ACTUAL, COMMANDED, JOG, PROGRAMMED, and RAPID. If a subType is not specified, it MUST default to PROGRAMMED. AxisFeedrate MUST be reported in units of MILLIMETER/SECOND.

Data Element SAMPLE	Response Format Sample	Description
CLOCK_TIME	ClockTime	The value provided by a timing device at a specific point in time. ClockTime MUST be reported in W3C ISO 8601 format of YYYY-MM-DDThh:mm:ss.ffff.
CONCENTRATION	Concentration	Percentage of one component within a mixture of components. Concentration MUST be reported in units of PERCENT.
CONDUCTIVITY	Conductivity	The ability of a material to conduct electricity. Conductivity MUST be reported in units of SIEMENS/METER.
DISPLACEMENT	Displacement	The change in position of an object. Displacement MUST be reported in units of MILLIMETER.
ELECTRICAL_ENERGY	ElectricalEnergy	The measurement of electrical energy consumption by a component. ElectricalEnergy MUST be reported in units of WATT_SECOND.
FILL_LEVEL	FillLevel	The measurement of the amount of a substance remaining compared to the planned maximum amount of that substance. FillLevel MUST be reported in units of PERCENT.
FLOW	Flow	The rate of flow of a fluid. Flow MUST be reported in units of LITER/SECOND.
FREQUENCY	Frequency	The measurement of the number of occurrences of a repeating event per unit time. Frequency MUST be reported in units of HERTZ.
GLOBAL_POSITION	GlobalPosition	DEPRECATED in Rel. 1.1
LEVEL	Level	DEPRECATED in Rel. 1.2 See FILL_LEVEL

Data Element SAMPLE	Response Format Sample	Description
LENGTH	Length	<p>The length of an object.</p> <p>Sub-types of Length are STANDARD, REMAINING, and USEABLE .</p> <p>Length MUST be report in units of MILLIMETER.</p>
LINEAR_FORCE	LinearForce	<p>The measure of the push or pull introduced by an actuator or exerted on an object.</p> <p>LinearForce MUST be reported in units of NEWTON.</p>
LOAD	Load	<p>The measurement of the actual versus the standard rating of a device.</p> <p>Load MUST 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 MUST be reported in units of KILOGRAM.</p>
PATH_FEEDRATE	PathFeedrate	<p>The feedrate of the tool path.</p> <p>Sub-types of PathFeedrate are ACTUAL, COMMANDED, JOG, PROGRAMMED, and RAPID.</p> <p>If a subType is not specified, it MUST default to PROGRAMMED.</p> <p>PathFeedrate MUST be reported in units of MILLIMETER / SECOND.</p>
PATH_POSITION	PathPosition	<p>The current program control point expressed in WORK coordinates. The coordinate system will revert to MACHINE coordinates if WORK coordinates are not available.</p> <p>Sub-types of PathPosition are ACTUAL, COMMANDED, TARGET, and PROBE .</p> <p>PathPosition MUST be provided as a space delimited vector of floating point numbers given in units of MILLIMETER_3D listed in order X, Y, and Z:</p> <p><PathPosition ...>10.123 55.232 100.981 </PathPosition></p> <p>Where X = 10.123, Y = 55.232, and Z=100.981.</p>
PH	PH	<p>The measure of the acidity or alkalinity.</p> <p>PH MUST be reported in units of PH.</p>

Data Element SAMPLE	Response Format Sample	Description
POSITION	Position	<p>The position of the Component.</p> <p>Sub-types of Position are ACTUAL, COMMANDED, and TARGET.</p> <p>When a subType for POSITION is not specified, it MUST default to ACTUAL.</p> <p>When Position type data is provided representing a measured value for the physical axes of the device, this data MUST be given in MACHINE coordinates.</p> <p>When Position type data is provided representing a logical or calculated location on the device, this data MUST be given in WORK coordinates and is associated with the Path element of the Controller.</p> <p>Position MUST 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 MUST be reported in units of PERCENT.</p>
PRESSURE	Pressure	<p>The force per unit area exerted by a gas or liquid.</p> <p>Pressure MUST be reported in units of PASCAL.</p>
RESISTANCE	Resistance	<p>The measurement of the degree to which an object opposes an electric current through it.</p> <p>Resistance MUST be reported in units of OHM.</p>
ROTARY_VELOCITY	RotaryVelocity	<p>The rotational speed of a rotary axis.</p> <p>Sub-types of RotaryVelocity are ACTUAL, COMMANDED, and PROGRAMMED.</p> <p>RotaryVelocity MUST be reported in units of REVOLUTION/MINUTE.</p>
SOUND_LEVEL	SoundLevel	<p>Measurement of a sound level or sound pressure level relative to atmospheric pressure.</p> <p>Sub-types of SoundLevel are NO_SCALE, A_SCALE, B_SCALE, C_SCALE, and D_SCALE.</p> <p>SoundLevel MUST be provided in DECIBEL.</p>
SPINDLE_SPEED	SpindleSpeed	<p>DEPRECATED in REL 1.2.</p> <p>Replaced by ROTARY_VELOCITY</p>
STRAIN	Strain	<p>The amount of deformation per unit length of an object when a load is applied.</p> <p>Strain MUST be reported in units of PERCENT.</p>

Data Element SAMPLE	Response Format Sample	Description
TEMPERATURE	Temperature	The measurement of temperature. Temperature MUST be reported in units of degrees CELSIUS.
TILT	Tilt	A measurement of angular displacement. Tilt MUST be reported in units of MICRO_RADIAN.
TORQUE	Torque	The turning force exerted on an object or by an object. Torque MUST be reported in units of NEWTON_METER.
VOLT_AMPERE	VoltAmpere	The measure 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) VoltAmpere MUST be reported in units of VOLT_AMPERE.
VOLT_AMPERE_REACTIVE	VoltAmpereReactive	The measurement of reactive power in an AC electrical circuit (commonly referred to as VAR), VoltAmpereReactive MUST be reported in units of VOLT_AMPERE_REACTIVE.
VELOCITY	Velocity	The rate of change of position of a component. When provided as the Velocity of the Axes component, it represents the value of the velocity vector for all given axis, similar to a path feedrate. 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. Velocity MUST be reported in units of MILLIMETER/SECOND.
VISCOSITY	Viscosity	A measurement of a fluid's resistance to flow. Viscosity MUST be reported in units of PASCAL_SECOND.
VOLTAGE	Voltage	The measurement of electrical potential between two points. Sub-types of Voltage are ALTERNATING and DIRECT. Voltage MUST be reported in units of VOLT.
WATTAGE	Wattage	The measurement of power consumed or dissipated by an electrical circuit or device. Wattage MUST be reported in units of WATT.

487 **5.5 Events XML Elements**

488 Events is a XML container type element. Events organizes the data returned in the
489 MTConnectStreams document for the EVENT type data items for a specific Structural
490 Element of a device.

Element	Description	Occurrence
Events	A XML container type element which organizes the data returned in the MTConnectStreams XML document for the EVENT type data items for a specific Structural Element of a device. An Events XML element MUST contain at least one Event element.	1..INF

491

492 **5.6 Event XML Elements**

493 An Event XML Element provides the information and data provided from a data source for an
494 EVENT type data item. Event is an abstract type XML element and will never appear directly
495 in the MTConnectStreams XML document. As an abstract type XML element, each Event
496 element will be replaced in the XML document by a specific Event XML element representing
497 the EVENT category data item defined in the Device Information Model. XML elements
498 representing Event are described in *Section 5.6.3* of this document and include elements such
499 as Block, Execution, and Line.

500 Event type elements **MAY** have values defined by a controlled vocabulary as specified in
501 *Section 5.6.3* of this document, **MAY** have numeric values, or **MAY** contain a character string
502 representing text or a message provided by the device.

503 An Event is similar to a Sample, but its value can change with unpredictable frequency.
504 Events do not report intermediate values. As an example: When Availability transitions
505 from UNAVAILABLE to AVAILABLE, there is no intermediate state that can be inferred. As a
506 result, many Event type elements have a controlled vocabulary as their content.

507 An Event may also represent a message.

508

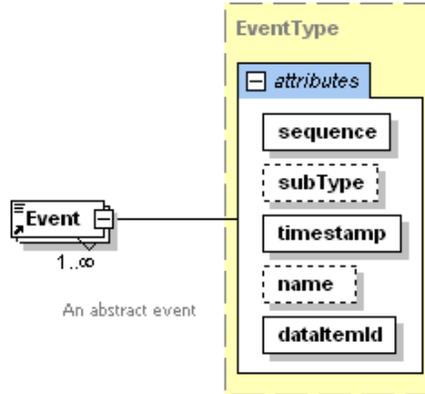
Element	Description	Occurrence
Event	An abstract XML Element. Replaced in the XML document by type(s) of Event XML elements representing EVENT category data items defined for a Device in the Device Information Model. There can be multiple types of Event XML Elements in an Events container.	1..INF

509

510

511 5.6.1 XML Schema Structure for Event

512 The following XML tree represents the structure of a Event XML element showing the
 513 attributes defined for Event type XML elements.



514

515

Figure 7: Event Schema Diagram

516 5.6.2 Attributes for Event:

517 The following table defines the attributes that may be used to provide additional information for
 518 an Event type XML element.

Attribute	Description	Occurrence
name	The name of the Event. The name MUST match the name of the data item defined in the Device Information Model that this Event element represents. An NMTOKEN XML type.	0..1
sequence	A number representing the sequential position of an occurrence of the Event in the data buffer of the Agent. The value MUST be represented as an unsigned 64 bit with valid values from 1 to 2 ⁶⁴ -1.	1
timestamp	The time the data for the Event was reported. The most accurate time available to the device MUST be used for the timestamp.	1
dataItemID	The unique identifier for the Event. The dataItemID MUST match the id attribute of the data item defined in the Device Information Model that this Event element represents.	1
subType	The subtype of the DataItem defined in the Device Information Model that this Event element represents.	0..1

519

520 If the representation of an EVENT type data item is specified as DISCRETE , it indicates
 521 that each occurrence of the reported data may have the same value as the previous occurrence of
 522 the reported data. Basically, there is no reported state change between occurrences of the data
 523 and each occurrence of the data item represents a different and unique Event.
 524 In this case, the MTConnect Agent **MAY** provide successive samples of a data item with
 525 identical data values and duplicate occurrences of the same data value **SHOULD NOT** be
 526 suppressed.

527 Examples of a data type that could be characterized by an EVENT type element with a
 528 representation of DISCRETE would be a Parts Counter that reports the completion of each part
 529 produced, versus the accumulation of parts over time. In this case, the data returned would be
 530 represented by the PartsCountDiscrete data element and each occurrence of this data
 531 element in a MTConnectStreams document would indicate the completion of a fixed number
 532 of parts (typically 1).

533 Message is another Event type XML data element which may not report a state change
 534 between occurrences of the data and each occurrence of the data item represents a different and
 535 unique Event. In this case, the message does not have a reset state. If a message does not have
 536 a reset state, then it **SHOULD** be defined with a representation of DISCRETE. In this
 537 case, the data returned would be represented by the MessageDiscrete data element and each
 538 occurrence of this data element in a MTConnectStreams document would indicate a unique
 539 occurrence of the message.

540 5.6.3 Event XML Element Tag Names

541 The following table provides a list of all of the XML elements that can be placed in the Events
 542 section of the ComponentStream. The table shows the type for each EVENT category data
 543 item defined in the Device Information Model, the corresponding transformed designation for
 544 that same data item that **MUST** be provided as an Event XML element in the
 545 MTConnectStreams XML document, and the Controlled Vocabulary for those Event types
 546 that represent specific state information from a device.

547 (Note: The Event response format **MUST** be extended to represent those data items
 548 where the representation attribute is DISCRETE. See Section 5.1 of this document
 549 for details on extending the response format.)

Data Element EVENT	Response Format Event	Description and Valid Data Values
ACTUATOR_STATE	ActuatorState	ActuatorState represents the operational state of an apparatus for moving or controlling a mechanism or system. Valid Data Values: -ACTIVE: The actuator is operating and is active -INACTIVE: The actuator is not operating and is inactive

Data Element EVENT	Response Format Event	Description and Valid Data Values
ALARM	Alarm	DEPRECATED: Replaced with CONDITION category data items in <i>Rel. 1.1</i> .
ACTIVE_AXES	ActiveAxes	<p>The set of axes being controlled by a Path.</p> <p>The value reported MUST be a space delimited set of axes names. For example:</p> <p style="text-align: center;"><ActiveAxes ...>X Y Z C</ActiveAxes></p> <p>If it is not specified elsewhere in the Device Information Model, it MUST be assumed that the Path is controlling all of the axes.</p>
AVAILABILITY	Availability	<p>Represents the ability of a Structural Element to communicate. AVAILABILITY MUST be provided for a Device Element and MAY be provided for any other Structural Element.</p> <p>Valid Data Values:</p> <ul style="list-style-type: none"> -AVAILABLE: The Structural Element is active and capable of providing data. -UNAVAILABLE: The Structural Element is either inactive or not capable of providing data.
AXIS_COUPLING	AxisCoupling	<p>Describes the way the axes will be associated to each other. This is used in conjunction with COUPLED_AXES to indicate the way they are interacting.</p> <p>The coupling MUST be viewed from the perspective of the specified axis. Therefore a MASTER coupling indicates that this axis is the master for the COUPLED_AXES.</p> <p>Valid Data Values:</p> <ul style="list-style-type: none"> -TANDEM: The axes are physically connected to each other and MUST operate as a single unit. -SYNCHRONOUS: The axes are not physically connected to each other but are operating together in lockstep. -MASTER: The axis is the master of the CoupledAxes -SLAVE: The axis is a slave to the CoupledAxes

Data Element EVENT	Response Format Event	Description and Valid Data Values
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>Sub-types of AxisFeedrateOverride are JOG, PROGRAMMED, and RAPID.</p> <p>If a subType is not specified, it MUST default to PROGRAMMED. A Valid Data Value MUST 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>Valid Data Values:</p> <ul style="list-style-type: none"> -ACTIVE: The axis lockout function is activated, power has been removed from the axis, and the axis is allowed to move freely. -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.
AXIS_STATE	AxisState	<p>An indicator of the motion state of an axis or whether it is in a homed position.</p> <p>Valid Data Values:</p> <ul style="list-style-type: none"> -HOME: The axis is in its home position. -TRAVEL: The axis is in motion -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. -STOPPED: The axis is stopped
BLOCK	Block	<p>Block is the line of code or command being executed by a Controller Structural Element. Block MUST include the entire expression for a line of program code, including all parameters</p> <p>A Valid Data Value MAY be any text string.</p>

Data Element EVENT	Response Format Event	Description and Valid Data Values
CHUCK_INTERLOCK	ChuckInterlock	<p>An indication of the state of an interlock function or control logic state intended to prevent the associated CHUCK Structural Element from being operated.</p> <p>A CHUCK Structural 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 MAY be further characterized by specifying a subType of MANUAL_UNCLAMP.</p> <p>Valid Data Values:</p> <ul style="list-style-type: none"> -ACTIVE : The chuck MAY NOT be unclamped -INACTIVE: The chuck MAY be unclamped.
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 device.</p> <p>Valid Data Values:</p> <ul style="list-style-type: none"> -OPEN : The CHUCK is open to the point of a positive confirmation -CLOSED : The CHUCK is closed to the point of a positive confirmation -UNLATCHED : The CHUCK 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.
CODE	Code	DEPRECATED in Rel 1.1.

Data Element EVENT	Response Format Event	Description and Valid Data Values
CONTROLLER_MODE	ControllerMode	<p>The current operating mode of the Controller.</p> <p>Valid Data Values:</p> <ul style="list-style-type: none"> -AUTOMATIC: The controller is configured to automatically execute a program. -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. -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. There is no active program currently being executed. -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. -EDIT: The controller is currently functioning as a programming device and is not capable of executing an active program.
COUPLED_AXES	CoupledAxes	<p>Refers to a set of associated axes.</p> <p>Used in conjunction with <code>AxisCoupling</code> to describe how the <code>CoupledAxes</code> relate to each other.</p> <p>A Valid Data Value MUST be a space delimited set of axes names.</p> <p><code><CoupledAxes ...>Y1 Y2</CoupledAxes ></code></p>
DIRECTION	Direction	<p>The direction of motion.</p> <p>A <code>subType</code> MUST always be specified. Sub-types of <code>Direction</code> are <code>LINEAR</code> and <code>ROTARY</code>.</p> <p>Valid Data Values:</p> <ul style="list-style-type: none"> -CLOCKWISE: A <code>ROTARY</code> type component is rotating in a clockwise fashion using the right hand rule. -COUNTER_CLOCKWISE: A <code>ROTARY</code> type component is rotating in a counter clockwise fashion using the right hand rule. -POSITIVE: A <code>LINEAR</code> type component is moving in the direction of increasing position value -NEGATIVE: A <code>LINEAR</code> type component is moving in the direction of decreasing position value

Data Element EVENT	Response Format Event	Description and Valid Data Values
DOOR_STATE	DoorState	<p>The operational state of a DOOR type component.</p> <p>Valid Data Values:</p> <ul style="list-style-type: none"> -OPEN: The Door is open to the point of a positive confirmation -CLOSED: The Door is closed to the point of a positive confirmation -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.
END_OF_BAR	EndOfBar	<p>An indication of whether the end of a piece of bar stock being feed by a bar feeder has been reached.</p> <p>Sub-types of EndOfBar are PRIMARY and AUXILIARY .</p> <p>Valid Data Values:</p> <ul style="list-style-type: none"> -YES: The EndOfBar has been reached. -NO: The EndOfBar has not been reached.
EMERGENCY_STOP	EmergencyStop	<p>The current state of the emergency stop signal for a machine, device, controller path, or any other component or subsystem of a device.</p> <p>Valid Data Values:</p> <ul style="list-style-type: none"> -ARMED: The emergency stop circuit is complete and the device is allowed to operate. -TRIGGERED: The emergency stop circuit is open and the operation of the device is inhibited.

Data Element EVENT	Response Format Event	Description and Valid Data Values
EXECUTION	Execution	<p>The execution status of the Controller.</p> <p>Valid Data Values:</p> <ul style="list-style-type: none"> -READY: The controller is ready to execute instructions. It is currently idle. -ACTIVE: The controller is actively executing an instruction. -INTERRUPTED: The execution of the controller's program has been stopped in an unplanned manner. Action is required to resume execution. -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. -STOPPED: The execution of the controller's program has been stopped in an unplanned manner and execution of the program cannot be resumed. -OPTIONAL_STOP: The controller's program has been intentionally stopped using an M01 or similar code. The program may or may not be stopped at the designated location based upon the state of a secondary indication provided to the controller defining whether the program execution must be stopped at this location or program execution should continue. -PROGRAM_STOPPED: The execution of the controller's program has been stopped in a planned manner. Action is required to resume execution. -PROGRAM_COMPLETED: The program has completed execution .

Data Element EVENT	Response Format Event	Description and Valid Data Values
FUNCTIONAL_MODE	FunctionalMode	<p>The current intended production status or intended use of the device or component.</p> <p>Typically, the FunctionalMode SHOULD be associated with the Device Structural Element, but it MAY be associated with any Structural Element in the XML document.</p> <p>Valid Data Values:</p> <ul style="list-style-type: none"> -PRODUCTION: The Device or other Structural Element is currently producing product, ready to produce product, or its current intended use is to be producing product. -SETUP: The Device or other Structural Element is not currently producing product. It is being prepared or modified to begin production of product. -TEARDOWN: The Device or other Structural Element 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. -MAINTENANCE: The Device or other Structural Element 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. -PROCESS_DEVELOPMENT: The Device or other Structural Element is not currently producing product in a production mode. It 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.
INTERFACE_STATE	InterfaceState	<p>The current functional or operational state of an INTERFACE type element indicating whether the interface is active or not currently functioning.</p> <p>Valid Data Values:</p> <ul style="list-style-type: none"> -ENABLED: The interface is currently operational and performing as expected. -DISABLED: The interface is currently not operational. <p>When the INTERFACE_STATE is DISABLED, the state of all other data elements associated with that Interface MUST be set to NOT_READY.</p>

Data Element EVENT	Response Format Event	Description and Valid Data Values
LINE	Line	<p>The current line number of the program code being executed by a Controller Structural Element.</p> <p>For a RS274/NGC formatted part program, the line number begins with an N and is followed by 1 to 5 digits (0 – 99999).</p> <p>If there is not an assigned line number in the control program as in an RS274 formatted program, the line number MAY refer to the position in the current code being executed in the control program.</p> <p>Sub-types of Line are MAXIMUM and MINIMUM.</p> <p>A Valid Data Value MUST be an alpha-numeric text string.</p>
MESSAGE	Message	<p>Any text string of information to be transferred from a device to a client software application.</p> <p>A Valid Data Value MAY be any text string.</p>
OPERATOR_ID	OperatorID	<p>The identifier of the person currently assigned to a piece of equipment.</p> <p>A Valid Data Value MAY be any text string.</p>
PALLET_ID	PalletID	<p>The identifier for the pallet currently associated with a piece of equipment.</p> <p>A Valid Data Value MAY be any text string.</p>
PART_COUNT	PartCount	<p>The current count of parts or other product produced as reported by the Controller Structural Element.</p> <p>Sub-types of PartCount are ALL, GOOD, BAD, TARGET, and REMAINING.</p> <p>Part_Count will not be accumulated by the MTConnect Agent and MUST only be supplied if the Controller provides the count.</p> <p>Part_Count MAY have a representation of DISCRETE. In this case, each occurrence of PartCount in an MTConnectStreams document represents a unique count of parts or product produced – it is not an accumulated count of parts or product produced.</p> <p>A Valid Data Value MUST be a floating point number, usually an integer.</p>

Data Element EVENT	Response Format Event	Description and Valid Data Values
PART_ID	PartID	<p>The identifier for a part or other type of product currently associated with a piece of equipment.</p> <p>A Valid Data Value MAY be any text string.</p>
PATH_FEEDRATE_OVERRIDE	PathFeedrate-Override	<p>The value of a signal or calculation issued to adjust the feedrate for the axes associated with a Path component - may represent a single axis or the coordinated movement of multiple axes.</p> <p>The value provided for PathFeedrateOverride is expressed as a percentage of the designated feedrate for the path.</p> <p>Sub-types of PathFeedrateOverride are JOG, PROGRAMMED, and RAPID.</p> <p>If a subType is not specified, it MUST default to PROGRAMMED.</p> <p>A Valid Data Value MUST be a floating point number.</p>
PATH_MODE	PathMode	<p>Used to describe the operational relationship between a PATH Structural Element and another PATH Structural Element for pieces of equipment comprised of multiple logical groupings of controlled axes or other logical operations.</p> <p>Valid Data Values:</p> <ul style="list-style-type: none"> - INDEPENDENT: The path is operating independently and without the influence of another path. - 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 - SYNCHRONOUS: The axes associated with the path are following the motion of the MASTER type path. - MIRROR: The axes associated with the path are mirroring the motion of the MASTER path. <p>When PathMode is not specified, it MUST be assumed to be INDEPENDENT.</p>

Data Element EVENT	Response Format Event	Description and Valid Data Values
POWER_STATE	PowerState	<p>The indication of the status of the source of energy for a Structural Element to allow it to perform its intended function. Sub-types of PowerState are LINE and CONTROL.</p> <p>When the subType is LINE, PowerState represents the primary source of energy for a Structural Element.</p> <p>When the subType is CONTROL, PowerState represents an enabling signal providing permission for the Structural Element to perform its function(s).</p> <p>Valid Data Values:</p> <ul style="list-style-type: none"> -ON: The source of energy for a Structural Element or the enabling signal providing permission for the Structural Element to perform its function(s) is present and active. -OFF: The source of energy for a Structural Element or the enabling signal providing permission for the Structural Element to perform its function(s) is not present or is disconnected. <p>DEPRECATION WARNING: PowerState MAY be deprecated in the future.</p>
POWER_STATUS	PowerStatus	DEPRECATED in <i>Rel. 1.1</i> .
PROGRAM	Program	<p>The name of the control program being executed by the Controller Structural Element. This is usually the name of the file containing the program instructions.</p> <p>A Valid Data Value MAY be any text string.</p>

Data Element EVENT	Response Format Event	Description and Valid Data Values
PROGRAM_EDIT	ProgramEdit	<p>An indication of the status of the Controller Structural Element's program editing function.</p> <p>On many controls, a program can be edited while another program is currently being executed. ProgramEdit provides an indication of when/if the controller is being used to edit programs.</p> <p>Valid Data Values:</p> <ul style="list-style-type: none"> -ACTIVE: The controller is in the program edit mode. -READY: The controller is capable of entering the program edit mode and no function is inhibiting a change to that mode. -NOT_READY: A function is inhibiting the controller from entering the program edit mode.
PROGRAM_EDIT_NAME	ProgramEditName	<p>The name of the control program being edited.</p> <p>This is used in conjunction with PROGRAM_EDIT when it is in an ACTIVE state.</p> <p>A Valid Data Value MAY be any text string.</p>
PROGRAM_COMMENT	ProgramComment	<p>A comment or non-executable statement in the control program.</p> <p>A Valid Data Value MAY be any text string.</p>
PROGRAM_HEADER	ProgramHeader	<p>The non-executable header section of the current control program. The content SHOULD be limited to 512 bytes.</p> <p>A Valid Data Value MAY be any text string.</p>
ROTARY_MODE	RotaryMode	<p>The current operating mode for a Rotary type axis.</p> <p>Valid Data Values:</p> <ul style="list-style-type: none"> -SPINDLE: The axis is functioning as a spindle. Generally, it is configured to rotate at a defined speed. -INDEX: The axis is configured to index to a set of fixed positions or to incrementally index by a fixed amount. -CONTOUR: The position of the axis is being interpolated as part of the PathPosition defined by the Controller Structural Element.

Data Element EVENT	Response Format Event	Description and Valid Data Values
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>A Valid Data Value MUST be a floating point number.</p>
SPINDLE_INTERLOCK	SpindleInterlock	<p>An indication of the status of the spindle for a device when power has been removed and it is free to rotate.</p> <p>Valid Data Values:</p> <ul style="list-style-type: none"> -ACTIVE: Power has been removed and the spindle cannot be operated. -INACTIVE: Spindle has not been deactivated.
TOOL_ID	ToolID	<p>DEPRECATED in Rel. 1.2. See Tool_ASSET_ID. The identifier of the tool currently in use for a given Path</p>
TOOL_ASSET_ID	ToolAssetID	<p>The identifier of an individual tool asset which is the current tool in use by a piece of equipment.</p> <p>A Valid Data Value MAY be any text string.</p>
TOOL_NUMBER	ToolNumber	<p>The identifier assigned by the Controller component to the current tool in use by a piece of equipment.</p> <p>A Valid Data Value MAY be any text string.</p>
WORKHOLDING_ID	WorkholdingID	<p>The identifier for the current workholding or part clamp in use by a piece of equipment.</p> <p>The data MUST be any text string.</p>

550

551

552

553 **5.6.4 EVENT Category DataItem Types Specific for Interface**

554 A special set of DataItem types have been defined to be used in conjunction with
 555 Interface type Structural Elements. These DataItem types are Event Category data
 556 items. They provide information from a piece of equipment that indicates that it has a
 557 requirement for a service or services to be performed by a second piece of equipment. The
 558 behavior of these data items is the same as all other Event type data items, except that they all
 559 share the same set of Valid Data Values and the information represented by these data items
 560 defines specific types of interactions between pieces of equipment.

561 See *Part 3.1: Interfaces* for a detailed description of the operation for Interfaces.

562 The following table provides a list of the additional XML data elements that can be placed in the
 563 Events section of the ComponentStream for an Interface type Structural
 564 Element. The table shows the type for each of these EVENT category data and the
 565 corresponding transformed designation for that same data item that MUST be provided as an
 566 Event XML element in the MTConnectStreams XML document. The Controlled Vocabulary for
 567 these Event types represent specific state information from a device. The Controlled Vocabulary
 568 is the same for all of these data item types and is listed in the second table below.
 569

Data Element EVENT	Response Format Event	Description
MATERIAL_FEED	MaterialFeed	Service to load or feed material or product to a piece of equipment from a continuous or bulk source
MATERIAL_CHANGE	MaterialChange	Service to request a change in the type of material or product being loaded or fed to a piece of equipment.
MATERIAL_RETRACT	MaterialRetract	Service to request that material or product be removed or retracted from a piece of equipment.
PART_CHANGE	PartChange	Service to request that the type of part or product being made by a piece of equipment be changed to a different part or product type. Coupled with PART_ID to indicate the part or product type.
MATERIAL_LOAD	MaterialLoad	Service to request for a piece of material or product be loaded to a piece of equipment.
MATERIAL_UNLOAD	MaterialUnload	Service to request for a piece of material or product be unloaded from a piece of equipment.
OPEN_DOOR	OpenDoor	Service to request another piece of equipment to open a door.
CLOSE_DOOR	CloseDoor	Service to request another piece of equipment to close a door.
OPEN_CHUCK	OpenChuck	Service to request another piece of equipment to open a chuck.
CLOSE_CHUCK	CloseChuck	Service to request another piece of equipment to close a chuck.

570

571 For each of the data item types above, the Valid Data Values for the CDATA for these data items
 572 is defined by a Controlled Vocabulary. This Controlled Vocabulary represents specific state
 573 information from a device and is defined in the table below:

574

Valid Data Value	Description
NOT_READY	The request or response is not ready to perform the action
READY	The request or response capable of performing an action, but no action has been initiated
ACTIVE	The request or response is actively performing the action.
FAIL	The request or response has failed to perform the action
COMPLETE	The response is now completed.

575

576 5.7 Condition XML Elements

577 Condition is a XML container type element. Condition organizes the data returned in the
 578 MTConnectStreams document for the CONDITION type data items for a specific Structural
 579 Element of a device.

580

Element	Description	Occurrence
Condition	A XML container type element which organizes the data returned in the MTConnectStreams XML document for the CONDITION type data items for a specific Structural Element of a device. A Condition XML container element MUST contain at least one Condition type Data Element.	1..INF

581

582 The contents of a Condition type container are represented differently in the XML document
 583 than the contents of Samples and Events container type elements. Samples and Events
 584 container type elements hold individual Sample and Event type data items which are defined
 585 in the Device Information Model. In the MTConnectStreams XML document, the
 586 Condition type data items defined in the Device Information Model are replaced with data
 587 elements that communicate the ability of the device to function.

588 **5.8 Condition Type Data Elements**

589 A Condition type Data Element provides the information and data provided from a data
590 source for a CONDITION type data item. Condition provides a method by which the device
591 can communicate its health and ability to function. A Condition type Data Element is an
592 abstract type XML element and will never appear directly in the MTConnectStreams XML
593 document. As an abstract type XML element, each Condition type XML Element will be
594 replaced in the XML document by a specific Condition Data Element representing a
595 CONDITION category data item defined in the Device Information Model.

596

Element	Description	Occurrence
Condition Type Data Elements	An abstract XML Element. Replaced in the XML document by type(s) of Condition type Data Elements representing CONDITION category data items defined for a Device in the Device Information Model. There can be multiple types of Condition type XML Elements in a Condition container.	1..INF

597

598 The Returned Data Value for Condition type Data Elements is reported differently than
599 Sample or Event elements. Condition type Data Elements **MUST** be reported as
600 NORMAL, WARNING, FAULT, or UNAVAILABLE. Also, some Condition type Data
601 Element **MAY** have multiple active Conditions at one time whereas a Sample or Event
602 can only have a single value at any point in time.

603 XML elements representing Condition type elements are described in *Section 5.8.3* of this
604 document.

605 The Returned Data Values defined for Condition type Data Elements are defined as follows:

606 **Normal:** The item being monitored is operating normally and no action is required.
607 Normal also indicates a Fault or Warning condition has been cleared if the item was
608 previously identified with Fault or Warning.

609 **Warning:** The item being monitored is moving into an abnormal range and should be
610 observed. No action is required at this time. Transition to a Normal condition indicates
611 that the Warning condition has been cleared.

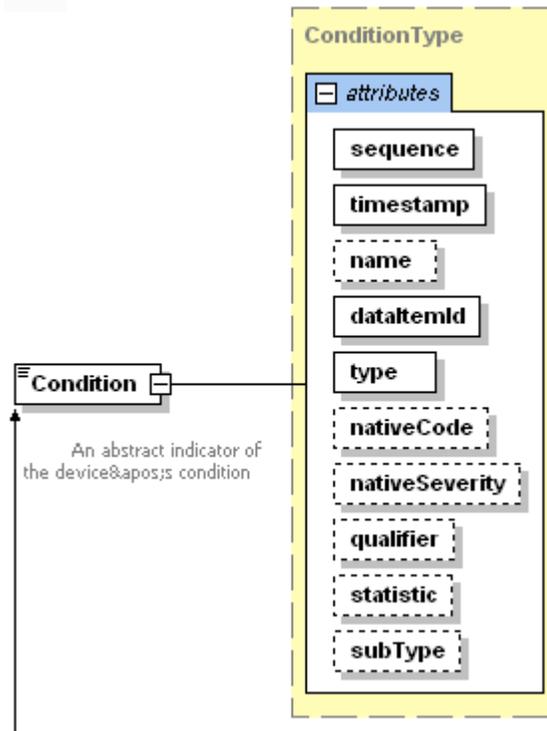
612 **Fault:** The item has failed and intervention is required to return to a Normal condition.
613 Transition to a Normal condition indicates that the Fault condition has been cleared.
614 A Fault condition is something that always needs to be acknowledged before operation
615 can continue.

616

617 **Unavailable:** The value of the item is in an indeterminate state since the data source is
618 no longer providing data. This will also be the initial state of the Condition before a
619 connection is established with the data source. The Condition **MUST** be
620 Unavailable when the value is unknown.

621 5.8.1 XML Schema Structure for Condition

622 The following XML tree represents the structure of a Condition XML element showing the
623 attributes defined for Condition type Data Elements.



624

625

626

627

Figure 8: Condition Schema Diagram

628 5.8.2 Condition Attributes

629 The following table defines the attributes that may be used to provide additional information for
 630 a Condition type Data Element.

631

Attribute	Description	Occurrence
sequence	A number representing the sequential position of an occurrence of the Condition in the data buffer of the Agent. The value MUST be represented as an unsigned 64 bit with valid values from 1 to 2 ⁶⁴ -1.	1
timestamp	The time the data for the Condition was reported. The most accurate time available to the device MUST be used for the timestamp.	1
dataItemID	The unique identifier for the Condition. The dataItemID MUST match the id attribute of the data item defined in the Device Information Model that this Condition element represents.	1
name	The name of the Condition. The name MUST match the name of the data item defined in the Device Information Model that this Condition element represents. An NMTOKEN XML type.	0..1
type	The type of the DataItem in the Device Information Model that this Condition element represents.	1
subType	The subType of the DataItem defined in the Device Information Model that this Condition element represents.	0..1
qualifier	Qualifies the Condition and adds context or additional clarification. This optional attribute can be used to convey information such as HIGH or LOW type Warning and Fault condition to indicate differing types of condition states.	0..1
nativeCode	The native code (usually an alpha-numeric value) generated by the controller of a piece of equipment providing a reference identifier for a condition state or alarm. This is the same information an operator or maintenance personnel would see as a reference code designating a specific type of Condition when viewed at the piece of equipment. Usually this reference code is used to point to a more detailed description of the Condition.	0..1
nativeSeverity	If the data source assigns a severity level to a Condition, nativeSeverity is used to report that severity information to a client software application.	0..1
statistic	The type of statistical calculation specified for the DataItem defined in the Device Information Model that this Condition element represents.	0..1

Attribute	Description	Occurrence
xs:lang	An optional attribute that specifies the language of the alarm or condition text. Refer to IETF RFC 4646 (http://www.ietf.org/rfc/rfc4646.txt) or successor for a full definition of the values for this attribute. Does not appear in the Header schema diagrams.	0..1

632

633 Most `Condition` type Data Elements representing `SAMPLE` type data items are reporting
634 information about a `WARNING` or `FAULT` associated with a continuously variable numeric data
635 value. In this case, it is important to indicate whether the reported value is above or below the
636 expected range. These differences are defined by the `qualifier` attribute. As an example,
637 `Condition` for an `AMPERAGE` type data item may differentiate between a `HIGH` amperage and
638 a `LOW` amperage by using the `qualifier` attribute.

639

640 For these data items, there are five possible `Condition` states:

- 641 `FAULT, LOW`
- 642 `WARNING, LOW`
- 643 `NORMAL`
- 644 `WARNING, HIGH`
- 645 `FAULT, HIGH`

646 A `Condition` type Data Elements **MAY** contain `CDATA` as part of the content between the
647 element tags to provide additional information further defining the meaning of the `Condition`
648 being reported. In this case, the `CDATA` will be a text string providing that additional
649 descriptive information. An example of a `CONDITION` category data item reporting its state as
650 `WARNING` and providing additional information defining the meaning of the `Condition` type
651 data element is formatted as follows in the XML document:

```
652     <Warning type="AMPERAGE" id="amplp" qualifier="high" sequence="39320"  
653         timestamp="2013-11-13T08:00:00">"Lube Pump - High Current"/>
```

654 (Note: The highlighted items are identified for emphasis only)

655 In this example, the “Lube Pump - High Current” is the `CDATA` for the `Condition` representing the
656 `AMPERAGE` type data item.

657 5.8.3 `Condition` Type Data Elements

658 All `DataItem` types in the `SAMPLE` category **MAY** have associated `CONDITION` states.
659 These `CONDITION` states are reported in the `MTConnectStreams` document as `Condition`
660 type Data Elements.

661 While all `DataItem` types in the `EVENT` category **MAY** have associated `CONDITION` states,
662 many typically will not have associated `CONDITION` states based on the type of data that they
663 represent.

664 Other `Condition` type Data Elements represent `CONDITION` type data items that have been
 665 defined to represent the health and fault status of Structural Elements. These `Condition` type
 666 elements are unlike any other data types since they **MAY** have multiple concurrently active
 667 values at any point in time. Each occurrence of one of these `Condition` types represents a
 668 separate and unique `Fault (Condition)`. Each `Fault` must be identified and tracked
 669 individually in the `MTConnectStreams` document. For example, if there are multiple errors
 670 detected in a part program for a CNC controller, each must be identified and tracked until it is
 671 corrected. The program will only execute once all of the errors have been cleared.

672 The following table lists those `Condition` types that have been defined to represent the health
 673 and fault status of Structural Elements and those associated with `EVENT` type data items that
 674 typically will have associated `CONDITION` states.

675

Data Element <code>CONDITION</code>	Response Format <code>Condition</code>	Description
<code>ACTUATOR</code>	<code>ACTUATOR</code>	An actuator's status.
<code>CHUCK_INTERLOCK</code>	<code>CHUCK_INTERLOCK</code>	An indication of the operational condition of the interlock function for an electronically controller chuck.
<code>COMMUNICATIONS</code>	<code>COMMUNICATIONS</code>	A communications failure indicator.
<code>DATA_RANGE</code>	<code>DATA_RANGE</code>	Information provided is outside of expected value range
<code>DIRECTION</code>	<code>DIRECTION</code>	An indication of a fault associated with the direction of motion of a Structural Element
<code>END_OF_BAR</code>	<code>END_OF_BAR</code>	An indication that the end of a piece of bar stock has been reached.
<code>HARDWARE</code>	<code>HARDWARE</code>	The hardware subsystem of the Structural Element's operation condition.
<code>INTERFACE_STATE</code>	<code>INTERFACE_STATE</code>	An indication of the operation condition of an Interface.
<code>LOGIC_PROGRAM</code>	<code>LOGIC_PROGRAM</code>	An indication that an error has occurred in the logic program or PLC (programmable logic controller) associated with a <code>Controller</code> component.
<code>MOTION_PROGRAM</code>	<code>MOTION_PROGRAM</code>	An indication that an error has occurred in the motion program associated with a <code>Controller</code> component.
<code>SYSTEM</code>	<code>SYSTEM</code>	A <code>CONDITION</code> representing something that is not the operator, program, or hardware.

676

677

Appendices

678

679

A. Bibliography

680

681

682

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

683

684

685

686

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

687

688

689

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

690

691

692

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

693

694

695

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

696

697

698

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

699

700

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

701

702

703

704

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

705

706

707

708

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

709

710

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

711

712

713

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

- 714 12. ASME B5.57: *Methods for Performance Evaluation of Computer Numerically Controlled*
715 *Lathes and Turning Centers*, 1998
- 716 13. ASME/ANSI B5.54: *Methods for Performance Evaluation of Computer Numerically*
717 *Controlled Machining Centers*. 2005.
- 718 14. OPC Foundation. *OPC Unified Architecture Specification, Part 1: Concepts Version 1.00.*
719 *July 28, 2006.*
- 720 15. IEEE STD 1451.0-2007, *Standard for a Smart Transducer Interface for Sensors and*
721 *Actuators – Common Functions, Communication Protocols, and Transducer Electronic*
722 *Data Sheet (TEDS) Formats*, IEEE Instrumentation and Measurement Society, TC-9, The
723 *Institute of Electrical and Electronics Engineers, Inc., New York, N.Y. 10016, SH99684,*
724 *October 5, 2007.*
- 725 16. IEEE STD 1451.4-1994, *Standard for a Smart Transducer Interface for Sensors and*
726 *Actuators – Mixed-Mode Communication Protocols and Transducer Electronic Data*
727 *Sheet (TEDS) Formats*, IEEE Instrumentation and Measurement Society, TC-9, The
728 *Institute of Electrical and Electronics Engineers, Inc., New York, N.Y. 10016, SH95225,*
729 *December 15, 2004.*
- 730