



MTConnect[®] Standard
Part 3.0 – Streams Information Model
Version 1.6.0

Prepared for: MTConnect Institute
Prepared on: July 15, 2020

MTConnect[®] is a registered trademark of AMT - The Association for Manufacturing Technology. Use of *MTConnect* is limited to use as specified on <http://www.mtconnect.org/>.

MTConnect Specification and Materials

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

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

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

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

Table of Contents

1	Purpose of This Document	2
2	Terminology and Conventions	3
2.1	Glossary	3
2.2	Acronyms	10
2.3	MTCConnect References	11
3	Streams Information Model	12
4	Structural Elements for MTCConnectStreams	15
4.1	Streams	17
4.2	DeviceStream	19
4.2.1	XML Schema for DeviceStream	19
4.2.2	Attributes for DeviceStream	20
4.2.3	Elements for DeviceStream	21
4.3	ComponentStream	22
4.3.1	XML Schema for ComponentStream	22
4.3.2	Attributes for ComponentStream	23
4.3.3	Elements for ComponentStream	27
5	Data Entities	28
5.1	Element Names for Data Entities	30
5.1.1	Element Names when MTCConnectDevices category is SAMPLE or EVENT	30
5.1.2	Changes to Element Names when representation attribute is used .	31
5.1.3	Element Names when MTCConnectDevices category is CONDITION	31
5.2	Samples Container	32
5.3	Sample Data Entities	32
5.3.1	XML Schema Structure for Sample	33
5.3.2	Attributes for Sample	34
5.3.2.1	duration Attribute for Sample	38
5.3.2.2	resetTriggered Attribute for Sample	38
5.3.3	Valid Data Values for Sample	39
5.3.4	Unavailability of Valid Data Values for Sample	41
5.4	Events Container	41
5.5	Event Data Entities	42
5.5.1	XML Schema Structure for Event	43
5.5.2	Attributes for Event	44
5.5.3	Valid Data Values for Event	45
5.5.4	Unavailability of Valid Data Value for Event	47
5.6	Representations	47

5.6.1	Observations for DataItem with representation of TIME_SERIES	48
5.6.1.1	XML Schema for Time Series Observation	48
5.6.1.2	Attributes for Time Series Observation	50
5.6.2	Observations for DataItem with representation of DISCRETE (DEP- RECATED)	50
5.6.3	Observations for DataItem with representation of DATA_SET . .	50
5.6.3.1	XML Schema for Data Set Observation	51
5.6.3.2	Entry Element for Data Set Observation	52
5.6.3.3	Attributes for Entry Element for Data Set Observation .	53
5.6.3.4	Constraints for Entry Values	53
5.6.4	Management of Data Set Observations	54
5.6.5	Observations for DataItem with representation of TABLE	54
5.6.5.1	Structure of Table Observations	55
5.6.5.2	Attributes of Table Observations	57
5.6.5.3	Elements of Table Observations	57
5.6.5.3.1	Structure for Table Entry for an Observation .	57
5.6.5.3.2	Attributes for Table Entry for an Observation .	57
5.6.5.3.3	Elements for Table Cell for an Observation . .	58
5.6.5.3.4	Structure for Table Cell for an Entry	58
5.6.5.3.5	Attributes for Table Cell for an Observation . .	58
5.6.5.3.6	Constraints for Cell Values	58
5.6.5.3.7	Example Table Observation	59
5.7	Condition Container	59
5.8	Condition Data Entity	60
5.8.1	Element Names for Condition	61
5.8.2	XML Schema Structure for Condition	62
5.8.3	Attributes for Condition	63
5.8.3.1	qualifier Attribute for Condition	66
5.8.4	Valid Data Value for Condition	67
5.9	Unavailability of Fault State for Condition	67
6	Listing of Data Entities	69
6.1	Sample Element Names	69
6.2	Event Element Names	93
6.3	Types of Condition Elements	135
	Appendices	137
A	Bibliography	137

Table of Figures

Figure 1: Streams Data Structure	16
Figure 2: Streams Schema Diagram	18
Figure 3: DeviceStream Schema Diagram	20
Figure 4: ComponentStream Schema Diagram	23
Figure 5: ComponentStream XML Tree Diagram	28
Figure 6: Sample Schema Diagram	34
Figure 7: Event Schema Diagram	43
Figure 8: AbsTimeSeries Schema Diagram	49
Figure 9: Sample Data Set Schema Diagram	51
Figure 10:Entry Element Schema Diagram	52
Figure 11:Table Schema Diagram	56
Figure 12:Condition Schema Diagram	62

List of Tables

Table 1: MTConnect Streams Element	18
Table 2: MTConnect DeviceStream Element	19
Table 3: Attributes for DeviceStream	20
Table 4: Elements for DeviceStream	21
Table 5: Attributes for ComponentStream	24
Table 6: Elements for ComponentStream	27
Table 7: MTConnect Samples Element	32
Table 8: MTConnect Sample Element	33
Table 9: Attributes for Sample	34
Table 10: Values for resetTriggered	39
Table 11: MTConnect Event Element	42
Table 12: MTConnect Event Element	43
Table 13: Attributes for Event	44
Table 14: Attributes for Time Series Observation	50
Table 15: Attributes for Data Set Observation	51
Table 16: Elements for Data Set Observation	52
Table 17: Attributes for Entry	53
Table 18: Attributes for Table	57
Table 19: Elements for Table	57
Table 20: Elements for Table Cell	58
Table 21: Attributes for Table Cell	58
Table 22: MTConnect Condition Element Container	60
Table 23: MTConnect Condition Element	61
Table 24: Attributes for Condition	63
Table 25: Element Names for Sample	69
Table 26: Element Names for Event	94
Table 27: Element Names for Condition	136

1 **1 Purpose of This Document**

2 This document, *MTConnect Standard: Part 3.0 - Streams Information Model* of the MT-
3 Connect Standard, establishes the rules and terminology that describes the information
4 returned by an *MTConnect Agent* from a piece of equipment. The *Streams Information*
5 *Model* also defines, in *Section 3 - Streams Information Model*, the structure for the XML
6 documents that are returned from an *Agent* in response to a *Sample Request* or *Current*
7 *Request*.

8 *MTConnect Standard: Part 3.0 - Streams Information Model* is not a stand-alone docu-
9 ment. This document is used in conjunction with *MTConnect Standard Part 1.0 - Overview*
10 *and Fundamentals* which defines the fundamentals of the operation of the *MTConnect*
11 *Standard* and *MTConnect Standard: Part 2.0 - Devices Information Model* that defines
12 the semantic model representing the information that may be returned from a piece of
13 equipment.

14 Note: *MTConnect Standard: Part 5.0 - Interfaces* provides details on extensions to
15 the *Streams Information Model* required to describe the interactions between pieces of
16 equipment.

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

20 **2 Terminology and Conventions**

21 Refer to *Section 3 of MTConnect Standard Part 1.0 - Overview and Fundamentals* for a
22 dictionary of terms, reserved language, and document conventions used in the MTConnect
23 Standard.

24 **2.1 Glossary**

25 **CDATA**

26 General meaning:

27 An abbreviation for Character Data.

28 CDATA is used to describe a value (text or data) published as part of an XML ele-
29 ment.

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

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

32 Appears in the documents in the following form: CDATA

33 **HTTP**

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

36 Note: HTTP is an IETF standard and is defined in RFC 7230.

37 See <https://tools.ietf.org/html/rfc7230> for more information.

38 **NMTOKEN**

39 The data type for XML identifiers.

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

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

44 **XML**

45 Stands for eXtensible Markup Language.

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

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

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

50 ***Agent***

51 Refers to an MTConnect Agent.

52 Software that collects data published from one or more piece(s) of equipment, orga-
53 nizes that data in a structured manner, and responds to requests for data from client
54 software systems by providing a structured response in the form of a *Response Doc-*
55 *ument* that is constructed using the *semantic data models* defined in the Standard.

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

57 ***Asset Document***

58 An electronic document published by an *Agent* in response to a *Request* for infor-
59 mation from a client software application relating to Assets.

60 ***Child Element***

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

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

64 ***Component***

65 General meaning:

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

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

69 Used in *Information Models*:

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

- 72 • When used as an XML container to organize *Lower Level* Component ele-
73 ments.

74 Appears in the documents in the following form: *Component s*.

- 75 • When used as an abstract XML element. *Component* is replaced in a data
76 model by a type of *Component* element. *Component* is also an XML con-
77 tainer used to organize *Lower Level* Component elements, *Data Entities*, or
78 both.

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

80 ***Condition***

81 General meaning:

82 An indicator of the health of a piece of equipment or a *Component* and its ability to
83 function.

84 Used as a modeling element:

85 A data modeling element used to organize and communicate information relative to
86 the health of a piece of equipment or *Component*.

87 Appears in the documents in the following form: *Condition*.

88 Used in *Information Models*:

89 An XML element used to represent *Condition* elements.

90 ● When used as an XML container to organize *Lower Level Condition* ele-
91 ments.

92 Appears in the documents in the following form: *Condition*.

93 ● When used as a *Lower Level* element, the form *Condition* is an abstract
94 type XML element. This *Lower Level* element is a *Data Entity*. *Condition*
95 is replaced in a data model by type of *Condition* element.

96 Appears in the documents in the following form: *Condition*.

97 Note: The form *Condition* is used to represent both above uses.

98 ***Controlled Vocabulary***

99 A restricted set of values that may be published as the *Valid Data Value* for a *Data*
100 *Entity*.

101 Appears in the documents in the following form: *Controlled Vocabulary*.

102 ***Current Request***

103 An HTTP request to the *Agent* for returning latest known values for the *DataItem*
104 as an *MTConnectStreams* XML document

105 ***Data Entity***

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

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

110 ***Data Set***

111 A set of *key-value pairs* where each entry is uniquely identified by the *key*.

112 ***Devices Information Model***

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

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

116 ***Document***

117 General meaning:

118 A piece of written, printed, or electronic matter that provides information.

119 Used to represent an *MTCConnect Document*:

120 Refers to printed or electronic document(s) that represent a *Part(s)* of the MTCConnect
121 Standard.

122 Appears in the documents in the following form: *MTCConnect Document*.

123 Used to represent a specific representation of an *MTCConnect Document*:

124 Refers to electronic document(s) associated with an *Agent* that are encoded using
125 XML; *Response Documents* or *Asset Documents*.

126 Appears in the documents in the following form: *MTCConnect XML Document*.

127 Used to describe types of information stored in an *Agent*:

128 In an implementation, the electronic documents that are published from a data source
129 and stored by an *Agent*.

130 Appears in the documents in the following form: *Asset Document*.

131 Used to describe information published by an *Agent*:

132 A document published by an *Agent* based upon one of the *semantic data models*
133 defined in the MTCConnect Standard in response to a request from a client.

134 Appears in the documents in the following form: *Response Document*.

135 ***Element Name***

136 A descriptive identifier contained in both the `start-tag` and `end-tag` of an
137 XML element that provides the name of the element.

138 Appears in the documents in the following form: `element name`.

139 Used to describe the name for a specific XML element:

140 Reference to the name provided in the `start-tag`, `end-tag`, or `empty-element`
141 `tag` for an XML element.

142 Appears in the documents in the following form: *Element Name*.

143 ***Equipment Metadata***

144 See *Metadata*

145 ***Fault State***

146 In the MTCConnect Standard, a term that indicates the reported status of a *Condition*
147 category *Data Entity*.

148 Appears in the documents in the following form: *Fault State*.

149 ***Information Model***

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

152 For example, an information model is used to define the structure for each *MTCCon-*
153 *nect Response Document*; the definition of each piece of information within those
154 documents and the relationship between pieces of information.

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

156 ***Interaction Model***

157 The definition of information exchanged to support the interactions between pieces
158 of equipment collaborating to complete a task.

159 Appears in the documents in the following form: *Interaction Model*.

160 ***Interface***

161 General meaning:

162 The exchange of information between pieces of equipment and/or software systems.

163 Appears in the documents in the following form: interface.

164 Used as an *Interaction Model*:

165 An *Interaction Model* that describes a method for inter-operations between pieces
166 of equipment.

167 Appears in the documents in the following form: *Interface*.

168 Used as an XML container or element:

169 - When used as an XML container that consists of one or more types of Inter-
170 face XML elements.

171 Appears in the documents in the following form: *Interfaces*.

172 - When used as an abstract XML element. It is replaced in the XML document
173 by types of *Interface* elements.

174 Appears in the documents in the following form: *Interface*

175 ***key***

176 A unique identifier in a *key-value pair* association.

177 key-value pair

178 An association between an identifier referred to as the *key* and a value which taken
179 together create a *key-value pair*. When used in a set of *key-value pairs* each *key* is
180 unique and will only have one value associated with it at any point in time.

181 Lower Level

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

183 Metadata

184 Data that provides information about other data.

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

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

190 MTConnect Document

191 See *Document*.

192 MTConnect XML Document

193 See *Document*.

194 observation

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

196 Parent Element

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

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

200 Request

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

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

204 reset

205 A reset is associated with an occurrence of a *Data Entity* indicated by the `reset-`
206 `Triggered` attribute. When a reset occurs, the accumulated value or statistic are
207 reverted back to their initial value. A *Data Entity* with a *Data Set* representation
208 removes all *key-value pairs*, setting the *Data Set* to an empty set.

209 ***Response Document***

210 See *Document*.

211 ***Sample Request***

212 A request from the *Agent* for a stream of time series data.

213 ***semantic data model***

214 A methodology for defining the structure and meaning for data in a specific logical
215 way.

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

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

219 ***sequence number***

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

222 *sequence number* is a monotonically increasing number within an instance of an
223 *Agent*.

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

225 ***Streaming Data***

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

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

229 ***Streams Information Model***

230 The rules and terminology (*semantic data model*) that describes the *Streaming Data*
231 returned by an *Agent* from a piece of equipment in response to a *Sample Request* or
232 a *Current Request*.

233 Appears in the documents in the following form: *Streams Information Model*.

234 ***Structural Element***

235 General meaning:

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

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

239 Used to indicate hierarchy of Components:

240 When used to describe a primary physical or logical construct within a piece of
241 equipment.

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

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

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

246 **Table**

247 A two dimensional set of values given by a set of *key-value pairs Table Entries*.
248 Each *Table Entry* contains a set of *key-value pairs of Table Cells*. The `Entry` and
249 `Cell` elements comprise a tabular representation of the information.

250 **Table Cell**

251 A subdivision of a *Table Entry* representing a singular value.

252 **Table Entry**

253 A subdivision of a *Table* containing a set of *key-value pairs* representing *Table Cells*.

254 **Top Level**

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

257 **Valid Data Value**

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

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

261 **XML Schema**

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

264 **2.2 Acronyms**

265 **AMT**

266 The Association for Manufacturing Technology

267 **2.3 MTConnect References**

- 268 [MTConnect Part 1.0] *MTConnect Standard Part 1.0 - Overview and Fundamentals*. Ver-
269 sion 1.5.0.
- 270 [MTConnect Part 2.0] *MTConnect Standard: Part 2.0 - Devices Information Model*. Ver-
271 sion 1.5.0.
- 272 [MTConnect Part 3.0] *MTConnect Standard: Part 3.0 - Streams Information Model*. Ver-
273 sion 1.5.0.
- 274 [MTConnect Part 5.0] *MTConnect Standard: Part 5.0 - Interfaces*. Version 1.5.0.

275 3 Streams Information Model

276 The *Streams Information Model* provides a representation of the data reported by a piece
 277 of equipment used for a manufacturing process, or used for any other purpose. Additional
 278 descriptive information associated with the reported data is defined in the *MTConnect-*
 279 *Devices* document, which is described in *MTConnect Standard: Part 2.0 - Devices*
 280 *Information Model*.

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

- 286 • `sample`: Returns a designated number of time stamped *Data Entities* from an *Agent*
 287 associated with a piece of equipment; subject to any HTTP filtering associated with
 288 the request. See *Section 8.3.3* of *MTConnect Standard Part 1.0 - Overview and Fun-*
 289 *damentals* of the *MTConnect Standard* for details on the `sample` HTTP request.
- 290 • `current`: Returns a snapshot of either the most recent values or the values at a
 291 given sequence number for all *Data Entities* associated with a piece of equipment
 292 from an *Agent*; subject to any HTTP filtering associated with the request. See *Sec-*
 293 *tion 8.3.2* of *MTConnect Standard Part 1.0 - Overview and Fundamentals* of the
 294 *MTConnect Standard* for details on the `current` HTTP request.

295 An *Agent* responds to either the `sample` or `current` HTTP request with an
 296 *MTConnectStreams XML* document. This document contains information describing
 297 *Data Entities* reported by an *Agent* associated with a piece of equipment. A client software
 298 application may correlate the information provided in the *MTConnectStreams XML*
 299 document with the physical and logical structure for that piece of equipment defined in the
 300 *MTConnectDevices* document to form a clear and unambiguous understanding of the
 301 information provided. (See details on the structure for a piece of equipment described in
 302 *MTConnect Standard: Part 2.0 - Devices Information Model*).

303 The *MTConnectStreams XML* document is comprised of two sections: `Header` and
 304 `Streams`.

305 The `Header` section contains protocol related information as defined in *Section 6.5* of
 306 *MTConnect Standard Part 1.0 - Overview and Fundamentals* of the *MTConnect Standard*.

307 The `Streams` section of the *MTConnectStreams* document contains a
 308 *DeviceStream XML* container for each piece of equipment represented in the docu-

309 ment. Each `DeviceStream` container is comprised of two primary types of XML ele-
 310 ments – *Structural Elements* and *Data Entities*. The contents of the `DeviceStream` con-
 311 tainer are described in detail in this document, *MTConnect Standard: Part 3.0 - Streams*
 312 *Information Model* of the MTConnect Standard.

313 *Structural Elements* are defined for both the `MTConnectDevices` and the `MTCon-`
 314 `nectStreams` XML documents. These *Structural Elements* are used to provide a logi-
 315 cal organization of the information provided in each document. While used for a similar
 316 purpose, the *Structural Elements* in the `MTConnectStreams` document are specifically
 317 designed to be distinctly different from those in the `MTConnectDevices` document:

- 318 • `MTConnectDevices` document: *Structural Elements* organize information that
 319 represents the physical and logical parts and sub-parts of a piece of equipment. (See
 320 *MTConnect Standard: Part 2.0 - Devices Information Model*, Section 4 of the MT-
 321 Connect Standard for more details on *Structural Elements* used in the `MTConnect-`
 322 `Devices` document).

- 323 • `MTConnectStreams` document: *Structural Elements* provide the structure to or-
 324 ganize the data returned from a piece of equipment and establishes the proper context
 325 for that data. The *Structural Elements* specifically defined for use in the `MTCon-`
 326 `nectStreams` document are `DeviceStream` (see *Section 4.2 - DeviceStream*)
 327 and `ComponentStream` (see *Section 4.3 - ComponentStream*).

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

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

- 334 • `MTConnectDevices` document: *Data Entity* elements define the data that may
 335 be returned from a piece of equipment. *MTConnect Standard: Part 2.0 - Devices*
 336 *Information Model*, Sections 7 and 8 lists the possible *Data Entity* XML elements
 337 that can be returned in a `MTConnectDevices` document.

- 338 • `MTConnectStreams` document: *Data Entity* elements provide the data reported
 339 by a piece of equipment. This data is organized in separate `ComponentStream`
 340 XML containers for each of the *Structural Elements* defined in the `MTConnectDe-`
 341 `vices` document associated with the data that is reported by a piece of equipment.

342 Within each `ComponentStream` XML container in the `MTConnectStreams` docu-
343 ment, *Data Entities* are organized into three types of XML container elements - `Samples`,
344 `Events`, and `Conditions`. (See *Section 5 - Data Entities* and *Section 6 - Listing of*
345 *Data Entities* for more information on these elements.)

346 4 Structural Elements for MTConnectStreams

347 *Structural Elements* are XML elements that form the logical structure for the MTCon-
348 nectStreams XML document. These elements are used to organize the information
349 and data that is reported by an *Agent* for a piece of equipment. See *Figure 1* for an
350 overview of the *Structural Elements* used in an MTConnectStreams document.

351 The first, or highest level, *Structural Element* in an MTConnectStreams XML docu-
352 ment is *Streams*. *Streams* is a container type XML element used to group the data
353 reported from one or more pieces of equipment into a single XML document. *Streams*
354 **MUST** always appear in the MTConnectStreams document.

355 *DeviceStream* is the next *Structural Element* in the MTConnectStreams document.
356 *DeviceStream* is also a XML container type element. A separate *DeviceStream*
357 container is used to organize the information and data reported by each piece of equip-
358 ment represented in the MTConnectStreams document. There **MUST** be at least one
359 *DeviceStream* element in the *Streams* container.

360 A *DeviceStream* element provides the data reported by a piece of equipment. Each
361 *DeviceStream* element **MUST** contain the attributes *name* and *uuid* to correlate the
362 *DeviceStream* with a specific *Device* defined in the MTConnectDevices docu-
363 ment. Once the *DeviceStream* element is associated with a specific piece of equipment
364 based on this identity, all data reported by that piece of equipment is directly associated
365 with that unique identity and that association does not need to be repeated for every piece
366 of data reported. A client software application may then directly relate the information
367 provided in the MTConnectDevices document with the data provided in the MTCon-
368 nectStreams document based on this identity.

369 *ComponentStream* is the next level XML element in the MTConnectStreams docu-
370 ment. *ComponentStream* is also a container type XML element. There **MUST** be
371 a separate *ComponentStream* XML element for each of the *Structural Elements* (*De-*
372 *vice elements, Top Level Component elements, or Lower Level Component elements*)
373 defined for that piece of equipment in the associated MTConnectDevices XML docu-
374 ment. A *ComponentStream* representing a *Structural Element* will only appear if there
375 is data reported for that *Structural Element*. (Note: See *MTConnect Standard: Part 2.0 -*
376 *Devices Information Model* of the MTConnect Standard for a description of the *Structural*
377 *Elements* for a piece of equipment).

378 There are three (3) *Structural Elements* – *Samples*, *Events*, and *Condition* at the
379 next level of the MTConnectStreams document. Each one of these *Structural Elements*
380 is a container type XML element. These *Structural Elements* group the data reported for
381 each component of a piece of equipment according to the *Data Entity* categories defined

382 in *MTConnect Standard: Part 2.0 - Devices Information Model*, Sections 7 and 8.

- 383 • Samples contains SAMPLE category *Data Entities* defined in the *MTConnect-*
384 *Devices XML* document (See *MTConnect Standard: Part 2.0 - Devices Informa-*
385 *tion Model*, Section 8.1)
- 386 • Events contains EVENT category *Data Entities* defined in the *MTConnectDe-*
387 *vices XML* document (See *MTConnect Standard: Part 2.0 - Devices Informa-*
388 *tion Model*, Section 8.2)
- 389 • Condition contains CONDITION category *Data Entities* defined in the *MTCon-*
390 *nectDevices XML* document (See *MTConnect Standard: Part 2.0 - Devices*
391 *Information Model*, Section 8.3)

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

394 *Figure 1* XML tree structure illustrates the various *Structural Elements* used to organize
395 the data reported by a piece of equipment and the relationship between these elements.

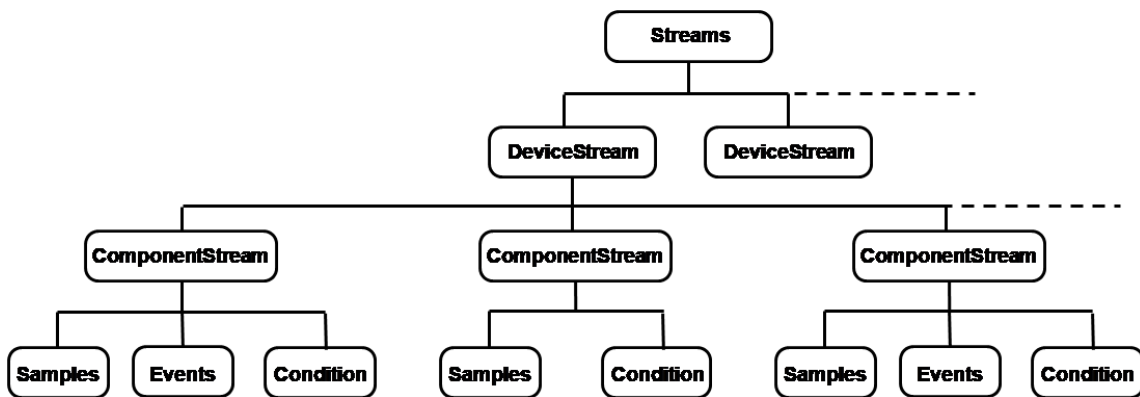


Figure 1: Streams Data Structure

396 *Example 1* is a sample from an *MTConnectStreams XML* document that contains the
397 response from an *Agent* representing two pieces of equipment, *mill-1* and *mill-2*. The data
398 from each piece of equipment is reported in a separate *DeviceStream* container.

Example 1: Example of DeviceStream

```

399 1 <MTConnectStreams ...>
400 2   <Header ... />
401 3   <Streams>
402 4     <DeviceStream name="mill-1" uuid="1">
403 5       <ComponentStream component="Device" name="mill-1">

```

```

404 6         componentId="d1">
405 7     <Events>
406 8         <Availability dataItemId="avail1" name="avail"
407 9             sequence="5"
408 10             timestamp="2010-04-06T06:19:35.153141">
409 11             AVAILABLE</Availability>
410 12     </Events>
411 13 </ComponentStream>
412 14 </DeviceStream>
413 15 <DeviceStream name="mill-2" uuid="2">
414 16     <ComponentStream component="Device" name="mill-2"
415 17         componentId="d2">
416 18     <Events>
417 19         <Availability dataItemId="avail2" name="avail"
418 20             sequence="15"
419 21             timestamp="2010-04-06T06:19:35.153141">
420 22             AVAILABLE</Availability>
421 23     </Events>
422 24     </ComponentStream>
423 25 </DeviceStream>
424 26 </Streams>
425 27 </MTConnectStreams>

```

426 In *Example 1*, it should be noted that the *sequence numbers* are unique across the two
427 pieces of equipment. Client software applications **MUST NOT** assume that the `Events`
428 and `Samples` sequence numbers are strictly in sequence. All sequence numbers **MAY**
429 **NOT** be included. For instance, such a case would occur when HTTP filtering is applied to
430 the request and the `SAMPLE`, `EVENT`, and `CONDITION` data types for other components
431 are not returned. Another case would occur when an *Agent* is supporting more than one
432 piece of equipment and data from only one piece of equipment is requested. Refer to MT-
433 Connect Standard *MTConnect Standard Part 1.0 - Overview and Fundamentals, Section 5*
434 for more information on *sequence numbers*.

435 4.1 Streams

436 `Streams` is a container type XML element that **MUST** contain only `DeviceStream`
437 elements. `Streams` **MAY** contain any number of `DeviceStream` elements. If there is
438 no data to be reported for a request for data, an `MTConnectStreams` document **MUST**
439 be returned with an empty `Streams` container. *Data Entities* **MAY NOT** be directly
440 associated with the `Streams` container.

441 The XML schema in *Figure 2* represents the structure of the `Streams` XML element.

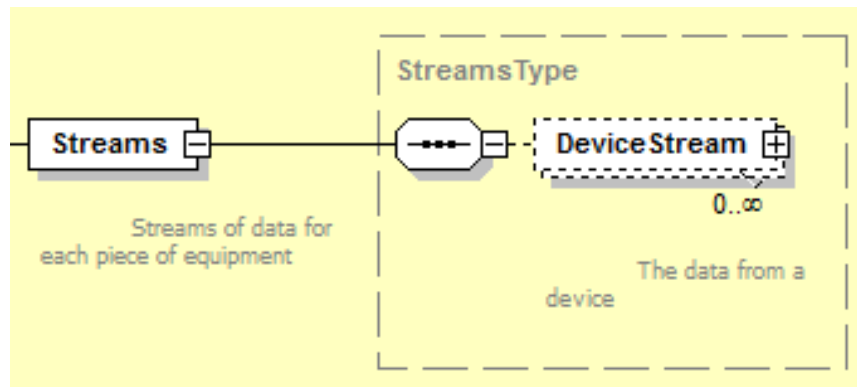


Figure 2: Streams Schema Diagram

Table 1: MTConnect Streams Element

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

442 4.2 DeviceStream

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

446 A DeviceStream **MAY** contain any number of ComponentStream elements; lim-
 447 ited to one for each component element represented in the MTConnectDevices doc-
 448 ument. If the response to the request for data from an *Agent* does not contain any data
 449 for a specific piece of equipment, an empty DeviceStream element **MAY** be created to
 450 indicate that the piece of equipment exists, but there was no data available. In this case,
 451 there will be no ComponentStream elements provided.

Table 2: MTConnect DeviceStream Element

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

452 4.2.1 XML Schema for DeviceStream

453 The XML schema in *Figure 3* represents the structure of the DeviceStream XML
 454 element showing the attributes defined for DeviceStream and the elements that **MAY**
 455 be associated with DeviceStream.

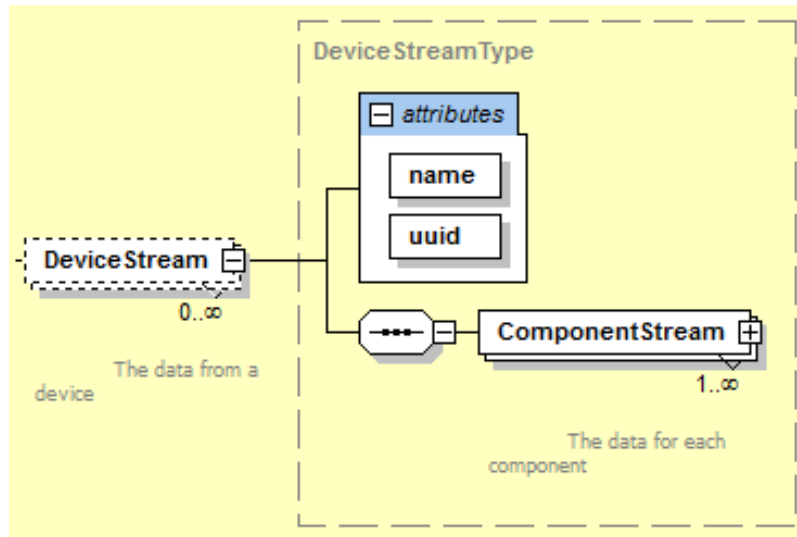


Figure 3: DeviceStream Schema Diagram

456 **4.2.2 Attributes for DeviceStream**

457 Table 3 defines the attributes that **MUST** be provided to uniquely identify each specific
 458 piece of equipment associated with the information provided in each DeviceStream.

Table 3: Attributes for DeviceStream

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

Continuation of Table 3		
Attribute	Description	Occurrence
uuid	<p>The uuid associated with the piece of equipment reporting the data contained in this DeviceStream container.</p> <p>uuid is a required attribute.</p> <p>The value reported for uuid MUST be the same as the value defined for the uuid attribute of the same piece of equipment in the MTConnectDevices document.</p>	1

459 4.2.3 Elements for DeviceStream

460 Table 4 lists the XML element(s) that **MAY** be provided in the DeviceStream XML
 461 element.

Table 4: Elements for DeviceStream

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

462 4.3 ComponentStream

463 ComponentStream is a XML container that organizes the data associated with each
464 *Structural Element* (Device element, *Top Level* Component, or *Lower Level* Com-
465 ponent element) defined for that piece of equipment in the associated MTConnectDe-
466 vices XML document. The data reported in each ComponentStream element **MUST**
467 be grouped into individual XML containers based on the value of the category attribute
468 (SAMPLE, EVENT, or CONDITION) defined for each *Data Entity* in the MTConnect-
469 Devices XML document. These containers are Samples, Events, and Condition.

470 4.3.1 XML Schema for ComponentStream

471 The XML schema in *Figure 4* represents the structure of a ComponentStream XML
472 element showing the attributes defined for ComponentStream and the elements that
473 **MAY** be associated with ComponentStream.

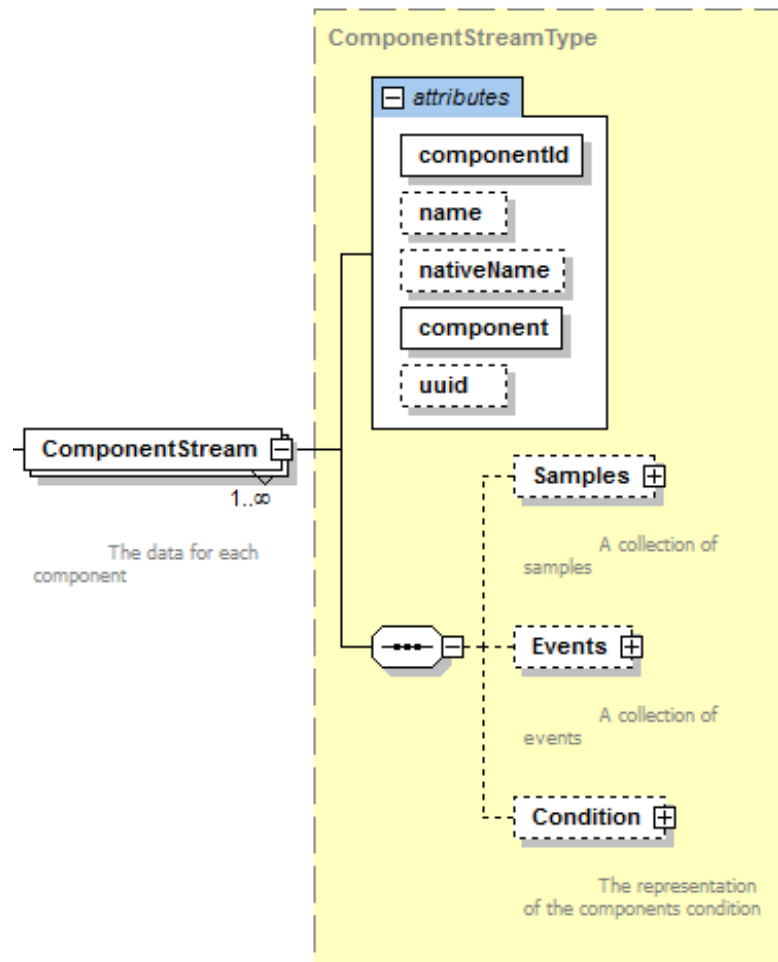


Figure 4: ComponentStream Schema Diagram

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

480 4.3.2 Attributes for ComponentStream

481 The *Table 5* defines the attributes used to uniquely identify the specific *Structural Element(s)* of a piece of equipment associated with the data reported in the MTConnect-Streams document.

Table 5: Attributes for ComponentStream

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

Continuation of Table 5		
Attribute	Description	Occurrence
nativeName	<p>nativeName identifies the common name normally associated with the ComponentStream element.</p> <p>nativeName is an optional attribute.</p> <p>If nativeName is not defined for a specific <i>Structural Element</i> in the MTConnectDevices document, it MUST NOT be provided for the corresponding ComponentStream element in the MTConnectStreams document.</p> <p>If nativeName is defined for a specific <i>Structural Element</i> in the MTConnectDevices document, it MAY be provided for the corresponding ComponentStream element in the MTConnectStreams document.</p> <p>If provided, the value reported for nativeName MUST be the same as the value defined for the nativeName attribute of the corresponding <i>Structural Element</i> (Device element, <i>Top Level Component</i> element, or <i>Lower Level Component</i> element) defined in the MTConnectDevices XML document.</p>	0..1

Continuation of Table 5		
Attribute	Description	Occurrence
component	<p>component identifies the <i>Structural Element</i> (Device, <i>Top Level Component</i>, or <i>Lower Level Component</i>) associated with the ComponentStream element.</p> <p>component is a required attribute.</p> <p>The value reported for component MUST be the same as the value defined for the Element Name of the XML container representing the corresponding <i>Structural Element</i> (Device element, <i>Top Level Component</i> element, or <i>Lower Level Component</i> element) defined in the MTConnectDevices XML document.</p> <p>Examples of Component are Device, Axes, Controller, Linear, Electric and Loader.</p>	1
uuid	<p>uuid of the ComponentStream element.</p> <p>uuid is an optional attribute.</p> <p>If uuid is not defined for a specific <i>Structural Element</i> in the MTConnectDevices document, it MUST NOT be provided for the corresponding ComponentStream element in the MTConnectStreams document.</p> <p>If uuid is defined for a specific <i>Structural Element</i> in the MTConnectDevices document, it MAY be provided for the corresponding ComponentStream element in the MTConnectStreams document, but it is not required.</p> <p>If provided, the value reported for uuid MUST be the same as the value defined for the uuid attribute of the corresponding <i>Structural Element</i> (Device element, <i>Top Level Component</i> element, or <i>Lower Level Component</i> element) defined in the MTConnectDevices XML document.</p>	0..1

484 4.3.3 Elements for ComponentStream

485 In the `ComponentStream` container, an *Agent* **MUST** organize the data reported in
 486 each `ComponentStream` into individual `Samples`, `Events`, or `Condition XML`
 487 containers based on the value of the `category` attribute (i.e., `SAMPLE`, `EVENT`, or `CON-`
 488 `DITION`) defined for each *Data Entity* defined in the `MTConnectDevices XML` doc-
 489 ument.

490 Each `ComponentStream` element **MUST** include at least one `Events`, `Samples`, or
 491 `Condition XML` container element. *Data Entities* returned in each of the `Compo-`
 492 `nentStream` container elements are defined in the *Table 6*.

Table 6: Elements for ComponentStream

Element	Description	Occurrence
<code>Samples</code>	An XML container type element. <code>Samples</code> organizes the <code>SAMPLE</code> type <i>Data Entities</i> defined in the <code>MTConnectDevices</code> document that are reported in each <code>ComponentStream XML</code> element.	0..1 †
<code>Events</code>	An XML container type element. <code>Events</code> organizes the <code>EVENT</code> type <i>Data Entities</i> defined in the <code>MTConnectDevices</code> document that are reported in each <code>ComponentStream XML</code> element.	0..1 †
<code>Condition</code>	An XML container type element. <code>Condition</code> organizes the <code>CONDITION</code> type <i>Data Entities</i> defined in the <code>MTConnectDevices</code> document that are reported in each <code>ComponentStream XML</code> element.	0..1 †

493 Note: †The `ComponentStream` element **MUST** contain at least one of these ele-
 494 ment types.

495 5 Data Entities

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

501 `DataItem` elements defined with a `category` attribute of `SAMPLE` in the `MTCon-`
 502 `nectDevices` document are mapped to the `Samples` XML container in the associated
 503 `ComponentStream` element.

504 `DataItem` elements defined with a `category` attribute of `EVENT` in the `MTCon-`
 505 `nectDevices` document are mapped to the `Events` XML container in the associated
 506 `ComponentStream` element.

507 `DataItem` elements defined with a `category` attribute of `CONDITION` in the `MT-`
 508 `ConnectDevices` document are mapped to the `Condition` XML container in the
 509 associated `ComponentStream` element.

510 The XML tree in *Figure 5* demonstrates how *Data Entities* are organized in these contain-
 511 ers.

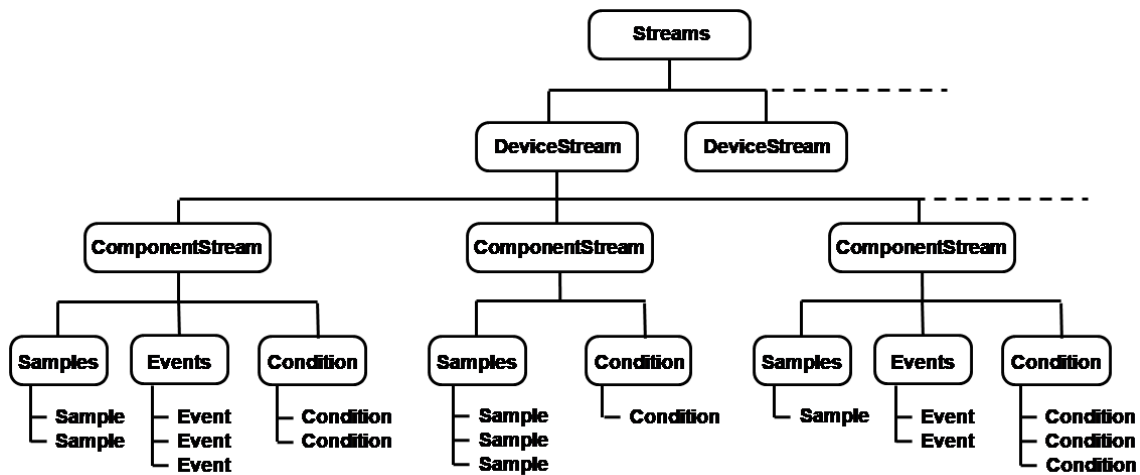


Figure 5: ComponentStream XML Tree Diagram

512 *Example 2* is an illustration of the structure of an XML document demonstrating how *Data*
 513 *Entities* are reported in a `MTConnectStreams` document:

Example 2: Example of MTConnectStreams

```

514 1 <MTConnectStreams>
515 2   <Header/>
516 3   <Streams>
517 4     <DeviceStream>
518 5       <ComponentStream>
519 6         <Samples>
520 7           <Sample/>
521 8           <Sample/>
522 9         </Samples>
523 10        <Events>
524 11          <Event/>
525 12          <Event/>
526 13        </Events>
527 14        <Condition>
528 15          <Condition/>
529 16          <Condition/>
530 17        </Condition>
531 18      </ComponentStream>
532 19      <ComponentStream>
533 20        <Samples>
534 21          <Sample/>
535 22          <Sample/>
536 23        </Samples>
537 24        <Events>
538 25          <Event/>
539 26          <Event/>
540 27        </Events>
541 28        <Condition>
542 29          <Condition/>
543 30          <Condition/>
544 31        </Condition>
545 32      </ComponentStream>
546 33    </DeviceStream>
547 34  </Streams>
548 35 </MTConnectStreams>

```

549 **Note:** There are no specific requirements defining the sequence in which the `ComponentStream` XML elements are organized in the `MTConnectStreams`
550 document. They **MAY** be organized in any sequence based on the implementation of an *Agent*. The sequence in which the `ComponentStream` XML
551 elements appear does not impact the ability for a client software application to
552 interpret the information that it receives in the document.
553
554

555 When an *Agent* responds to a current HTTP request, the information returned in the
556 `MTConnectStreams` document **MUST** include the most current value for every *Data*
557 *Entity* defined in the `MTConnectDevices` document subject to any filtering included
558 within the request.

559 When an *Agent* responds to a sample HTTP request, the information returned in the
 560 MTConnectStreams document **MUST** include the occurrences for each *Data Entity*
 561 that are available to an *Agent* subject to filtering and the count parameter included within
 562 the request (see *MTConnect Standard Part 1.0 - Overview and Fundamentals* for a full
 563 definition of the protocol).

564 5.1 Element Names for Data Entities

565 In the MTConnectDevices document, *Data Entities* are grouped as DataItem XML
 566 elements within each Device, *Top Level* Component, and *Lower Level* Component
 567 *Structural Element*. The *Data Entities* reported in the MTConnectStreams document
 568 associated with each of these *Structural Elements* are represented with an *Element Name*
 569 based on the category and type defined for each of the DataItem elements in the
 570 MTConnectDevices document.

571 5.1.1 Element Names when MTConnectDevices category is SAMPLE 572 or EVENT

573 The *Data Entities* reported in the MTConnectStreams document associated with each
 574 DataItem element defined in the MTConnectDevices document with a category
 575 attribute of SAMPLE or EVENT **MUST** be identified in the MTConnectStreams docu-
 576 ment with an *Element Name* derived from the type attribute defined for that DataItem
 577 element in the MTConnectDevices document.

578 The element name **MUST** derive from the DataItem type converted to *Pascal-Case*
 579 by removing underscores (`_`) and capitalizing each word. The conversion **MUST NOT**
 580 apply to the following abbreviated words: PH, AC, and DC.

581 *Example 3* describes the most common method used to derive the *Element Name* for a *Data*
 582 *Entity* reported in the MTConnectStreams document from the information describing
 583 that DataItem element in the MTConnectDevices document:

584 DataItem Represented in the MTConnectDevices Document

Example 3: DataItem Represented in MTConnectDevices Document

```
585 1 <DataItem type="AXIS_FEEDRATE" id="xf" name="Xfrt "  
586 2   category="SAMPLE" units="MILLIMETER/SECOND "  
587 3   nativeUnits="MILLIMETER/SECOND"/>
```

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

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

- 591 • type, category, units, and nativeUnits: Attributes that provide addi-
592 tional information regarding each data item in the MTConnectDevices docu-
593 ment.

594 Response Format reported in the MTConnectStreams Document

Example 4: Response Format reported in the MTConnectStreams Document

```
595 1 <AxisFeedrate name="Xfirt" sequence="61315517"
596 2     timestamp="2016-07-28T02:06:01.364428Z"
597 3     dataItemId="xf">10.83333</AxisFeedrate>
```

- 598 • AXIS_FEEDRATE: The *Element Name* provided in the MTConnectStreams re-
599 sponse format for the data item. The *Element Name* for a data item is defined by
600 the type attribute of AXIS_FEEDRATE in the MTConnectDevices document.
601 The *Element Name* **MUST** be provided in Pascal case format (first letter of each
602 word is capitalized).

603 5.1.2 Changes to Element Names when representation attribute is 604 used

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

608 5.1.3 Element Names when MTConnectDevices category is CONDI- 609 TION

610 *Data Entities* defined in the MTConnectDevices document with a category attribute
611 of CONDITION are reported with an *Element Name* that is defined differently from other
612 *Data Entity* types. The *Element Name* for these *Data Entities* are defined based on
613 the *Fault State* (Normal, Warning, or Fault) associated with each *Data Entity* at the
614 time that a value for that *Data Entity* is reported. See *Section 5.8.1 - Element Names for*
615 *Condition* and *Section 5.9 - Unavailability of Fault State for Condition* for details on how
616 these *Data Entities* are reported in the MTConnectStreams document.

617 5.2 Samples Container

618 `Samples` is a XML container type element. `Samples` organizes the *Data Entities* re-
619 turned in the `MTConnectStreams` XML document for those `DataItem` elements de-
620 fined with a `category` attribute of `SAMPLE` in the `MTConnectDevices` document.

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

Table 7: MTConnect Samples Element

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

624 5.3 Sample Data Entities

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

628 `Sample` is an abstract type XML element and will never appear directly in the `MTCon-`
629 `nectStreams` XML document. As an abstract type XML element, `Sample` will be
630 replaced in the XML document by a specific type of `Sample` specified by the *Element*
631 *Name* for that *Data Entity*. The different types of `Sample` elements are defined in
632 *Section 6.1 - Sample Element Names*. Examples of XML elements representing `Sample`
633 include `PathPosition`, `Temperature`.

Table 8: MTConnect Sample Element

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

634 5.3.1 XML Schema Structure for Sample

635 The XML schema in *Figure 6* represents the structure of a `Sample` XML element show-
 636 ing the attributes defined for `Sample` elements.

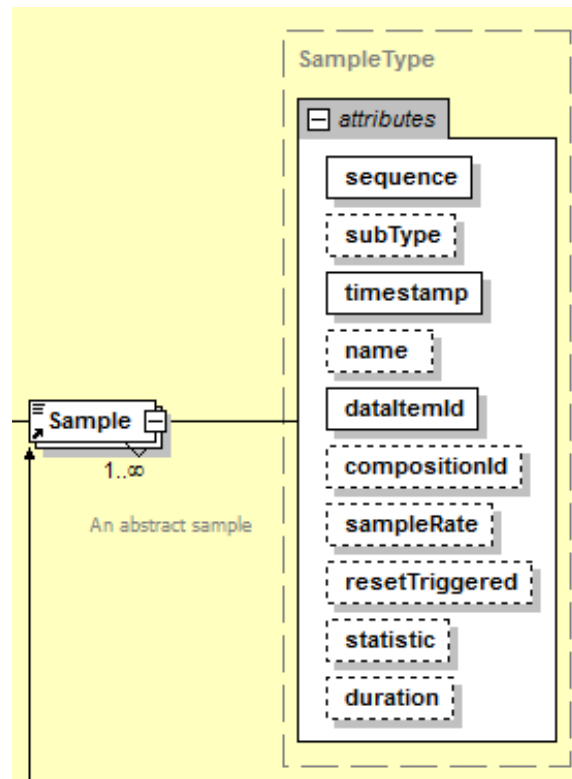


Figure 6: Sample Schema Diagram

637 **5.3.2 Attributes for Sample**

638 The *Table 9* defines the attributes used to provide additional information for a `Sample`
 639 XML element.

Table 9: Attributes for Sample

Attribute	Description	Occurrence
sequence	<p>A number representing the sequential position of an occurrence of the <code>Sample</code> in the data buffer of an <i>Agent</i>.</p> <p>sequence is a required attribute.</p> <p>sequence MUST have a value represented as an unsigned 64-bit value from 1 to $2^{64} - 1$.</p>	1

Continuation of Table 9		
Attribute	Description	Occurrence
subType	<p>The subType of the <i>Data Entity</i>.</p> <p>subType is an optional attribute.</p> <p>subType MUST match the subType attribute of the DataItem element as defined in the MTConnectDevices document that the Sample element represents.</p>	0..1
timestamp	<p>The most accurate time available to a piece of equipment that represents the point in time that the data reported for the Sample was measured.</p> <p>When the Sample element represents a DataItem element defined in the MTConnectDevices document with a representation or statistic attribute, timestamp MUST represent the time that the data collection was completed.</p> <p>timestamp is a required attribute.</p>	1
name	<p>The name of the Sample element.</p> <p>name is an optional attribute.</p> <p>name MUST match the name attribute of the DataItem element defined in the MTConnectDevices document that the Sample element represents.</p> <p>An NMTOKEN XML type.</p>	0..1
dataItemId	<p>The unique identifier for the Sample element.</p> <p>dataItemId is a required attribute.</p> <p>dataItemId MUST match the id attribute of the DataItem element defined in the MTConnectDevices document that the Sample element represents.</p>	1

Continuation of Table 9		
Attribute	Description	Occurrence
sampleRate	<p>The rate at which successive samples of the value of a data item are recorded. sampleRate is expressed in terms of samples per second.</p> <p>sampleRate is an optional attribute.</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>sampleRate MUST be provided when the representation attribute of the DataItem element defined in the MTConnectDevices document that this Sample element represents is TIME_SERIES.</p> <p>For DataItem elements where the representation attribute defined in the MTConnectDevices document that this Sample element represents is not TIME_SERIES, it MUST be assumed that the data reported is represented by a single value and sampleRate MUST NOT be reported in the MTConnectStreams document.</p>	0..1
statistic	<p>The type of statistical calculation defined by the statistic attribute of the DataItem element defined in the MTConnectDevices document that this Sample element represents.</p> <p>statistic is an optional attribute.</p>	0..1

Continuation of Table 9		
Attribute	Description	Occurrence
duration	<p>The time-period over which the data was collected.</p> <p>duration is an optional attribute.</p> <p>duration MUST be provided when the <code>theStatistic</code> attribute of the <code>DataItem</code> element is defined in the <code>MTConnectDevices</code> document that this <code>Sample</code> element represents.</p>	0..1
resetTriggered	<p>For those <code>DataItem</code> elements that report data that may be periodically reset to an initial value, <code>resetTriggered</code> identifies when a reported value has been reset and what has caused that reset to occur.</p> <p><code>resetTriggered</code> is an optional attribute.</p> <p><code>resetTriggered</code> MUST only be provided for the specific occurrence of a <i>Data Entity</i> reported in the <code>MTConnectStreams</code> document when the reset occurred and MUST NOT be provided for any other occurrence of the <i>Data Entity</i> reported in a <code>MTConnectStreams</code> document.</p>	0..1
compositionId	<p>The identifier of the <code>Composition</code> element defined in the <code>MTConnectDevices</code> document associated with the data reported for the <code>Sample</code> element.</p> <p><code>compositionId</code> is an optional attribute.</p>	0..1

640 5.3.2.1 duration Attribute for Sample

641 Sample elements that represent the result of a computed value of a statistic **MUST** con-
642 tain a `duration` attribute. For these *Data Entities*, the `timestamp` associated with
643 the Sample **MUST** reference the time the data collection was completed. `timestamp`
644 **MUST NOT** represent any other time associated with the data collection or the calcula-
645 tion of the statistic. The actual time the interval began can be computed by subtracting the
646 `duration` from the `timestamp`.

647 Two Sample elements **MAY** have overlapping time periods when statistics are computed
648 at different frequencies. For example, there may be two *Data Entities* reporting a statistic
649 representing the average value for the readings of the same measured signal calculated over
650 one and five minute intervals. These *Data Entities* can both have the same start time for
651 their calculations (e.g., 05:10:00), but the `timestamp` and `duration` will be 05:11:00
652 and 60 seconds, respectively, for the *Data Entity* reporting the one-minute average and
653 05:15:00 and 300 seconds, respectively, for the *Data Entity* reporting the five-minute av-
654 erage. This allows for varying statistical methods to be applied with different interval
655 lengths each having different values for the `timestamp` and `duration` attributes.

656 5.3.2.2 resetTriggered Attribute for Sample

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

662 `resetTriggered` defines the type of reset action that caused the value of the reported
663 data to be reset. The value reported for `resetTriggered` **MAY** be defined by the
664 `ResetTrigger` element for the *Data Entity* in the `MTConnectDevices` document
665 that this Sample element represents. If the `ResetTrigger` element is not defined in the
666 `MTConnectDevices` document, a `resetTriggered` attribute **SHOULD** be reported
667 in the `MTConnectStreams` document if the type of reset action can be determined and
668 reported by the piece of equipment.

669 `resetTriggered` **MUST** only be reported for the first occurrence of a *Data Entity*
670 after a reset action has occurred and **MUST NOT** be provided for any other occurrence
671 of the *Data Entity* reported in a `MTConnectStreams` document. When a reset occurs,
672 the piece of equipment **MUST** report an occurrence of the *Data Entity* that was reset even
673 if that occurrence of the *Data Entity* would normally be suppressed based on the filtering
674 criteria established in the `MTConnectDevices` document that this Sample element
675 represents.

676 The *Table 10* provides the values that **MAY** be reported for `resetTriggered`:

Table 10: Values for `resetTriggered`

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

677 5.3.3 Valid Data Values for Sample

678 All `Sample` elements reported in an `MtConnectStreams` XML document **MUST** provide a value in the CDATA of the *Data Entity*.
679

680 The value returned in the CDATA **MUST** be reported as either a *Valid Data Value* representing the information reported from a piece of equipment or `UNAVAILABLE` when a
681 *Valid Data Value* cannot be determined.
682

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

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

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

692 Note: If a representation attribute is not specified in the MTConnectDe-
693 vices document for a DataItem element, it **MUST** be assumed that the
694 data reported in the MTConnectStreams document for the *Data Entity* has
695 a representation type of VALUE.

696 In the case of a Sample element associated with a SAMPLE category DataItem element
697 with a representation attribute of TIME_SERIES, the data provided **MUST** be a
698 series of data values representing multiple sequential samples of the measured value that
699 will be provided only at the end of the completion of a sampling period. (See Section
700 *Section 5.6.1 - Observations for DataItem with representation of TIME_SERIES* for more
701 information on TIME_SERIES type data).

702 In the case of a Sample element associated with a SAMPLE category DataItem element
703 with a representation attribute of DATA_SET, the data reported for each *key-value*
704 *pair* **MUST** be provided in the same *Valid Data Values* and units as specified by the type
705 attribute for the DataItem element.

706 When an Agent responds to a *Current Request*, the information returned in the MTCon-
707 nectStreams document for a *Data Entity* defined to represent a *Data Set* **MUST** in-
708 clude the full set of *key-value pairs* that are valid for that *Data Entity*. If the *Current*
709 *Request* includes an at query parameter, the Agent **MUST** provide the set of *key-value*
710 *pairs* that are valid at the specified *sequence number*.

711 When an Agent responds to a *Sample Request*, the information returned in the MTCon-
712 nectStreams document for a *Data Entity* defined to represent a *Data Set* **MUST** in-
713 clude only those *key-value pairs* that are valid for the *Data Entity* at each *sequence number*.

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

718 Note: For some Sample elements, the *Valid Data Value* **MAY** be restricted to spe-
719 cific formats. See Section 6.1 of this document for a description of any restric-
720 tions of the acceptable format for *Valid Data Value*.

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

725 The *Valid Data Value* reported as CDATA for a `Sample` element **MUST** be formatted as part of the content between the element tags in the XML element representing that *Data Entity*. As an example, a `Position` is formatted as shown in *Example 5*.

Example 5: Example showing CDATA of a `DataItem` Element

```
728 1 <Position sequence="112" name="Xabs"
729 2     timestamp="2016-07-28T02:06:01.364428Z"
730 3     dataItemId="10">123.3333</Position>
```

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

734 5.3.4 Unavailability of Valid Data Values for Sample

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

737 *Example 6* demonstrates how an *Agent* reports the value for a `Sample` in the CDATA when it is unable to determine a *Valid Data Value*:

Example 6: Example of CDATA when Data Entity is UNAVAILABLE

```
739 1 <Samples>
740 2   <PathPosition dataItemId="p2"
741 3     timestamp="2009-03-04T19:45:50.458305"
742 4     subType="ACTUAL" name="Zact"
743 5     sequence="15065113">UNAVAILABLE</PathPosition>
744 6   <Temperature dataItemId="t6"
745 7     timestamp="2009-03-04T19:45:50.458305" name="temp"
746 8     sequence="150651134">UNAVAILABLE</Temperature>
747 9 </Samples>
```

748 5.4 Events Container

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

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

Table 11: MTConnect Event Element

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

755 5.5 Event Data Entities

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

759 `Event` is an abstract type XML element and will never appear directly in the `MTCon-`
 760 `nectStreams` XML document. As an abstract type XML element, `Event` will be
 761 replaced in the XML document by a specific type of `Event` specified by the *Element*
 762 *Name* for that *Data Entity*. The different types of `Event` elements are defined in *Sec-*
 763 *tion 6.2 - Event Element Names*. Examples of XML elements representing `Event` include
 764 `Block` and `Execution`.

765 `Event` is similar to `Sample`, but its value can change with unpredictable frequency.
 766 `Events` do not report intermediate values. As an example, when `Availability` tran-
 767 sitions from `UNAVAILABLE` to `AVAILABLE`, there is no intermediate state that can be
 768 inferred.

769 `Event` elements **MAY** report data values defined by a controlled vocabulary as speci-

770 fied in *Section 6.2 - Event Element Names*, by numeric values, or by a character string
 771 representing text or a message provided by the piece of equipment.

Table 12: MTConnect Event Element

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

772 **5.5.1 XML Schema Structure for Event**

773 The XML schema in *Figure 7* represents the structure of an Event XML element show-
 774 ing the attributes defined for Event elements.

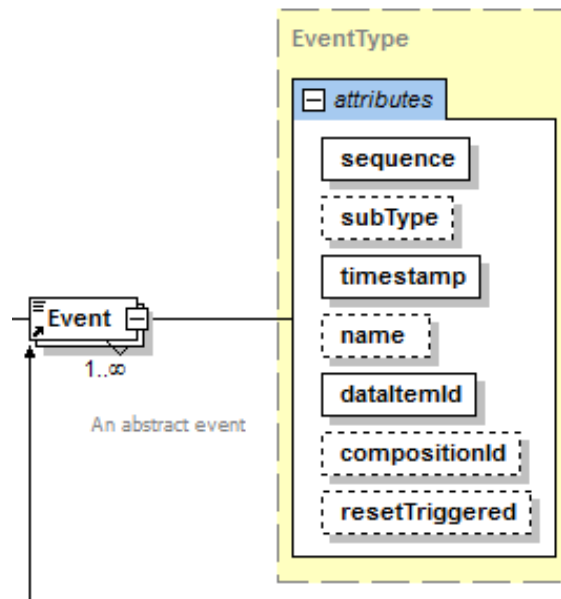


Figure 7: Event Schema Diagram

775 5.5.2 Attributes for Event

776 *Table 13* defines the attributes that **MAY** be used to provide additional information for an
 777 Event XML element.

Table 13: Attributes for Event

Attribute	Description	Occurrence
sequence	<p>A number representing the sequential position of an occurrence of the <code>Event</code> in the data buffer of an <i>Agent</i>.</p> <p>sequence is a required attribute.</p> <p>sequence MUST have a value represented as an unsigned 64-bit value from 1 to $2^{64} - 1$.</p>	1
subType	<p>The subType of the <i>Data Entity</i>.</p> <p>subType is an optional attribute.</p> <p>subType MUST match the subType attribute of the <code>DataItem</code> element as defined in the <code>MTConnectDevices</code> document that the <code>Event</code> element represents.</p>	0..1
timestamp	<p>The most accurate time available to a piece of equipment that represents the point in time that the data reported for the <code>Event</code> was measured.</p> <p>timestamp is a required attribute.</p>	1
name	<p>The name of the <code>Event</code> element.</p> <p>name is an optional attribute.</p> <p>name MUST match the name attribute of the <code>DataItem</code> element defined in the <code>MTConnectDevices</code> document that the <code>Event</code> element represents.</p> <p>An NMTOKEN XML type.</p>	0..1

Continuation of Table 13		
Attribute	Description	Occurrence
dataItemId	<p>The unique identifier for the <code>Event</code> element.</p> <p><code>dataItemId</code> is a required attribute.</p> <p><code>dataItemId</code> MUST match the <code>id</code> attribute of the <code>DataItem</code> element defined in the <code>MTConnectDevices</code> document that the <code>Event</code> element represents.</p>	1
resetTriggered	<p>For those <code>DataItem</code> elements that report data that may be periodically reset to an initial value, <code>resetTriggered</code> identifies when a reported value has been reset and what has caused that reset to occur.</p> <p><code>resetTriggered</code> is an optional attribute.</p> <p><code>resetTriggered</code> MUST only be provided for the specific occurrence of a <i>Data Entity</i> reported in the <code>MTConnectStreams</code> document when the reset occurred and MUST NOT be provided for any other occurrence of the <i>Data Entity</i> reported in a <code>MTConnectStreams</code> document.</p>	0..1
compositionId	<p>The identifier of the <code>Composition</code> element defined in the <code>MTConnectDevices</code> document associated with the data reported for the <code>Event</code> element.</p> <p><code>compositionId</code> is an optional attribute.</p>	0..1

778 5.5.3 Valid Data Values for Event

779 Event elements reported in an `MTConnectStreams` XML document **MUST** provide
780 a value in the CDATA of the *Data Entity*.

781 The value reported in the CDATA **MUST** be reported as either a *Valid Data Value* rep-
782 resenting the information reported from a piece of equipment or UNAVAILABLE when a
783 *Valid Data Value* cannot be determined.

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

788 The `representation` attribute for an `EVENT` category data item defined in the `MT-`
 789 `ConnectDevices` document specifies how an *Agent* **MUST** record instances of data
 790 associated with that data item and how that data **MUST** be reported as an `Event` element
 791 in the `MTConnectStreams` document.

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

796 If a `representation` attribute is not specified for a data item in an `MTConnectDe-`
 797 `vices` document, the designation for the `representation` attribute **MUST** be inter-
 798 preted as `VALUE`.

799 In the case of an `Event` element associated with a `EVENT` category `DataItem` element
 800 with a `representation` attribute of `DATA_SET`, the data reported for each *key-value*
 801 *pair* **MUST** be provided in the same *Valid Data Values* and units as specified by the `type`
 802 attribute for the `DataItem` element.

803 When an *Agent* responds to a *Current Request*, the information returned in the `MTCon-`
 804 `nectStreams` document for a *Data Entity* defined to represent a *Data Set* **MUST** in-
 805 clude the full set of *key-value pairs* that are valid for that *Data Entity*. If the *Current*
 806 *Request* includes an `at` query parameter, the *Agent* **MUST** provide the set of *key-value*
 807 *pairs* that are valid at the specified *sequence number*.

808 When an *Agent* responds to a *Sample Request*, the information returned in the `MTCon-`
 809 `nectStreams` document for a *Data Entity* defined to represent a *Data Set* **MUST** in-
 810 clude only those *key-value pairs* that are valid for the *Data Entity* at each *sequence number*
 811 The *Valid Data Value* reported as `CDATA` for an `Event` element **MUST** be formatted as
 812 part of the content between the element tags in the XML element representing that *Data*
 813 *Entity*. As an example, `Event` elements are formatted as shown in *Example 7*:

Example 7: Example of Event Element

```
814 1 <PartCount dataItemId="pc4"
815 2     timestamp="2009-02-26T02:02:36.48303"
816 3     name="pcount" sequence="185">238</PartCount>
817 4 <ControllerMode dataItemId="p3"
818 5     timestamp="2009-02-26T02:02:35.716224"
819 6     name="mode" sequence="192">AUTOMATIC</ControllerMode>
820 7 <Block dataItemId="cn2" name="block" sequence="206"
```

821 8 timestamp="2009-02-26T02:02:37.394055">G0Z1</Block>

822 In these examples, 238 is the CDATA for PartCount and is a numeric value; AUTO-
823 MATIC is the CDATA for the ControllerMode and is a descriptive value representing
824 a state for the *Data Entity*; and G0Z1 is a text string representing a message describing the
825 program code associated with the Block *Data Entity*.

826 5.5.4 Unavailability of Valid Data Value for Event

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

829 The example in *Example 8* demonstrates how an *Agent* reports the value for an Event in
830 the CDATA when it is unable to determine a *Valid Data Value*:

Example 8: Example of Event Element when data value is UNAVAILABLE

```
831 1 <Events>
832 2   <ControllerMode dataItemId="p3"
833 3     timestamp="2009-02-26T02:02:35.716224" name="mode"
834 4     sequence="182">UNAVAILABLE</ControllerMode>
835 5 </Events>
```

836 5.6 Representations

837 A representation specifies the format and structure of the information for an *obser-*
838 *vation*. The default representation is VALUE indicating the format as specified in
839 *MTConnect Standard: Part 3.0 - Streams Information Model*.

840 A representation, other than VALUE, will modify the *Element Name* of the *obser-*
841 *vation* by appending the pascal case of the representation as follows:

- 842 • A DataItem with type TEMPERATURE and representation of TIME_
843 SERIES becomes TemperatureTimeSeries
- 844 • **DEPRECATED** A DataItem with type PART_COUNT and representa-
845 tion of DISCRETE (**DEPRECATED** in *Version 1.5*) becomes PartCount-
846 Discrete
- 847 • A DataItem with type VARIABLE and representation of DATA_SET be-
848 comes VariableDataSet

- 849 • A DataItem with type WORK_OFFSET and representation of TABLE be-
850 comes WorkOffsetTable

851 The following constraints apply to each representation:

- 852 • A DataItem with representation TIME_SERIES **MUST** have a cate-
853 gory SAMPLE

- 854 • **DEPRECATED** A DataItem with representation DISCRETE (**DEPRECATED**
855 in *Version 1.5*) **MUST** have a category EVENT

- 856 • A DataItem with representation DATA_SET **MUST** have a category
857 EVENT or SAMPLE

- 858 • A DataItem with representation TABLE **MUST** have a category EVENT
859 or SAMPLE

860 **5.6.1 Observations for DataItem with representation of TIME_SE-** 861 **RIES**

862 A DataItem with TIME_SERIES representation **MUST** have a category of
863 SAMPLE.

864 A *Time Series observation* **MUST** have a sampleCount attribute.

865 *Time Series observation* **MUST** report multiple values at fixed intervals in a single *obser-*
866 *vation*. At minimum, one of DataItem or *observation* **MUST** specify the sampleR-
867 ate in *hertz* (values/second); fractional rates are permitted. When the *observation* and
868 the DataItem specify the sampleRate, the *observation* sampleRate supersedes
869 the DataItem.

870 The *observation* **MUST** set the timestamp to the time the last value was observed. The
871 duration **MAY** indicate the time interval from the first to the last value in the series.

872 In XML, the format of the *Time Series observation* **MUST** be space-separated floating-
873 point numbers.

874 **5.6.1.1 XML Schema for Time Series Observation**

875 *Figure 8* shows the attributes that can be applied to all TIME_SERIES *observations*.

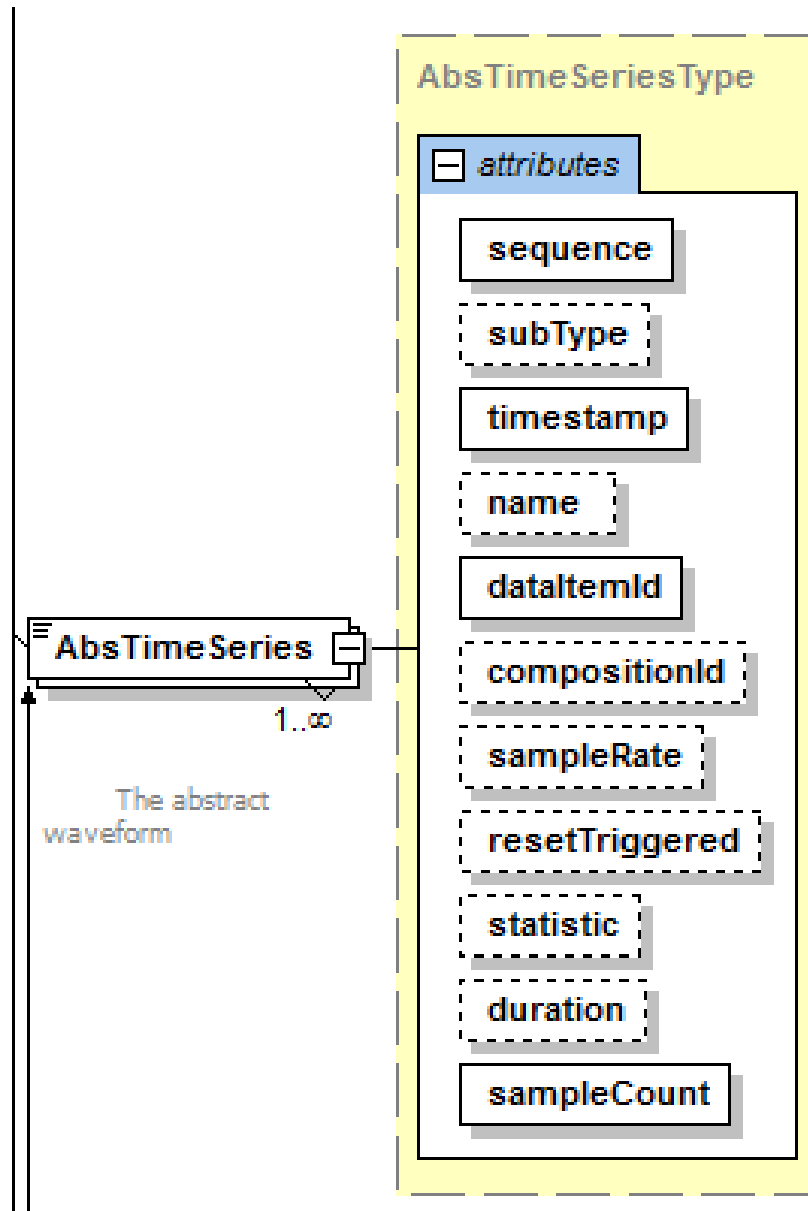


Figure 8: AbsTimeSeries Schema Diagram

876 5.6.1.2 Attributes for Time Series Observation

877 *Table 14* defines the additional attribute provided for a `DataItem` of category SAM-
878 PLE with a representation attribute of `TIME_SERIES`.

Table 14: Attributes for Time Series Observation

Attribute	Description	Occurrence
<code>sampleCount</code>	The number of values given for the <i>observation</i>	1

879 5.6.2 Observations for DataItem with representation of DISCRETE 880 (DEPRECATED)

881 *MTConnect* Version 1.5 replaced representation `DISCRETE` (**DEPRECATED** in
882 *Version 1.5*) with a discrete *attribute* for `DataItem`.

883 `DISCRETE` (**DEPRECATED** in *Version 1.5*) **MUST** only be used with a `DataItem`
884 with a category of `EVENT`.

885 Each occurrence of the *observation* **MAY** have the same value as the previous occurrence,
886 and **MUST NOT** suppress duplicates.

887 Examples of `DISCRETE` (**DEPRECATED** in *Version 1.5*) information as follows: A
888 `PartCount` reporting the completion of each part using a 1 to indicate completion of a
889 single part, a `Message` that occurs each time a door opens.

890 5.6.3 Observations for DataItem with representation of DATA_SET

891 A `DataItem` with `DATA_SET` representation **MUST** have a category of SAM-
892 PLE or `EVENT`.

893 A *Data Set observation* **MUST** have a `count` attribute.

894 *Data Set observation* reports multiple values as a set of *key-value pairs* where each *key*
895 **MUST** be unique. The representation of the *key-value pair* in XML is an `Entry`. The
896 value of each `Entry` **MUST** have the same constraints and format as the *observation*
897 defined for the `VALUE` representation for the `DataItem` type.

898 The meaning of each `Entry` **MAY** be provided as the `DataItemEntryDefinition`.

899 **5.6.3.1 XML Schema for Data Set Observation**

900 *Figure 9* represents the *XML Schema* of a `DataItem` with a representation at-
 901 tribute of `DATA_SET`.

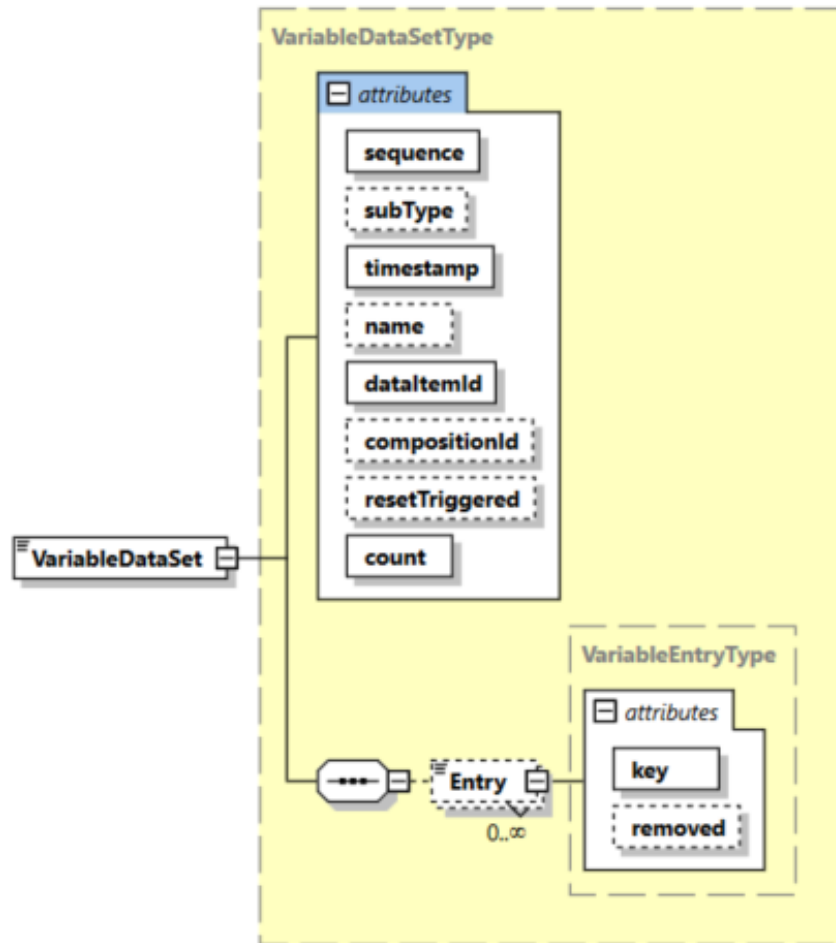


Figure 9: Sample Data Set Schema Diagram

902 *Table 15* defines the additional attribute provided for a `DataItem` with a represen-
 903 tation attribute of `DATA_SET`.

Table 15: Attributes for Data Set Observation

Attribute	Description	Occurrence
count	The number of <code>Entry</code> elements for the <i>observation</i> .	1

904 *Table 16* defines the elements provided for a `DataItem` with a `representation` at-
 905 tribute of `DATA_SET`.

Table 16: Elements for Data Set Observation

Element	Description	Occurrence
Entry	A <i>key-value pair</i> published as part of a <i>Data Set observation</i> .	0..*

906 **5.6.3.2 Entry Element for Data Set Observation**

907 *Figure 10* represents the *XML Schema* structure for a `Entry` XML element that represents
 908 the information published for a *key-value pair*. Any number of `Entry` elements **MAY** be
 909 provided for a *Data Entity* defined with a `representation` attribute of `DATA_SET`.

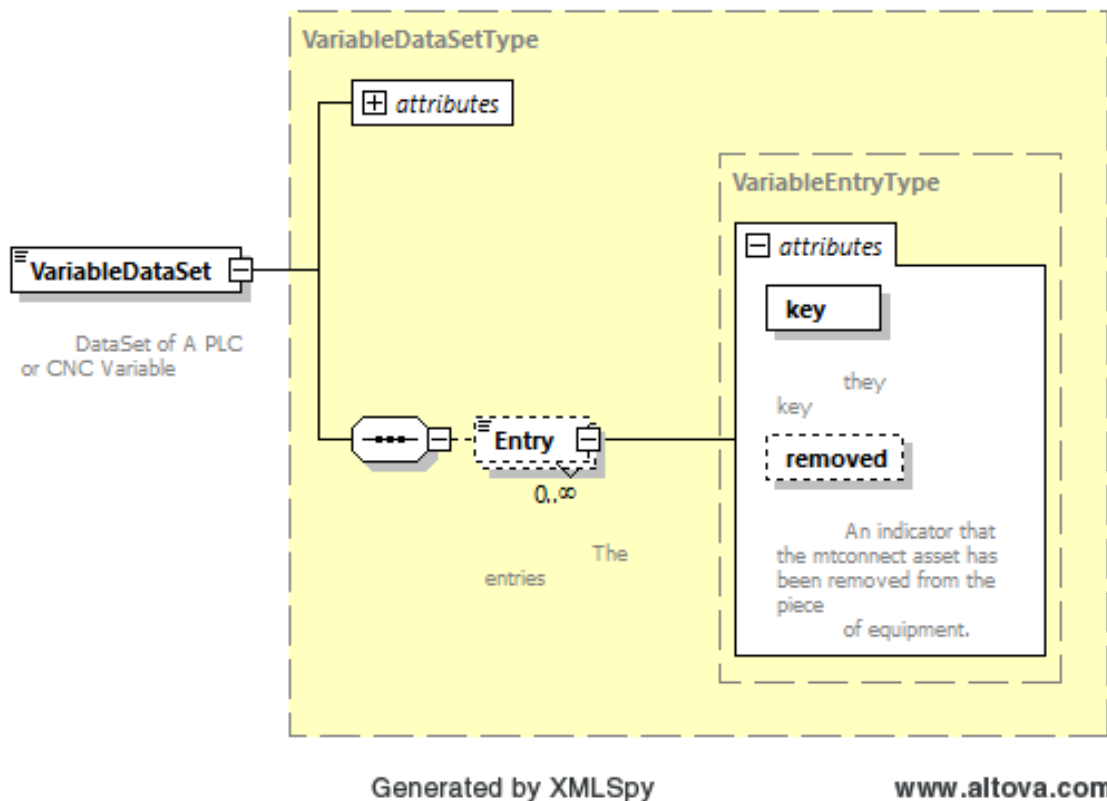


Figure 10: Entry Element Schema Diagram

910 Notes: The `VariableDataSet` is an example of a `DataItem` with type `VARI-`
 911 `ABLE` and `representation DATA_SET`.

912 The following is an example in XML of Entry elements for a DataItem with type
913 VARIABLE:

Example 9: Example of multiple key-value pairs Reported for a Data Entity

```
914 1 <VariableDataSet timestamp="..." sequence="..." count="2">
915 2   <Entry key="a101">100.21</Entry>
916 3   <Entry key="a102">609</Entry>
917 4   <Entry key="a103" removed="true" />
918 5 </VariableDataSet>
```

919 **5.6.3.3 Attributes for Entry Element for Data Set Observation**

920 *Table 17* defines the attributes provided for a Entry XML element.

Table 17: Attributes for Entry

Attribute	Description	Occurrence
key	A unique identifier for each <i>key-value pair</i> . The value provided for key MUST be unique in a set of Entry elements. The value provided for key MUST be an XML NMTOKEN type.	1
removed	Boolean removal indicator of a <i>key-value pair</i> that MUST be true or false. true indicates the Entry is removed. false (default) indicates the Entry is present.	0..1

921 **5.6.3.4 Constraints for Entry Values**

922 The value of each Entry **MUST** have the same restrictions as the value of an *observation*
923 with representaton of VALUE.

924 An Entry **MAY** be further constrained by the DataItem definition (see *MTConnect*
925 *Standard: Part 2.0 - Devices Information Model*), for example a VariableDataSet
926 having a string value **MAY** have a floating-point Temperature value. A restriction
927 **MUST NOT** be broadened or removed, for example, the value "READY" **MUST NOT**
928 occur with a TemperatureDataSet constrained to floating-point numbers.

929 The *MTCConnect Standard: Part 2.0 - Devices Information Model* `DataItem` Defini-
 930 tion **MAY** provide the type and units of an `Entry` for a key.

931 **5.6.4 Management of Data Set Observations**

932 An *Agent* **MUST** maintain the current state of the *Data Set* as described in *MTCConnect*
 933 *Standard Part 1.0 - Overview and Fundamentals Section Part 1: Management of Stream-*
 934 *ing Data Storage*.

935 One or more *key-value pairs* **MAY** be added, removed, or changed in an *observation*. An
 936 *Agent* **MUST** publish the changes to one or more *key-value pairs* as a single *observation*.
 937 An *Agent* **MUST** indicate the removal of a *key-value pair* from a *Data Set* using the
 938 removed attribute equal `true`.

939 When the `DataItem` `discrete` attribute is `false` or is not present, an *Agent* in re-
 940 sponse to a *sample request* **MUST** only publish the changed *key-value pair* since the pre-
 941 vious state of the *Data Set*.

942 When the `DataItem` `discrete` attribute is `true`, an *Agent*, in response to a *sample*
 943 *request*, **MUST** report all *key-value pairs* ignoring the state of the *Data Set*.

944 When an *Agent* responds to a *Current Request*, the *response document* **MUST** include the
 945 full set of *key-value pairs*. If the *Current Request* includes an `at query parameter`, the
 946 *Agent* **MUST** provide the set of *key-value pairs* at the *sequence number*.

947 When an *observation reset* occurs, the *Data Set* **MUST** remove all *key-value pairs* making
 948 the set empty. The *observation* **MAY** simultaneously populate the *Data Set* with new
 949 *key-value pairs*. The previous entries **MUST NOT** be included and **MUST NOT** have
 950 removed attribute equal `true`.

951 When the *observation* is `UNAVAILABLE` the *Data Set* **MUST** remove all *key-value pairs*
 952 making the set empty.

953 **5.6.5 Observations for DataItem with representation of TABLE**

954 A *Table* represents two-dimensional sets of *key-value pairs* where the `Entry` represents
 955 rows containing sets of *key-value pairs* given by `Cell` elements. The *Table* has the same
 956 behavior as the *Data Set* for change tracking, clearing, and history. When an `Entry`
 957 changes. All `Cell` elements update at the same time; they are not tracked separately like
 958 `Entry`.

959 The meaning of each Entry and Cell **MAY** be provided as the DataItem Entry-
960 Definition and CellDefinition.

961 The Entry key attribute **MUST** be the unique identity of the Entry within an *obser-*
962 *vation*. The Cell key attribute **MUST** be the unique identity of the Cell within an
963 Entry.

964 **5.6.5.1 Structure of Table Observations**

965 *Figure 11* represents the XML schema representing DataItem defined in the *MTCConnect*
966 *Standard: Part 2.0 - Devices Information Model* with a representation attribute of
967 TABLE.

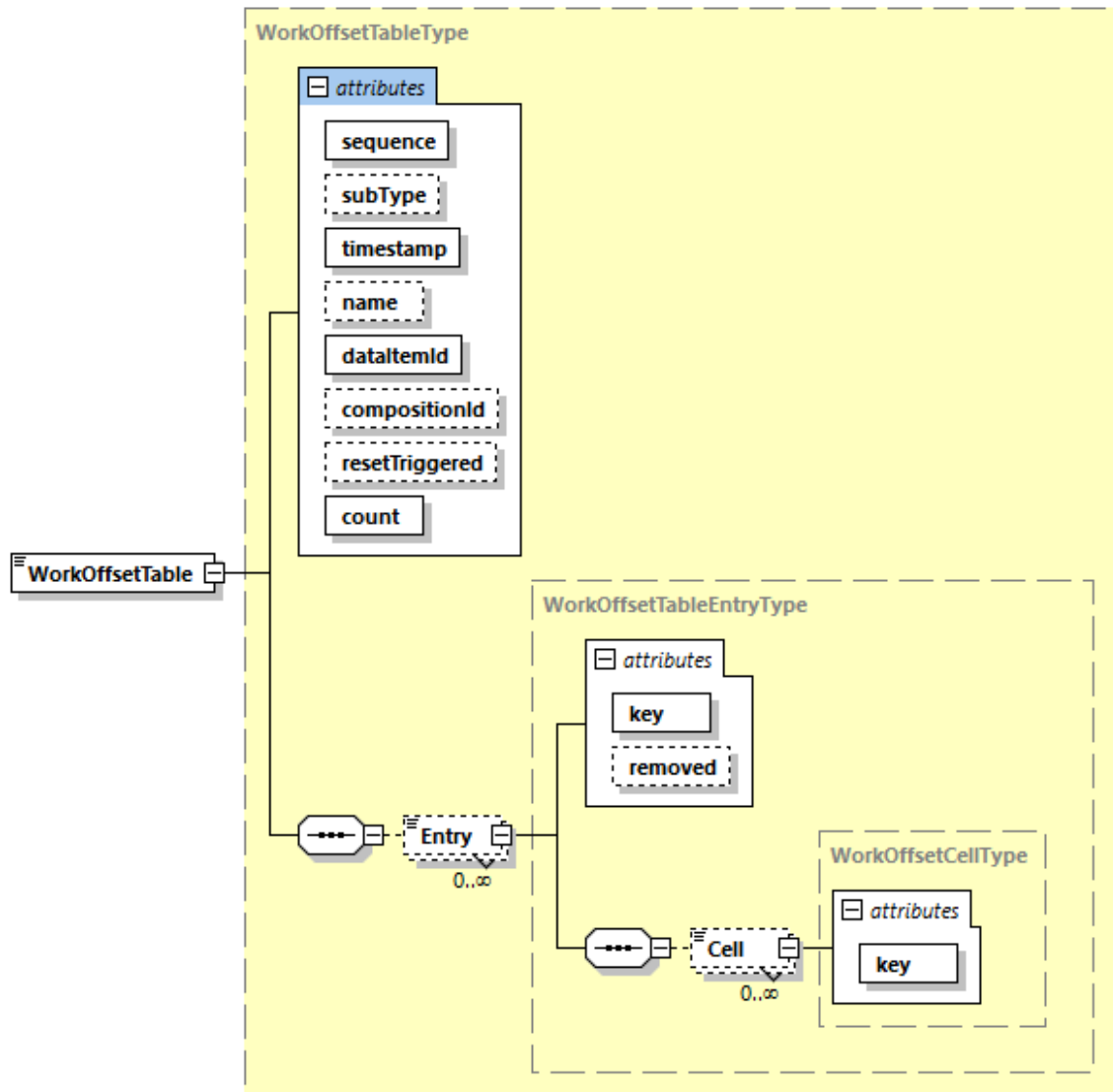


Figure 11: Table Schema Diagram

968 **5.6.5.2 Attributes of Table Observations****Table 18:** Attributes for Table

Attribute	Description	Occurrence
count	Represents the number of <i>key-value pairs</i> represented as Entry elements. count MUST be provided when the DataItem representation is TABLE.	1

969 **5.6.5.3 Elements of Table Observations**

970 *Table 19* An Entry is the only child element that **MAY** be associated with a *Table obser-*
971 *vation*.

Table 19: Elements for Table

Element	Description	Occurrence
Entry	A <i>key-value-pair</i> containing a set of <i>key-value pairs</i> .	0..*

972 **5.6.5.3.1 Structure for Table Entry for an Observation**

973 An Entry represents a *Row* subdivided into Cell elements when representing tabular
974 data. The meaning of an Entry **MAY** be given in the DataItem EntryDefinition
975 associated with its unique key.

976 **5.6.5.3.2 Attributes for Table Entry for an Observation**

977 See *Section 5.6.3.3 - Attributes for Entry Element for Data Set Observation*.

978 **5.6.5.3.3 Elements for Table Cell for an Observation****Table 20:** Elements for Table Cell

Element	Description	Occurrence
Cell	An element representing a <i>key-value pair</i> published as part of an Entry.	0..*

979 **5.6.5.3.4 Structure for Table Cell for an Entry**

980 A Cell represents a *Column* within a *Row* of a tabular data. The `DataItem CellDef-`
 981 `inition` **MAY** give the meaning of the Cell associated with its unique key.

982 Any number of Cell elements **MAY** be provided for an Entry for a *Table observation*.

983 The type of the `DataItem` constrains the *CDATA* of the Cell as specified in *MTConnect Standard: Part 2.0 - Devices Information Model*.

985 **5.6.5.3.5 Attributes for Table Cell for an Observation**

986 *Table 21* defines the attributes provided for a Cell XML element for an Entry.

Table 21: Attributes for Table Cell

Attribute	Description	Occurrence
key	A unique identifier for each <i>key-value pair</i> . The value provided for key MUST be unique in a set of Cell elements. The value provided for key MUST be an XML NMTOKEN type.	1

987 **5.6.5.3.6 Constraints for Cell Values**

988 The value of each Cell **MUST** have the same restrictions as the value of an *observation*
 989 with representaton of VALUE.

990 An Cell **MAY** be further constrained by the `DataItem` definition (see *MTConnect Stan-*

991 *ard: Part 2.0 - Devices Information Model*), for example a `VariableDataSet` having
 992 a string value **MAY** have a floating-point `Temperature` value. A restriction **MUST**
 993 **NOT** be broadened or removed, for example, the value "READY" **MUST NOT** occur
 994 with a `TemperatureDataSet` constrained limited to floating-point numbers.

995 The *MTConnect Standard: Part 2.0 - Devices Information Model* `DataItem` Defini-
 996 tion **MAY** provide the type and units of a `Cell` for a key.

997 5.6.5.3.7 Example Table Observation

Example 10: Example of `WorkpieceOffset` observation for a `TABLE` representation

```

998 1 <WorkpieceOffsetTable dataItemId="wp1" timestamp="TIME" name="wpo"
999 2   sequence="15" count="3">
1000 3   <Entry key="G53.1"><Cell key="X">1</Cell><Cell key="Y">2</Cell>
1001 4     <Cell key="Z">3</Cell></Entry>
1002 5   <Entry key="G53.2"><Cell key="X">4</Cell><Cell key="Y">5</Cell>
1003 6     <Cell key="Z">6</Cell></Entry>
1004 7   <Entry key="G53.3"><Cell key="U">10</Cell><Cell key="X">7</Cell>
1005 8     <Cell key="Y">8</Cell><Cell key="Z">9</Cell></Entry>
1006 9 </WorkpieceOffsetTable>

```

1007 5.7 Condition Container

1008 `Condition` is a `XML` container type element. `Condition` organizes the *Data Entities*
 1009 returned in the `MTConnectStreams` `XML` document for those `DataItem` elements
 1010 defined with a `category` attribute of `CONDITION` in the `MTConnectDevices` docu-
 1011 ment.

1012 A separate `Condition` container will be provided for the data returned for the `DataItem`
 1013 elements associated with each *Structural Element* of a piece of equipment defined in the
 1014 `MTConnectDevices` document.

Table 22: MTConnect Condition Element Container

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

1015 5.8 Condition Data Entity

1016 A Condition XML element provides the information and data provided from a piece of
 1017 equipment for those DataItem elements defined with a category attribute of CON-
 1018 DITION in the MTConnectDevices document.

1019 Condition provides information reported by a piece of equipment describing its health
 1020 and ability to function.

1021 Condition is an abstract type XML element and will never appear directly in the MT-
 1022 ConnectStreams XML document. As an abstract type XML element, Condition
 1023 will be replaced in the XML document by a *Data Entity* representing the CONDITION
 1024 category DataItem element defined in the MTConnectDevices document that this
 1025 Condition element represents.

1026 The *Data Entities* represented by Condition are structured differently than the *Data*
 1027 *Entities* representing Sample and Event. The *Element Name* for each Condition
 1028 element reported in the MTConnectStreams document defines the *Fault State* of the
 1029 *Data Entity*. A Condition element is identified by the *Structural Element* to which it is
 1030 associated, along with the type and dataItemId defined for the element. *Section 6.3*
 1031 *- Types of Condition Elements* provides details on the different types of Condition
 1032 elements.

Table 23: MTConnect Condition Element

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

1033 `CONDITION` type `DataItem` elements defined in the `MTConnectDevices` document
1034 **MAY** report multiple simultaneous *Fault States* in the `MTConnectStreams` document.
1035 This is unlike a `SAMPLE` or `EVENT` `DataItem` element that can only report a single
1036 occurrence of a `Sample` or `Event` element in the `MTConnectStreams` document at
1037 any one point in time.

1038 For example, a controller on a piece of equipment may detect and report multiple for-
1039 mat errors in a motion program. Each error represents a separate *Fault State* from the
1040 controller. Each *Fault State* is represented as a separate `Condition` element in the `MT-`
1041 `ConnectStreams` document since each *Fault State* **MUST** be identified and tracked
1042 individually in the document.

1043 5.8.1 Element Names for Condition

1044 `Condition` elements are reported differently from other *Data Entity* types. The *El-*
1045 *ement Name* reported for a `Condition` element represents the *Fault State* (Normal,
1046 Warning, or Fault) associated with each `Condition`.

1047 Examples of XML elements representing `Condition` elements for each of the possible
1048 *Fault States* are shown in *Example 11*:

Example 11: Example of Condition Element Fault States

```

1049 1 <Normal type="MOTION_PROGRAM" dataItemId="cc2" sequence="25"
1050 2     timestamp="2010-04-06T06:19:35.153141"></Normal>
1051 3 <Fault type="COMMUNICATIONS" dataItemId="cc1" sequence="26"

```

```

1052 4     nativeCode="IO1231" timestamp="2010-04-
1053 5     06T06:19:35.153141">Communications error</Fault>
1054 6 <Warning type="LOGIC_PROGRAM" dataItemId="pm6" sequence="32"
1055 7     timestamp="2010-04-06T06:19:35.153141">Warning/>

```

1056 5.8.2 XML Schema Structure for Condition

1057 The XML schema in *Figure 12* represents the structure of a Condition XML element
 1058 showing the attributes defined for Condition elements.

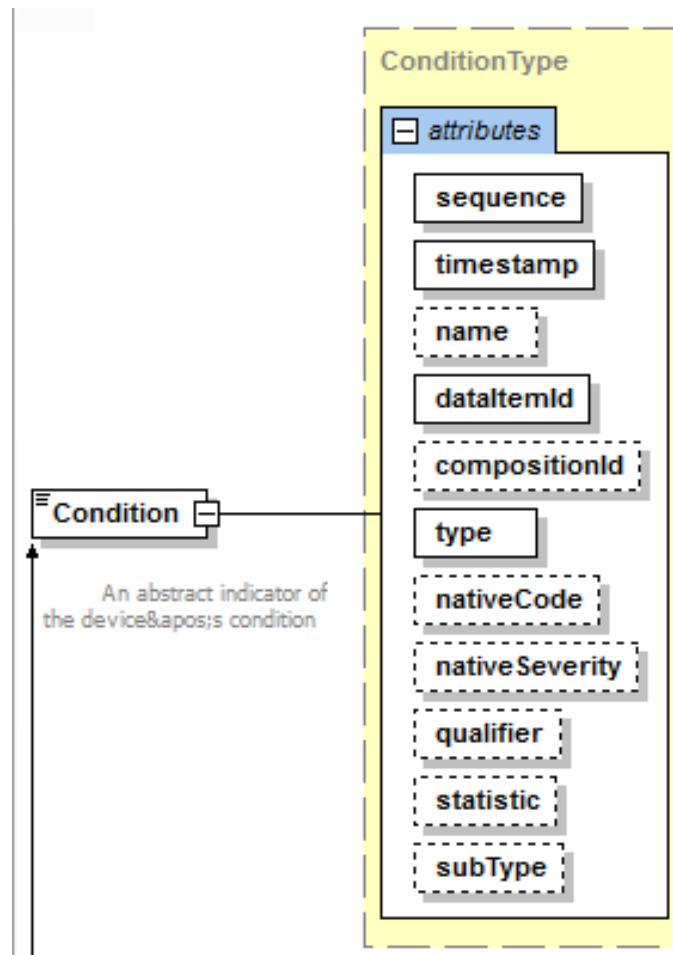


Figure 12: Condition Schema Diagram

1059 5.8.3 Attributes for Condition

1060 *Table 24* defines the attributes used to provide additional information for a `Condition`
 1061 XML element.

Table 24: Attributes for Condition

Attribute	Description	Occurrence
sequence	<p>A number representing the sequential position of an occurrence of the <code>Condition</code> in the data buffer of an <code>MTConnect Agent</code>.</p> <p>sequence is a required attribute.</p> <p>sequence MUST have a value represented as an unsigned 64-bit value from 1 to $2^{64} - 1$.</p>	1
timestamp	<p>The most accurate time available to a piece of equipment that represents the point in time that the data reported for the <code>Condition</code> was measured.</p> <p>timestamp is a required attribute.</p>	1
name	<p>The name of the <code>Condition</code> element.</p> <p>name is an optional attribute.</p> <p>name MUST match the name attribute of the <code>DataItem</code> element defined in the <code>MTConnectDevices</code> document that the <code>Condition</code> element represents.</p> <p>An <code>NMTOKEN</code> XML type.</p>	0..1
dataItemId	<p>The unique identifier for the <code>Condition</code> element.</p> <p>dataItemId is a required attribute.</p> <p>dataItemId MUST match the id attribute of the <code>DataItem</code> element defined in the <code>MTConnectDevices</code> document that the <code>Condition</code> element represents.</p>	1

Continuation of Table 24		
Attribute	Description	Occurrence
type	<p>An identifier of the <code>type</code> of fault represented by the <code>Condition</code> element.</p> <p><code>type</code> is a required attribute.</p> <p><code>type</code> MUST match the <code>type</code> attribute of the <code>DataItem</code> element defined in the <code>MTConnectDevices</code> document that this <code>Condition</code> element represents.</p>	1
nativeCode	<p>The native code (usually an alpha-numeric value) generated by the controller of a piece of equipment providing a reference identifier for a <code>Condition</code>.</p> <p><code>nativeCode</code> is an optional attribute.</p> <p>This is the same information an operator or maintenance personnel may see as a reference code designating a specific fault code provided by the piece of equipment.</p>	0..1
nativeSeverity	<p>If the piece of equipment designates a severity level to a fault, <code>nativeSeverity</code> reports that severity information to a client software application.</p> <p><code>nativeSeverity</code> is an optional attribute.</p>	0..1

Continuation of Table 24		
Attribute	Description	Occurrence
qualifier	<p>qualifier provides additional information regarding a <i>Fault State</i> associated with the measured value of a process variable.</p> <p>qualifier is an optional attribute.</p> <p>qualifier defines whether the <i>Fault State</i> represented by the Condition indicates a measured value that is above or below an expected value of a process variable.</p> <p>If the <i>Fault State</i> represents a measured value that is greater than the expected value for the process variable, qualifier MUST report a value of HIGH.</p> <p>If the <i>Fault State</i> represents a measured value that is less than the expected value for the process variable, qualifier MUST report a value of LOW.</p>	0..1
statistic	<p>statistic provides additional information describing the meaning of the Condition element.</p> <p>statistic is an optional attribute.</p> <p>statistic MUST match the statistic attribute of the DataItem element defined in the MTConnectDevices document that this Condition element represents.</p>	0..1
subType	<p>subType provides additional information describing the meaning of the Condition element.</p> <p>subType is an optional attribute.</p> <p>subType MUST match the subType attribute of the DataItem element defined in the MTConnectDevices document that this Condition element represents.</p>	0..1

Continuation of Table 24		
Attribute	Description	Occurrence
compositionId	The identifier of the <code>Composition</code> element defined in the <code>MTConnectDevices</code> document associated with the data reported for the <code>Condition</code> element. compositionId is an optional attribute.	0..1
xs:lang	An optional attribute that specifies the language of the CDATA returned for the <code>Condition</code> . Refer to IETF RFC 4646 (http://www.ietf.org/rfc/rfc4646.txt) or successor for a full definition of the values for this attribute. xs:lang does not appear in the schema diagram.	0..1

1062 **5.8.3.1 qualifier Attribute for Condition**

1063 Many `Condition` elements report the *Fault State* associated with the measured value of
1064 a process variable.

1065 `qualifier` provides an indication whether the measured value is above or below an
1066 expected value of a process variable.

1067 As an example, a `Condition` element with a `type` attribute of `AMPERAGE` may differ-
1068 entiate between a higher than expected amperage and a lower than expected amperage by
1069 using the `qualifier` attribute.

1070 When a `qualifier` of either `HIGH` or `LOW` is used with `Fault` and `Warning`, the
1071 *Fault States* can be differentiated as follows:

1072 `Fault,LOW`

1073 `Warning,LOW`

1074 `Normal`

1075 `Warning,HIGH`

1076 Fault,HIGH

1077 *Example 12* is an example of an XML element representing Condition using quali-
1078 fier:

Example 12: Example of a Condition Element using qualifier

```
1079 1 <Warning type="FILL_LEVEL" dataItemId="pm6"
1080 2     qualifier="HIGH" sequence="32"
1081 3     timestamp="2009-11-13T08:32:18">...</Warning>
```

1082 5.8.4 Valid Data Value for Condition

1083 Condition elements reported in an MTConnectStreams XML document **MAY** pro-
1084 vide a value in the CDATA of the *Data Entity* when additional information regarding the
1085 *Fault State* is available.

1086 A *Valid Data Value* for the CDATA included in a Condition element **MAY** be any text
1087 string. A *Valid Data Value* is not required to be reported for a Condition category *Data*
1088 *Entity*. The *Fault State* and the attributes provided in a Condition element **MAY** be
1089 sufficient to fully describe the *Data Entity*.

1090 The *Valid Data Value* reported as CDATA for a Condition element **MUST** be formatted
1091 as part of the content between the element tags in the XML element representing that *Data*
1092 *Entity*. As an example, Condition elements are formatted as shown in *Example 13*:

Example 13: Example of CDATA for Condition

```
1093 1 <Warning type="FILL_LEVEL" dataItemId="pm6"
1094 2     qualifier="HIGH" sequence="32" timestamp=
1095 3     "2009-11-13T08:32:18">Fill Level on Tank
1096 4     #12 is reaching a high level</Warning>
```

1097 In this example, the “Fill Level on Tank #12 is reaching a high level” is the CDATA for
1098 the *Data Entity*.

1099 5.9 Unavailability of Fault State for Condition

1100 When an *Agent* cannot determine a valid *Fault State* for a Condition element, it **MUST**
1101 report the *Element Name* for the *Data Entity* as Unavailable.

1102 *Example 14* demonstrates how an *Agent* reports a Condition category *Data Entity* when
1103 it is unable to determine a valid *Fault State*:

Example 14: Example of Condition when Fault State is UNAVAILABLE

```
1104 1 <Unavailable type="MOTION_PROGRAM" dataItemId="cc2"  
1105 2     sequence="25" timestamp=  
1106 3     "2009-11-13T08:32:18">...</Unavailable>  
1107 4 <Unavailable type="COMMUNICATIONS" dataItemId="cc1"  
1108 5     sequence="26" timestamp=  
1109 6     "2009-11-13T08:32:18">...</Unavailable>  
1110 7 <Unavailable type="LOGIC_PROGRAM" dataItemId="cc3"  
1111 8     sequence="28" timestamp=  
1112 9     "2009-11-13T08:32:18">...</Unavailable>  
1113 10 <Unavailable type="LOGIC_PROGRAM" dataItemId="pm6"  
1114 11     sequence="32" timestamp=  
1115 12     "2009-11-13T08:32:18">...</Unavailable>
```

1116 6 Listing of Data Entities

1117 *Data Entities* that report data in MTConnectStreams documents are represented by
 1118 Sample, Event, or Condition elements based upon the category and type at-
 1119 tributes defined for the corresponding DataItem XML element in the MTConnectDe-
 1120 vices document.

1121 Each *Data Entity* in the MTConnectStreams document has an *Element Name*, as de-
 1122 fined in the following sections, based upon the corresponding category attribute defined
 1123 for that DataItem element in the MTConnectDevices document.

1124 6.1 Sample Element Names

1125 *Table 25* lists the XML elements that can be placed in the Samples container of the
 1126 ComponentStream element.

1127 The *Table 25* shows both the type attribute for each SAMPLE category DataItem ele-
 1128 ment as defined in the MTConnectDevices document and the corresponding *Element*
 1129 *Name* for the *Data Entity* that **MUST** be reported as a Sample element in the MTCon-
 1130 nectStreams document.

Table 25: Element Names for Sample

DataItem Type	Element Name	Description
ACCELERATION	Acceleration	The measurement of the rate of change of velocity. Acceleration MUST be reported in units of MILLIMETER/SECOND ² .

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
ACCUMULATED_TIME	AccumulatedTime	<p>The measurement of accumulated time for an activity or event.</p> <p>AccumulatedTime MUST be reported in units of MILLIMETER/SECOND².</p> <p>DEPRECATION WARNING : May be deprecated in the future. Recommend using ProcessTimer and EquipmentTimer.</p>
AMPERAGE	Amperage	<p>DEPRECATED in <i>Version 1.6</i>. Replaced by AMPERAGE_AC and AMPERAGE_DC.</p>
AMPERAGE_AC	AmperageAC	<p>The measurement of an electrical current that reverses direction at regular short intervals.</p> <p>Subtypes of AMPERAGE_AC are ACTUAL, COMMANDED and PROGRAMMED.</p> <p>AmperageAC is reported in units of AMPERE.</p>

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
AMPERAGE_DC	AmperageDC	<p>The measurement of an electric current flowing in one direction only.</p> <p>Subtypes of AMPERAGE_DC are ACTUAL, COMMANDED and PROGRAMMED.</p> <p>AmperageDC is reported in units of AMPERE.</p>
ANGLE	Angle	<p>The measurement of angular position.</p> <p>Subtypes of Angle are ACTUAL and COMMANDED.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subType of ACTUAL.</p> <p>Angle MUST be reported in units of DEGREE.</p>
ANGULAR_- ACCELERATION	AngularAcceleration	<p>The measurement rate of change of angular velocity.</p> <p>AngularAcceleration MUST be reported in units of DEGREE/SECOND².</p>
ANGULAR_VELOCITY	AngularVelocity	<p>The measurement of the rate of change of angular position.</p> <p>AngularVelocity MUST be reported in units of DEGREE/SECOND.</p>

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
AXIS_FEEDRATE	AxisFeedrate	<p>The measurement of the feedrate of a linear axis.</p> <p>Subtypes of AxisFeedrate are ACTUAL, COMMANDED, JOG, PROGRAMMED, and RAPID.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subType of PROGRAMMED.</p> <p>AxisFeedrate MUST be reported in units of MILLIMETER/SECOND.</p>
CAPACITY_FLUID	CapacityFluid	<p>The fluid capacity of an object or container.</p> <p>CapacityFluid MUST be reported in units of MILLILITER.</p>
CAPACITY_SPATIAL	CapacitySpatial	<p>The geometric capacity of an object or container.</p> <p>CapacitySpatial MUST be reported in units of CUBIC_MILLIMETER.</p>
CLOCK_TIME	ClockTime	<p>The value provided by a timing device at a specific point in time.</p> <p>ClockTime MUST be reported in W3C ISO 8601 format of yyyy-mm-ddthh:mm:ss.ffff.</p>

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
CONCENTRATION	Concentration	<p>The measurement of the percentage of one component within a mixture of components</p> <p>Concentration MUST be reported in units of PERCENT.</p>
CONDUCTIVITY	Conductivity	<p>The measurement of the ability of a material to conduct electricity.</p> <p>Conductivity MUST be reported in units of SIEMENS/METER.</p>
CUTTING_SPEED	CuttingSpeed	<p>The speed difference (relative velocity) between the cutting mechanism and the surface of the workpiece it is operating on.</p> <p>Subtypes of CUTTING_SPEED are ACTUAL, COMMANDED, and PROGRAMMED.</p> <p>If no subType is specified, the reported value must default to PROGRAMMED.</p> <p>CuttingSpeed is reported in units of MILLIMETER/SECOND.</p>
DENSITY	Density	<p>The volumetric mass of a material per unit volume of that material.</p> <p>Density MUST be reported in units of MILLIGRAM/CUBIC_MILLIMETER.</p>

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
DEPOSITION_- ACCELERATION_- VOLUMETRIC	DepositionAccelerationVolumetric	<p>The rate of change in spatial volume of material deposited in an additive manufacturing process.</p> <p>Subtypes of DepositionAccelerationVolumetric are ACTUAL and COMMANDED.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subtype of ACTUAL.</p> <p>DepositionAccelerationVolumetric MUST be reported in units of CUBIC_-MILLIMETER/SECOND².</p>
DEPOSITION_- DENSITY	DepositionDensity	<p>The density of the material deposited in an additive manufacturing process per unit of volume.</p> <p>Subtypes of DepositionDensity are ACTUAL and COMMANDED.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subtype of ACTUAL.</p> <p>DepositionDensity MUST be reported in units of MILLIGRAM/CUBIC_-MILLIMETER.</p>

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
DEPOSITION_MASS	DepositionMass	<p>The mass of the material deposited in an additive manufacturing process.</p> <p>Subtypes of DepositionMass are ACTUAL and COMMANDED.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subtype of ACTUAL.</p> <p>DepositionMass MUST be reported in units of MILLIGRAM.</p>
DEPOSITION_- RATE_VOLUMETRIC	DepositionRateVolumetric	<p>The rate at which a spatial volume of material is deposited in an additive manufacturing process.</p> <p>Subtypes of DepositionRateVolumetric are ACTUAL and COMMANDED.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subtype of ACTUAL.</p> <p>DepositionRateVolumetric MUST be reported in units of CUBIC_MILLIMETER/SECOND.</p>

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
DEPOSITION_ VOLUME	DepositionVolume	<p>The spatial volume of material deposited in an additive manufacturing process.</p> <p>Subtypes of DepositionVolume are ACTUAL and COMMANDED.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subtype of ACTUAL.</p> <p>DepositionVolume MUST be reported in units of CUBIC_MILLIMETER.</p>
DIAMETER	Diameter	<p>The measured dimension of a diameter.</p> <p>Diameter MUST be reported in units of MILLIMETER.</p>
DISPLACEMENT	Displacement	<p>The measurement of the change in position of an object.</p> <p>Displacement MUST be reported in units of MILLIMETER.</p>
ELECTRICAL_ ENERGY	ElectricalEnergy	<p>The measurement of electrical energy consumption by a component.</p> <p>ElectricalEnergy MUST be reported in units of WATT_SECOND.</p>

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
EQUIPMENT_TIMER	EquipmentTimer	<p>The measurement of the amount of time a piece of equipment or a sub-part of a piece of equipment has performed specific activities.</p> <p>Subtypes of EquipmentTimer are LOADED, WORKING, OPERATING, POWERED, and DELAY.</p> <p>A subType MUST always be specified.</p> <p>EquipmentTimer MUST be reported in units of SECOND.</p>
FILL_LEVEL	FillLevel	<p>The measurement of the amount of a substance remaining compared to the planned maximum amount of that substance.</p> <p>FillLevel MUST be reported in units of PERCENT.</p>
FLOW	Flow	<p>The measurement of the rate of flow of a fluid.</p> <p>Flow MUST be reported in units of LITER/SECOND.</p>
FREQUENCY	Frequency	<p>The measurement of the number of occurrences of a repeating event per unit time.</p> <p>Frequency MUST be reported in units of HERTZ.</p>

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
GLOBAL_POSITION	GlobalPosition	DEPRECATED in Version 1.1
HUMIDITY_-ABSOLUTE	HumidityAbsolute	<p>The amount of water vapor expressed in grams per cubic meter.</p> <p>Subtypes of HumidityAbsolute are ACTUAL and COMMANDED.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subtype of ACTUAL.</p> <p>HumidityAbsolute MUST be reported in units of GRAM/CUBIC_METER.</p>
HUMIDITY_-RELATIVE	HumidityRelative	<p>The amount of water vapor present expressed as a percent to reach saturation at the same temperature.</p> <p>Subtypes of HumidityRelative are ACTUAL and COMMANDED.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subtype of ACTUAL.</p> <p>HumidityRelative MUST be reported in units of PERCENT.</p>

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
HUMIDITY_ SPECIFIC	HumiditySpecific	<p>The ratio of the water vapor present over the total weight of the water vapor and air present expressed as a percent.</p> <p>Subtypes of HumiditySpecific are ACTUAL and COMMANDED.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subtype of ACTUAL.</p> <p>HumiditySpecific MUST be reported in units of PERCENT.</p>
LENGTH	Length	<p>The measurement of the length of an object.</p> <p>Subtypes of Length are STANDARD, REMAINING, and USEABLE.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subType of REMAINING.</p> <p>Length MUST be reported in units of MILLIMETER.</p>
LEVEL	Level	DEPRECATED in Version 1.2. See FILL_LEVEL

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
LINEAR_FORCE	LinearForce	<p>The measurement 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 piece of equipment.</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>
ORIENTATION	Orientation	<p>A measured or calculated orientation of a plane or vector relative to a cartesian coordinate system</p> <p>The value of Orientation MUST be three space-delimited floating-point numbers and MUST be in units of DEGREE_3D. The values represent the degrees of rotation around the X, Y, and Z axes respectively as the ordered values A, B, and C.</p> <p>If any of the rotations is not known, it MUST be zero (0).</p>

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
PATH_FEEDRATE	PathFeedrate	<p>The measurement of the feedrate for the axes, or a single axis, associated with a Path component—a vector.</p> <p>Subtypes of PathFeedrate are ACTUAL, COMMANDED, JOG, PROGRAMMED, and RAPID.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subType of PROGRAMMED.</p> <p>PathFeedrate MUST be reported in units of MILLIMETER/SECOND.</p>
PATH_FEEDRATE_PER_REVOLUTION	PathFeedratePerRevolution	<p>The feedrate for the axes, or a single axis.</p> <p>PathFeedratePerRevolution is reported in units of MILLIMETER/REVOLUTION.</p> <p>Subtypes of PathFeedratePerRevolution are ACTUAL, COMMANDED, and PROGRAMMED.</p>

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
PATH_POSITION	PathPosition	<p>A measured or calculated position of a control point reported by a piece of equipment expressed in WORK coordinates. The coordinate system will revert to MACHINE coordinates if WORK coordinates are not available.</p> <p>Subtypes of PathPosition are ACTUAL, PROGRAMMED, COMMANDED, TARGET, and PROBE.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subtype of ACTUAL.</p> <p>PathPosition MUST be reported as a set of space-delimited floating-point numbers representing a point in 3-D space. The position of the control point MUST be reported in units of MILLIMETER and listed in order of X, Y, and Z referenced to the coordinate system of the piece of equipment.</p>

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
PATH_POSITION (Continued)	PathPosition	An example of the value reported for PathPosition would be: <PathPosition ...>10.123 55.232 100.981 </PathPosition> Where X = 10.123, Y = 55.232, and Z=100.981.
PH	PH	A measure of the acidity or alkalinity of a solution. PH MUST be reported in units of PH.

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
POSITION	Position	<p>A measured or calculated position of a Component element as reported by a piece of equipment.</p> <p>Subtypes of Position are ACTUAL, COMMANDED, PROGRAMMED, and TARGET.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subType of ACTUAL.</p> <p>When Position is provided representing a measured value for the physical axes of the piece of equipment, the data MUST be provided in MACHINE coordinates.</p> <p>When Position is provided representing a logical or calculated position, the data MUST be provided in WORK coordinates and is associated with a Path element of the equipment controller.</p> <p>Position MUST be reported in units of MILLIMETER.</p>

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
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 measurement of force per unit area exerted by a gas or liquid. The measurement of force per unit area exerted by a gas or liquid.</p> <p>Pressure MUST be reported in units of PASCAL.</p>
PROCESS_TIMER	ProcessTimer	<p>The measurement of the amount of time a piece of equipment has performed different types of activities associated with the process being performed at that piece of equipment.</p> <p>Subtypes of ProcessTimer are PROCESS, and DELAY.</p> <p>A subType MUST always be specified.</p> <p>ProcessTimer MUST be reported in units of SECOND.</p>

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
RESISTANCE	Resistance	<p>The measurement of the degree to which a substance opposes the passage of an electric current.</p> <p>Resistance MUST be reported in units of OHM.</p>
ROTARY_VELOCITY	RotaryVelocity	<p>The measurement of the rotational speed of a rotary axis.</p> <p>Subtypes of RotaryVelocity are ACTUAL, COMMANDED and PROGRAMMED.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subType of ACTUAL.</p> <p>RotaryVelocity MUST be reported in units of REVOLUTION/MINUTE.</p>

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
SOUND_LEVEL	SoundLevel	<p>The measurement of a sound level or sound pressure level relative to atmospheric pressure.</p> <p>Subtypes of SoundLevel are NO_SCALE, A_SCALE, B_SCALE, C_SCALE and D_SCALE.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subType of NO_SCALE.</p> <p>SoundLevel MUST be reported in units of DECIBEL.</p>
SPINDLE_SPEED	SpindleSpeed	DEPRECATED in Version 1.2. Replaced by ROTARY_VELOCITY
STRAIN	Strain	<p>The measurement of 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>

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
TEMPERATURE	Temperature	<p>The measurement of temperature.</p> <p>Subtypes of Temperature are ACTUAL and COMMANDED.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subtype of ACTUAL.</p> <p>Temperature MUST be reported in units of CELSIUS.</p>
TENSION	Tension	<p>The measurement of a force that stretches or elongates an object.</p> <p>Tension MUST be reported in units of NEWTON.</p>
TILT	Tilt	<p>The measurement of angular displacement.</p> <p>Tilt MUST be reported in units of MICRO_RADIAN.</p>
TORQUE	Torque	<p>The measurement of the turning force exerted on an object or by an object.</p> <p>Torque MUST be reported in units of NEWTON_METER.</p>

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
VELOCITY	Velocity	<p>The measurement of the rate of change of position of a Component.</p> <p>When provided as the Velocity of the Axes Component, it represents the value of the velocity vector for all given axes, similar to PathFeedrate.</p> <p>When provided as the Velocity of an individual Axis Component, it represents the value of the velocity for that specific axis with no influence of the relative velocity of any other axes.</p> <p>Velocity MUST be reported in units of MILLIMETER/SECOND.</p>
VISCOSITY	Viscosity	<p>The measurement of a fluids resistance to flow.</p> <p>Viscosity MUST be reported in units of PASCAL_SECOND.</p>
VOLTAGE	Voltage	<p>DEPRECATED in <i>Version 1.6</i>. Replaced by VOLTAGE_AC and VOLTAGE_DC.</p>

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
VOLTAGE_AC	VoltageAC	<p>The measurement of the electrical potential between two points in an electrical circuit in which the current periodically reverses direction.</p> <p>Subtypes of VOLTAGE_AC are ACTUAL, PROGRAMMED, and COMMANDED.</p> <p>VoltageAC MUST be in units of VOLT.</p>
VOLTAGE_DC	VoltageDC	<p>The measurement of the electrical potential between two points in an electrical circuit in which the current is unidirectional.</p> <p>Subtypes of VOLTAGE_DC are ACTUAL, PROGRAMMED, and COMMANDED.</p> <p>VoltageDC MUST be in units of VOLT.</p>
VOLT_AMPERE	VoltAmpere	<p>The measurement of the apparent power in an electrical circuit, equal to the product of root-mean-square (RMS) voltage and RMS current (commonly referred to as VA).</p> <p>VoltAmpere MUST be reported in units of VOLT_AMPERE.</p>

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
VOLT_AMPERE_- REACTIVE	VoltAmpereReactive	<p>The measurement of reactive power in an AC electrical circuit (commonly referred to as VAR).</p> <p>VoltAmpereReactive MUST be reported in units of VOLT_AMPERE_-REACTIVE.</p>
VOLUME_FLUID	VolumeFluid	<p>The fluid volume of an object or container.</p> <p>Subtypes of VolumeFluid are ACTUAL and CONSUMED.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subtype of ACTUAL.</p> <p>VolumeFluid MUST be reported in units of MILLILITER.</p>
VOLUME_SPATIAL	VolumeSpatial	<p>The geometric volume of an object or container.</p> <p>Subtypes of VolumeSpatial are ACTUAL and CONSUMED.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subtype of ACTUAL.</p> <p>VolumeSpatial MUST be reported in units of CUBIC_MILLIMETER.</p>

Continuation of Table 25: Element Names for Sample		
DataItem Type	Element Name	Description
WATTAGE	Wattage	<p>The measurement of power flowing through or dissipated by an electrical circuit or piece of equipment.</p> <p>Subtypes of Wattage are ACTUAL and TARGET.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subType of ACTUAL.</p> <p>Wattage MUST be reported in units of WATT.</p>
X_DIMENSION	XDimension	<p>Measured dimension of an entity relative to the X direction of the referenced coordinate system.</p> <p>XDimension MUST be reported in units of MILLIMETER.</p>
Y_DIMENSION	YDimension	<p>Measured dimension of an entity relative to the Y direction of the referenced coordinate system.</p> <p>YDimension MUST be reported in units of MILLIMETER.</p>
Z_DIMENSION	ZDimension	<p>Measured dimension of an entity relative to the Z direction of the referenced coordinate system.</p> <p>ZDimension MUST be reported in units of MILLIMETER.</p>

1131 Note: The Sample response format **MUST** be extended when the represen-
1132 tation attribute for the data item is TIME_SERIES. See *Section 5.6.1 -*
1133 *Observations for DataItem with representation of TIME_SERIES* for details on
1134 extending the response format.

1135 6.2 Event Element Names

1136 *Table 26* lists the XML elements that can be placed in the Events container of the Com-
1137 ponentStream element.

1138 The *Table 25* shows both the type for each EVENT category DataItem element defined
1139 in the MTConnectDevices document and the corresponding *Element Name* for the
1140 *Data Entity* that **MUST** be reported as an Event element in the MTConnectStreams
1141 document.

1142 The table also defines the *Valid Data Value* for those Event type data items where the
1143 reported values are restricted to a *Controlled Vocabulary*.

Table 26: Element Names for Event

DataItem Type	Element Name	Description
ACTIVE_AXES	ActiveAxes	<p>The set of axes currently associated with a Path or Controller <i>Structural Element</i>.</p> <p>The <i>Valid Data Value</i> reported SHOULD be a space-delimited set of axes names. The names returned SHOULD match the name attribute of the Linear or Rotary <i>Structural Elements</i> defined in the MTConnectDevices document that this Event element represents. If name is not available, nativeName MUST be returned to identify the Linear or Rotary <i>Structural Elements</i>.</p> <p>For example:</p> <pre data-bbox="964 1209 1349 1276"><ActiveAxes ...>X Y Z W S</ActiveAxes></pre> <p>where X, Y, Z, W, and S are the nativeName attributes of the <i>Structural Elements</i>.</p> <p>If it is not specified elsewhere in the MTConnectDevices document, it MUST be assumed that all of the axes are associated with the Path component.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
ACTUATOR_STATE	ActuatorState	<p>Represents the operational state of an apparatus for moving or controlling a mechanism or system.</p> <p><i>Valid Data Values:</i></p> <p>ACTIVE: The actuator is operating</p> <p>INACTIVE: The actuator is not operating</p>
ALARM	Alarm	DEPRECATED : Replaced with CONDITION category data items in Version 1.1.0.
APPLICATION	Application	<p>The application on a component.</p> <p>Subtypes of APPLICATION are LICENSE, VERSION, RELEASE_DATE, INSTALL_DATE, and MANUFACTURER.</p> <p>The <i>Valid Data Value</i> MUST be a text string.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
AVAILABILITY	Availability	<p>Represents the <i>Agent's</i> ability to communicate with the data source.</p> <p>Availability MUST be provided for each <i>Device Structural Element</i> and MAY be provided for any other <i>Structural Element</i>.</p> <p><i>Valid Data Values:</i></p> <p>AVAILABLE: The <i>Structural Element</i> is active and capable of providing data.</p> <p>AVAILABLE: The <i>Structural Element</i> is either inactive or not capable of providing data.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
AXIS_ COUPLING	AxisCoupling	<p>Describes the way the axes will be associated to each other.</p> <p>This is used in conjunction with COUPLED_AXES to indicate the way they are interacting.</p> <p>The coupling of the axes MUST be viewed from the perspective of a specified axis. Therefore, a MASTER coupling indicates that this axis is the master for the COUPLED_AXES.</p> <p>AxisCoupling MUST be provided for each axis element associated with a set of axes defined by the COUPLED_AXES data item element defined in the MTConnectDevices document.</p> <p><i>Valid Data Values:</i></p> <p>TANDEM: The axes are physically connected to each other and operate as a single unit.</p> <p>SYNCHRONOUS: The axes are not physically connected to each other but are operating together in lockstep.</p> <p>MASTER: The axis is the master of the CoupledAxes</p> <p>SLAVE: The axis is a slave to the CoupledAxes</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
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 <code>AxisFeedrateOverride</code> is expressed as a percentage of the designated feedrate for the axis.</p> <p>Subtypes of <code>AxisFeedrateOverride</code> are JOG, PROGRAMMED, and RAPID.</p> <p>If a <code>subType</code> is not specified, the reported value for the data MUST default to the <code>subType</code> of PROGRAMMED.</p> <p>The <i>Valid Data Value</i> MUST be a floating-point number.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
AXIS_ INTERLOCK	AxisInterlock	<p>An indicator of the state of the axis lockout function when power has been removed and the axis is allowed to move freely.</p> <p><i>Valid Data Values:</i></p> <p>ACTIVE: The axis lockout function is activated, power has been removed from the axis, and the axis is allowed to move freely.</p> <p>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.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
AXIS_STATE	AxisState	<p>An indicator of the controlled state of a Linear or Rotary component representing an axis.</p> <p><i>Valid Data Values:</i></p> <p>HOME: The axis is in its home position.</p> <p>TRAVEL: The axis is in motion</p> <p>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.</p> <p>STOPPED: The axis is stopped</p>
BLOCK	Block	<p>The line of code or command being executed by a Controller <i>Structural Element</i>.</p> <p>Block MUST include the entire expression for a line of program code, including all parameters</p> <p>The <i>Valid Data Value</i> MUST be a text string.</p>
BLOCK_COUNT	BlockCount	<p>The total count of the number of blocks of program code that have been executed since execution started.</p> <p>The <i>Valid Data Value</i> MUST be an integer.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
CHUCK_ INTERLOCK	ChuckInterlock	<p>An indication of the state of an interlock function or control logic state intended to prevent the associated CHUCK component from being operated.</p> <p>A CHUCK component or composition element may be controlled by more than one type of ChuckInterlock function. When the</p> <p>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</p> <p>ChuckInterlock function SHOULD be further characterized by specifying a subType of MANUAL_UNCLAMP.</p> <p><i>Valid Data Values:</i></p> <p>ACTIVE: The chuck cannot be unclamped</p> <p>INACTIVE: The chuck can be unclamped.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
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 mechanism in place within a piece of equipment.</p> <p><i>Valid Data Values:</i></p> <p>OPEN: The CHUCK component or composition element is open to the point of a positive confirmation</p> <p>CLOSED: The CHUCK component or composition element is closed to the point of a positive confirmation</p> <p>UNLATCHED: The CHUCK component or composition element is not closed to the point of a positive confirmation and not open to the point of a positive confirmation. It is in an intermediate position.</p>
CODE	Code	DEPRECATED in Version 1.1.

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
COMPOSITION_ STATE	CompositionState	<p>An indication of the operating condition of a mechanism represented by a <code>Composition</code> type element.</p> <p>Subtypes of <code>CompositionState</code> are <code>ACTION</code>, <code>LATERAL</code>, <code>MOTION</code>, <code>SWITCHED</code>, and <code>VERTICAL</code>.</p> <p>A subType MUST be provided.</p> <p><i>Valid Data Values for subType <code>ACTION</code> are:</i></p> <p>ACTIVE: The <code>Composition</code> element is operating</p> <p>INACTIVE: The <code>Composition</code> element is not operating.</p> <p><i>Valid Data Values for subType <code>LATERAL</code> are:</i></p> <p>RIGHT : The position of the <code>Composition</code> element is oriented to the right to the point of a positive confirmation</p> <p>LEFT : The position of the <code>Composition</code> element is oriented to the left to the point of a positive confirmation</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
COMPOSITION_ STATE (Continued)	CompositionState	<p><i>Valid Data Values</i> for subType SWITCHED are:</p> <p>ON : The activation state of the Composition element is in an ON condition, it is operating, or it is powered.</p> <p>OFF : The activation state of the Composition element is in an OFF condition, it is not operating, or it is not powered. <i>Valid Data Values</i> for subType VERTICAL are:</p> <p>UP : The position of the Composition element is oriented in an upward direction to the point of a positive confirmation</p> <p>DOWN : The position of the Composition element is oriented in a downward direction to the point of a positive confirmation</p> <p>TRANSITIONING : The position of the Composition element is not oriented in an upward direction to the point of a positive confirmation and is not oriented in a downward direction to the point of a positive confirmation. It is in an intermediate position.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
COMPOSITION_ STATE (Continued)	CompositionState	<p>TRANSITIONING : The position of the Composition element is not oriented to the right to the point of a positive confirmation and is not oriented to the left to the point of a positive confirmation. It is in an intermediate position.</p> <p><i>Valid Data Values</i> for subType MOTION are:</p> <p>OPEN: The position of the Composition element is open to the point of a positive confirmation</p> <p>CLOSED: The position of the Composition element is closed to the point of a positive confirmation</p> <p>UNLATCHED: The position of the Composition element is not open to the point of a positive confirmation and is not closed to the point of a positive confirmation. It is in an intermediate position.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
CONTROLLER_ MODE	ControllerMode	<p>The current operating mode of the Controller component.</p> <p><i>Valid Data Values:</i></p> <p>AUTOMATIC: The controller is configured to automatically execute a program.</p> <p>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.</p> <p>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.</p> <p>SEMI_AUTOMATIC: The controller is operating in a mode that restricts the active program from processing its next process step without operator intervention.</p> <p>EDIT: The controller is currently functioning as a programming device and is not capable of executing an active program.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
CONTROLLER_ MODE_ OVERRIDE	ControllerModeOverride	<p>A setting or operator selection that changes the behavior of a piece of equipment.</p> <p>Subtypes of ControllerModeOverride are DRY_RUN, SINGLE_BLOCK, MACHINE_AXIS_LOCK, OPTIONAL_STOP, and TOOL_CHANGE_STOP.</p> <p>A subType MUST always be specified.</p> <p><i>Valid Data Values:</i></p> <p>ON : The indicator of the ControllerModeOverride is in the ON state and the mode override is active.</p> <p>OFF : The indicator of the ControllerModeOverride is in the OFF state and the mode override is inactive</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
COUPLED_AXES	CoupledAxes	<p>Refers to the 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>The <i>Valid Data Value</i> reported SHOULD be a space-delimited set of axes names. The names returned SHOULD match the name attribute of the <code>Linear</code> or <code>Rotary Structural Elements</code> defined in the <code>MTConnectDevices</code> document that this <code>Event</code> element represents. If name is not available, <code>nativeName</code> MUST be returned to identify the <code>Linear</code> or <code>Rotary Structural Elements</code>.</p> <p>Example:</p> <pre><CoupledAxes ...>Y1 Y2</CoupledAxes></pre>
DATE_CODE	DateCode	<p>The time and date code associated with a material or other physical item.</p> <p>Subtypes of <code>DateCode</code> are <code>MANUFACTURE</code>, <code>EXPIRATION</code>, and <code>FIRST_USE</code>.</p> <p>A <code>subType</code> MUST always be specified.</p> <p><code>DateCode</code> MUST be reported in ISO 8601 format.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
DEVICE_UUID	DeviceUuid	<p>The identifier of another piece of equipment that is temporarily associated with a component of this piece of equipment to perform a particular function.</p> <p><i>Valid Data Values</i> are the value of the UUID attribute of the associated device - a NMTOKEN XML type.</p>
DIRECTION	Direction	<p>The direction of motion.</p> <p>Subtypes of <code>Direction</code> are <code>ROTARY</code> and <code>LINEAR</code>.</p> <p><i>Valid Data Values</i> for subType <code>ROTARY</code> are as follows:</p> <p><code>CLOCKWISE</code>: Clockwise rotation using the right-hand rule.</p> <p><code>COUNTER_CLOCKWISE</code>: Counter-clockwise rotation using the right-hand rule.</p> <p><code>NONE</code>: No direction.</p> <p><i>Valid Data Values</i> for subType <code>LINEAR</code> are as follows:</p> <p><code>POSITIVE</code>: Linear position is increasing.</p> <p><code>NEGATIVE</code>: Linear position is decreasing.</p> <p><code>NONE</code>: No direction.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
DOOR_STATE	DoorState	<p>The operational state of a DOOR type component or composition element.</p> <p><i>Valid Data Values:</i></p> <p>OPEN: The DOOR is open to the point of a positive confirmation</p> <p>CLOSED: The DOOR is closed to the point of a positive confirmation</p> <p>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.</p>
EMERGENCY_-STOP	EmergencyStop	<p>The current state of the emergency stop signal for a piece of equipment, controller path, or any other component or subsystem of a piece of equipment.</p> <p><i>Valid Data Values:</i></p> <p>ARMED : The emergency stop circuit is complete and the piece of equipment, component, or composition element is allowed to operate.</p> <p>TRIGGERED : The emergency stop circuit is open and the operation of the piece of equipment, component, or composition element is inhibited.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
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>Subtypes of EndOfBar are PRIMARY and AUXILIARY.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subType of PRIMARY.</p> <p><i>Valid Data Values:</i></p> <p>YES : The EndOfBar has been reached.</p> <p>NO : The EndOfBar has not been reached.</p>
EQUIPMENT_MODE	EquipmentMode	<p>An indication that a piece of equipment, or a sub-part of a piece of equipment, is performing specific types of activities.</p> <p>Subtypes of EquipmentMode are LOADED, WORKING, OPERATING, and POWERED.</p> <p>A subType MUST always be specified.</p> <p><i>Valid Data Values:</i></p> <p>ON : The equipment is functioning in the mode designated by the subType.</p> <p>OFF : The equipment is not functioning in the mode designated by the subType.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
EXECUTION	Execution	<p>The execution status of a component.</p> <p><i>Valid Data Values:</i></p> <p>READY: The component is ready to execute instructions. It is currently idle.</p> <p>ACTIVE: The component is actively executing an instruction.</p> <p>INTERRUPTED: The component suspends the execution of the program due to an external signal. Action is required to resume execution.</p> <p>WAIT: The component suspends execution while a secondary operation executes. Execution resumes automatically once the secondary operation completes.</p> <p>FEED_HOLD: The motion of the active axes are commanded to stop at their current position.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
EXECUTION (continued)	Execution	<p>STOPPED: The component program is not READY to execute.</p> <p>OPTIONAL_STOP: A command from the program has intentionally interrupted execution. The component MAY have another state that indicates if the execution is interrupted or the execution ignores the interrupt instruction.</p> <p>PROGRAM_STOPPED: A command from the program has intentionally interrupted execution. Action is required to resume execution.</p> <p>PROGRAM_COMPLETED: The program completed execution.</p>
FIRMWARE	Firmware	<p>The embedded software of a component.</p> <p>Subtypes of FIRMWARE are LICENSE, VERSION, RELEASE_DATE, INSTALL_DATE, and MANUFACTURER.</p> <p>The <i>Valid Data Value</i> MUST be a text string.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
FUNCTIONAL_ MODE	FunctionalMode	<p>The current intended production status of the device or component.</p> <p>Typically, the FunctionalMode SHOULD be associated with the Device <i>Structural Element</i>, but it MAY be associated with any <i>Structural Element</i> in the XML document.</p> <p><i>Valid Data Values:</i></p> <p>PRODUCTION : The Device element or another <i>Structural Element</i> is currently producing product, ready to produce product, or its current intended use is to be producing product.</p> <p>SETUP : The Device element or another <i>Structural Element</i> is not currently producing product. It is being prepared or modified to begin production of product.</p> <p>TEARDOWN : The Device element or another <i>Structural Element</i> is not currently producing product. Typically, it has completed the production of a product and is being modified or returned to a neutral state such that it may then be prepared to begin production of a different product.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
FUNCTIONAL_ MODE (Continued)	FunctionalMode	<p>MAINTENANCE : The Device element or another <i>Structural Element</i> is not currently producing product. It is currently being repaired, waiting to be repaired, or has not yet been returned to a normal production status after maintenance has been performed.</p> <p>PROCESS_DEVELOPMENT : The Device element or another <i>Structural Element</i> is being used to prove-out a new process, testing of equipment or processes, or any other active use that does not result in the production of product.</p>
HARDNESS	Hardness	<p>The measurement of the hardness of a material.</p> <p>Subtypes of Hardness are ROCKWELL, VICKERS, SHORE, BRINELL, LEEB, and MOHS.</p> <p>A subType MUST always be specified.</p> <p>The <i>Valid Data Value</i> MUST be a floating-point number.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
HARDWARE	Hardware	<p>The hardware of a component.</p> <p>Subtypes of HARDWARE are LICENSE, VERSION, RELEASE_DATE, INSTALL_DATE, and MANUFACTURER.</p> <p>The <i>Valid Data Value</i> MUST be a text string.</p>
INTERFACE_ STATE	InterfaceState	<p>The current functional or operational state of an Interface type element indicating whether the <i>Interface</i> is active or not currently functioning.</p> <p><i>Valid Data Values:</i></p> <p>ENABLED: The <i>Interface</i> is currently operational and performing as expected.</p> <p>DISABLED: The Interface is currently not operational.</p> <p>When the INTERFACE_STATE is DISABLED, the state of all data items that are specific for the <i>Interaction Model</i> associated with that <i>Interface</i> MUST be set to NOT_READY.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
LIBRARY	Library	<p>The software library on a component.</p> <p>Subtypes of LIBRARY are LICENSE, VERSION, RELEASE_DATE, INSTALL_DATE, and MANUFACTURER.</p> <p>The <i>Valid Data Value</i> MUST be a text string.</p>
LINE	Line	DEPRECATED in Version 1.4.0.
LINE_LABEL	LineLabel	<p>An optional identifier for a BLOCK of code in a PROGRAM.</p> <p>The <i>Valid Data Value</i> MUST be any text string.</p>
LINE_NUMBER	LineNumber	<p>A reference to the position of a block of program code within a control program.</p> <p>Subtypes of LineNumber are ABSOLUTE and INCREMENTAL.</p> <p>A subType MUST always be specified.</p> <p>The <i>Valid Data Value</i> MUST be an integer.</p>
MATERIAL	Material	<p>The identifier of a material used or consumed in the manufacturing process.</p> <p>The <i>Valid Data Value</i> MUST be any text string.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
MATERIAL_ LAYER	MaterialLayer	<p>Designates the layers of material applied to a part or product as part of an additive manufacturing process.</p> <p>Subtypes of MaterialLayer are ACTUAL and TARGET.</p> <p>If a subType is not specified, the reported value for the data MUST default to the subtype of ACTUAL.</p> <p>The <i>Valid Data Value</i> MUST be an integer.</p>
MESSAGE	Message	<p>Any text string of information to be transferred from a piece of equipment to a client software application.</p> <p>The <i>Valid Data Value</i> MUST be any text string.</p>
NETWORK	Network	<p>Network details of a component.</p> <p>Subtypes of NETWORK are IPV4_ADDRESS, IPV6_ADDRESS, GATEWAY, SUBNET_MASK, VLAN_ID, MAC_ADDRESS, and WIRELESS.</p> <p>The <i>Valid Data Value</i> MUST be a text string.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
OPERATING_ SYSTEM	OperatingSystem	<p>The Operating System of a component.</p> <p>Subtypes of OPERATING_SYSTEM are LICENSE, VERSION, RELEASE_DATE, INSTALL_DATE, and MANUFACTURER.</p> <p>The <i>Valid Data Value</i> MUST be a text string.</p> <p>When specified with no subType, use the following vocabulary or specify the name of the operating system:</p> <ul style="list-style-type: none"> - WINDOWS - LINUX - MACINTOSH - PROPRIETARY
OPERATOR_ID	OperatorId	<p>The identifier of the person currently responsible for operating the piece of equipment.</p> <p>The <i>Valid Data Value</i> MAY be any text string.</p> <p>DEPRECATION WARNING : May be deprecated in the future. See USER below.</p>
PALLET_ID	PalletId	<p>The identifier for a pallet.</p> <p>The <i>Valid Data Value</i> MAY be any text string.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
PART_COUNT	PartCount	<p>The aggregate count of parts.</p> <p>When the <code>discrete</code> attribute is <code>true</code>, the value represents the number of parts since the previous occurrence of the event.</p> <p>Subtypes of <code>PartCount</code> are <code>ALL</code>, <code>GOOD</code>, <code>BAD</code>, <code>TARGET</code>, and <code>REMAINING</code>.</p> <p>The <i>Valid Data Value</i> MUST be numeric.</p>
PART_DETECT	PartDetect	<p>An indication designating whether a part or work piece has been detected or is present.</p> <p>The <i>Valid Data Value</i> MUST be:</p> <p style="padding-left: 40px;"><code>PRESENT</code>: if a part or work piece has been detected or is present.</p> <p style="padding-left: 40px;"><code>NOT_PRESENT</code>: if a part or work piece is not detected or is not present.</p>
PART_ID	PartId	<p>An identifier of a part in a manufacturing operation.</p> <p>The <i>Valid Data Value</i> MAY be any text string.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
PART_NUMBER	PartNumber	<p>An identifier of a part or product moving through the manufacturing process.</p> <p>The <i>Valid Data Value</i> MUST be a text string.</p> <p>DEPRECATION WARNING : May be deprecated in the future.</p>
PATH_- FEEDRATE_- OVERRIDE	PathFeedrateOverride	<p>The value of a signal or calculation issued to adjust the feedrate for the axes associated with a Path component that may represent a single axis or the coordinated movement of multiple axes.</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, the reported value for the data MUST default to the subType of PROGRAMMED.</p> <p>The <i>Valid Data Value</i> MUST be a floating-point number.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
PATH_MODE	PathMode	<p>Describes the operational relationship between a <i>Path Structural Element</i> and another <i>Path Structural Element</i> for pieces of equipment comprised of multiple logical groupings of controlled axes or other logical operations.</p> <p><i>Valid Data Values:</i></p> <p>INDEPENDENT : The path is operating independently and without the influence of another path.</p> <p>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</p> <p>SYNCHRONOUS: The axes associated with the path are following the motion of the MASTER type path.</p> <p>MIRROR : The axes associated with the path are mirroring the motion of the MASTER path. When PathMode is not specified, the operational mode of the path MUST be interpreted as INDEPENDENT .</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
POWER_STATE	PowerState	<p>The indication of the status of the source of energy for a <i>Structural Element</i> to allow it to perform its intended function or the state of an enabling signal providing permission for the <i>Structural Element</i> to perform its functions.</p> <p>Subtypes of PowerState are LINE and CONTROL.</p> <p>When the subType is LINE, PowerState represents the primary source of energy for a <i>Structural Element</i>.</p> <p>When the subType is CONTROL, PowerState represents an enabling signal providing permission for the <i>Structural Element</i> to perform its function(s).</p> <p>If a subType is not specified, the reported value for the data MUST default to the subType of LINE.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
POWER_STATE (Continued)	PowerState	<p><i>Valid Data Values:</i></p> <p>ON : The source of energy for a <i>Structural Element</i> or the enabling signal providing permission for the <i>Structural Element</i> to perform its function(s) is present and active.</p> <p>OFF : The source of energy for a <i>Structural Element</i> or the enabling signal providing permission for the <i>Structural Element</i> to perform its function(s) is not present or is disconnected.</p> <p>DEPRECATION WARNING : PowerState may be deprecated in the future.</p>
POWER_STATUS	PowerStatus	DEPRECATED in Version 1.1.0.
PROCESS_TIME	ProcessTime	<p>The time and date associated with an activity or event.</p> <p>Subtypes of ProcessTime are START, COMPLETE, and TARGET_COMPLETION.</p> <p>A subType MUST always be specified.</p> <p>ProcessTime MUST be reported in ISO 8601 format.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
PROGRAM	Program	<p>The identity of the logic or motion program being executed.</p> <p>The <i>Valid Data Value</i> MUST be any text string.</p> <p>Subtypes of PROGRAM are SCHEDULE, MAIN and ACTIVE.</p> <p>If a subType is not specified, it is assumed to be MAIN.</p>
PROGRAM_- COMMENT	ProgramComment	<p>A comment or non-executable statement in the control program.</p> <p>The <i>Valid Data Value</i> MUST be any text string.</p> <p>Subtypes of PROGRAM_COMMENT are SCHEDULE, MAIN and ACTIVE.</p> <p>If a subType is not specified, it is assumed to be MAIN.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
PROGRAM_EDIT	ProgramEdit	<p>An indication of the status of the Controller components program editing mode.</p> <p>On many controls, a program can be edited while another program is currently being executed.</p> <p>ProgramEdit provides an indication of whether the controller is being used to edit programs in either case.</p> <p><i>Valid Data Values:</i></p> <p>ACTIVE: The controller is in the program edit mode.</p> <p>READY : The controller is capable of entering the program edit mode and no function is inhibiting a change to that mode.</p> <p>NOT_READY : A function is inhibiting the controller from entering the program edit mode.</p>
PROGRAM_- EDIT_NAME	ProgramEditName	<p>The name of the program being edited.</p> <p>This is used in conjunction with PROGRAM_EDIT when in ACTIVE state.</p> <p>The <i>Valid Data Value</i> MUST be a text string.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
PROGRAM_ HEADER	ProgramHeader	<p>The non-executable header section of the control program.</p> <p>Subtypes of PROGRAM_HEADER are SCHEDULE, MAIN, and ACTIVE.</p> <p>The <i>Valid Data Value</i> MUST be a text string.</p>
PROGRAM_ LOCATION	ProgramLocation	<p>The Uniform Resource Identifier (URI) for the source file associated with PROGRAM.</p> <p>The <i>Valid Data Value</i> MUST be any text string.</p> <p>A subType MUST always be specified.</p> <p>Subtypes of PROGRAM_LOCATION are SCHEDULE, MAIN, and ACTIVE.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
PROGRAM_- LOCATION_- TYPE	ProgramLocationType	<p>Defines whether the logic or motion program defined by PROGRAM is being executed from the local memory of the controller or from an outside source.</p> <p>A subType MUST always be specified.</p> <p>Subtypes of PROGRAM_-LOCATION_-TYPE are SCHEDULE, MAIN, and ACTIVE.</p> <p><i>Valid Data Values are:</i></p> <p> LOCAL: Managed by the controller.</p> <p> EXTERNAL: Not managed by the controller.</p>
PROGRAM_- NEST_LEVEL	ProgramNestLevel	<p>An indication of the nesting level within a control program that is associated with the code or instructions that is currently being executed.</p> <p>If an initial value is not defined, the nesting level associated with the highest or initial nesting level of the program MUST default to zero (0).</p> <p>The value reported for ProgramNestLevel MUST be an integer.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
ROTARY_MODE	RotaryMode	<p>The current operating mode for a Rotary type axis.</p> <p><i>Valid Data Values:</i></p> <p>SPINDLE: The axis is functioning as a spindle. Generally, it is configured to rotate at a defined speed.</p> <p>INDEX: The axis is configured to index to a set of fixed positions or to incrementally index by a fixed amount.</p> <p>CONTOUR: The position of the axis is being interpolated as part of the PathPosition defined by the Controller <i>Structural Element</i>.</p>
ROTARY_- VELOCITY_- OVERRIDE	RotaryVelocityOverride	<p>The value of a command issued to adjust the programmed velocity for a Rotary type axis.</p> <p>This command represents a percentage change to the velocity calculated by a logic or motion program or set by a switch for a Rotary type axis.</p> <p>RotaryVelocityOverride is expressed as a percentage of the programmed RotaryVelocity.</p> <p>The <i>Valid Data Value</i> MUST be a floating-point number.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
ROTATION	Rotation	<p>A three space angular rotation relative to a coordinate system.</p> <p>The value MUST be three floating-point numbers representing rotations around the X, Y, and Z axes in degrees.</p> <p>The values in XML are space delimited.</p>
SERIAL_- NUMBER	SerialNumber	<p>The serial number associated with a Component, Asset, or Device. The <i>Valid Data Value</i> MUST be a text string.</p>
SPINDLE_- INTERLOCK	SpindleInterlock	<p>An indication of the status of the spindle for a piece of equipment when power has been removed and it is free to rotate.</p> <p><i>Valid Data Values:</i></p> <p>ACTIVE: Power has been removed and the spindle cannot be operated.</p> <p>INACTIVE: Spindle has not been deactivated.</p>
TOOL_ASSET_- ID	ToolAssetId	<p>The identifier of an individual tool asset. The <i>Valid Data Value</i> MUST be a text string.</p>
TOOL_GROUP	ToolGroup	<p>An identifier for the tool group associated with a specific tool. Commonly used to designate spare tools.</p> <p>The <i>Valid Data Value</i> MUST be any text string.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
TOOL_ID	ToolId	DEPRECATED in Version 1.2.0. See <code>TOOL_ASSET_ID</code> . The identifier of the tool currently in use for a given Path.
TOOL_NUMBER	ToolNumber	The identifier assigned by the Controller component to a cutting tool when in use by a piece of equipment. The <i>Valid Data Value</i> MUST be a text string.
TOOL_OFFSET	ToolOffset	A reference to the tool offset variables applied to the active cutting tool. Subtypes of <code>ToolOffset</code> are <code>RADIAL</code> and <code>LENGTH</code> . DEPRECATED in V1.5 A subType MUST always be specified. The <i>Valid Data Value</i> MUST be a text string.
TRANSLATION	Translation	A three space linear translation relative to a coordinate system. The value MUST be three floating-point numbers translation along the X, Y, and Z axes in millimeters. The values in XML are space delimited.

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
USER	User	<p>The identifier of the person currently responsible for operating the piece of equipment.</p> <p>Subtypes of <code>User</code> are <code>OPERATOR</code>, <code>MAINTENANCE</code>, and <code>SET_UP</code>.</p> <p>A <code>subType</code> MUST always be specified.</p> <p>The <i>Valid Data Value</i> MUST be any text string.</p>
VARIABLE	Variable	<p>A data value whose meaning may change over time due to changes in the operation of a piece of equipment or the process being executed on that piece of equipment.</p> <p>The <i>Valid Data Value</i> MUST be a string.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
WAIT_STATE	WaitState	<p>An indication of the reason that EXECUTION is reporting a value of WAIT.</p> <p><i>Valid Data Values are:</i></p> <p>POWERING_UP: An indication that execution is waiting while the equipment is powering up and is not currently available to begin producing parts or products.</p> <p>POWERING_DOWN: An indication that the execution is waiting while the equipment is powering down but has not fully reached a stopped state.</p> <p>PART_LOAD: An indication that the execution is waiting while one or more discrete workpieces are being loaded.</p> <p>PART_UNLOAD: An indication that the execution is waiting while one or more discrete workpieces are being unloaded.</p> <p>TOOL_LOAD: An indication that the execution is waiting while a tool or tooling is being loaded.</p> <p>TOOL_UNLOAD: An indication that the execution is waiting while a tool or tooling is being unloaded.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
WAIT_STATE (Continued)	WaitState	<p>MATERIAL_LOAD: An indication that the execution is waiting while bulk material or the container for bulk material used in the production process is being loaded. Bulk material includes those materials from which multiple workpieces may be created.</p> <p>MATERIAL_UNLOAD: An indication that the execution is waiting while bulk material or the container for bulk material used in the production process is being unloaded. Bulk material includes those materials from which multiple workpieces may be created.</p> <p>SECONDARY_PROCESS: An indication that the execution is waiting while another process is completed before the execution can resume.</p> <p>PAUSING: An indication that the execution is waiting while the equipment is pausing but the piece of equipment has not yet reached a fully paused state.</p> <p>RESUMING: An indication that the execution is waiting while the equipment is resuming the production cycle but has not yet resumed execution.</p>

Continuation of Table 26: Element Names for Event		
DataItem Type	Element Name	Description
WIRE	Wire	The identifier for the type of wire used as the cutting mechanism in Electrical Discharge Machining or similar processes. The <i>Valid Data Value</i> MUST be any text string.
WORKHOLDING_ID	WorkholdingId	The identifier for the current workholding or part clamp in use by a piece of equipment. The <i>Valid Data Value</i> MUST be a text string.
WORK_OFFSET	WorkOffset	A reference to the offset variables for a work piece or part associated with a Path in a Controller type component. The <i>Valid Data Value</i> MUST be a text string.

1144 6.3 Types of Condition Elements

1145 As described in *Section 5.8 - Condition Data Entity*, *Condition Data Entities* are re-
 1146 ported differently from other data item types. They are reported based on the *Fault State*
 1147 for each *Condition*. Unlike *Sample* and *Event* data items that are identified by their
 1148 *Element Name*, *Condition* data items are defined by the type and subType (where
 1149 applicable) attributes defined for each *Condition*.

1150 The type and subType (where applicable) attributes for a *Condition* element **MAY**
 1151 be any of the type and subType attributes defined for *SAMPLE* category or *EVENT*
 1152 category data item listed in the *Devices Information Model*.

1153 Table *Section 5.8.1 - Element Names for Condition* lists additional *Condition Data En-*
 1154 *tities* that have been defined to represent the health and fault status of *Structural Elements*.
 1155 The table defines the type attribute for each of these additional *Condition* category

1156 elements that **MAY** be reported in the `MTCConnectStreams` document.

Table 27: Element Names for Condition

DataItem Type	Description
ACTUATOR	An indication of a fault associated with an actuator.
CHUCK_INTERLOCK	An indication of the operational condition of the interlock function for an electronically controller chuck.
COMMUNICATIONS	An indication that the piece of equipment has experienced a communications failure.
DATA_RANGE	An indication that the value of the data associated with a measured value or a calculation is outside of an expected range.
DIRECTION	An indication of a fault associated with the direction of motion of a <i>Structural Element</i> .
END_OF_BAR	An indication that the end of a piece of bar stock has been reached.
HARDWARE	An indication of a fault associated with the hardware subsystem of the <i>Structural Element</i> .
INTERFACE_STATE	An indication of the operation condition of an <i>Interface</i> component.
LOGIC_PROGRAM	An indication that an error occurred in the logic program or programmable logic controller (PLC) associated with a piece of equipment.
MOTION_PROGRAM	An indication that an error occurred in the motion program associated with a piece of equipment.
SYSTEM	An indication of a fault associated with a piece of equipment or component that cannot be classified as a specific type.

1157 Appendices

1158 A Bibliography

- 1159 Engineering Industries Association. *EIA Standard - EIA-274-D*, Interchangeable Variable,
 1160 Block Data Format for Positioning, Contouring, and Contouring/Positioning Numerically
 1161 Controlled Machines. Washington, D.C. 1979.
- 1162 ISO TC 184/SC4/WG3 N1089. *ISO/DIS 10303-238*: Industrial automation systems and
 1163 integration Product data representation and exchange Part 238: Application Protocols: Ap-
 1164 plication interpreted model for computerized numerical controllers. Geneva, Switzerland,
 1165 2004.
- 1166 International Organization for Standardization. *ISO 14649*: Industrial automation sys-
 1167 tems and integration – Physical device control – Data model for computerized numerical
 1168 controllers – Part 10: General process data. Geneva, Switzerland, 2004.
- 1169 International Organization for Standardization. *ISO 14649*: Industrial automation sys-
 1170 tems and integration – Physical device control – Data model for computerized numerical
 1171 controllers – Part 11: Process data for milling. Geneva, Switzerland, 2000.
- 1172 International Organization for Standardization. *ISO 6983/1* – Numerical Control of ma-
 1173 chines – Program format and definition of address words – Part 1: Data format for posi-
 1174 tioning, line and contouring control systems. Geneva, Switzerland, 1982.
- 1175 Electronic Industries Association. *ANSI/EIA-494-B-1992*, 32 Bit Binary CL (BCL) and
 1176 7 Bit ASCII CL (ACL) Exchange Input Format for Numerically Controlled Machines.
 1177 Washington, D.C. 1992.
- 1178 National Aerospace Standard. *Uniform Cutting Tests - NAS Series: Metal Cutting Equip-*
 1179 *ment Specifications*. Washington, D.C. 1969.
- 1180 International Organization for Standardization. *ISO 10303-11*: 1994, Industrial automa-
 1181 tion systems and integration Product data representation and exchange Part 11: Descrip-
 1182 tion methods: The EXPRESS language reference manual. Geneva, Switzerland, 1994.
- 1183 International Organization for Standardization. *ISO 10303-21*: 1996, Industrial automa-
 1184 tion systems and integration – Product data representation and exchange – Part 21: Imple-
 1185 mentation methods: Clear text encoding of the exchange structure. Geneva, Switzerland,
 1186 1996.
- 1187 H.L. Horton, F.D. Jones, and E. Oberg. *Machinery's Handbook*. Industrial Press, Inc.

- 1188 New York, 1984.
- 1189 International Organization for Standardization. *ISO 841-2001: Industrial automation sys-*
1190 *tems and integration - Numerical control of machines - Coordinate systems and motion*
1191 *nomenclature*. Geneva, Switzerland, 2001.
- 1192 *ASME B5.57: Methods for Performance Evaluation of Computer Numerically Controlled*
1193 *Lathes and Turning Centers*, 1998.
- 1194 *ASME/ANSI B5.54: Methods for Performance Evaluation of Computer Numerically Con-*
1195 *trolled Machining Centers*. 2005.
- 1196 OPC Foundation. *OPC Unified Architecture Specification, Part 1: Concepts Version 1.00.*
1197 July 28, 2006.
- 1198 IEEE STD 1451.0-2007, *Standard for a Smart Transducer Interface for Sensors and Ac-*
1199 *tuators – Common Functions, Communication Protocols, and Transducer Electronic Data*
1200 *Sheet (TEDS) Formats*, IEEE Instrumentation and Measurement Society, TC-9, The In-
1201 *stitute of Electrical and Electronics Engineers, Inc., New York, N.Y. 10016, SH99684,*
1202 *October 5, 2007.*
- 1203 IEEE STD 1451.4-1994, *Standard for a Smart Transducer Interface for Sensors and Ac-*
1204 *tuators – Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet*
1205 *(TEDS) Formats*, IEEE Instrumentation and Measurement Society, TC-9, The Institute of
1206 *Electrical and Electronics Engineers, Inc., New York, N.Y. 10016, SH95225, December*
1207 *15, 2004.*