



MTConnect[®] Standard

Part 3.0 – Observation Information Model

Version 2.0.0

Prepared for: MTConnect Institute
Prepared from: MTConnectSysMLModel.xml
Prepared on: May 24, 2022

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.

The normative XMI is located at the following URL: `MTConnectSysMLModel.xml`

Table of Contents

1	Purpose of This Document	2
2	Terminology and Conventions	3
2.1	General Terms	3
2.2	Information Model Terms	9
2.3	Protocol Terms	10
2.4	HTTP Terms	12
2.5	XML Terms	14
2.6	MTConnect Terms	15
2.7	Acronyms	16
2.8	MTConnect References	28
3	Observation Information Model	29
3.1	DeviceStream	30
3.1.1	Value Properties of DeviceStream	30
3.1.2	Part Properties of DeviceStream	31
3.2	ComponentStream	32
3.2.1	Value Properties of ComponentStream	32
3.2.2	Reference Properties of ComponentStream	33
3.3	Observation	33
3.3.1	Value Properties of Observation	36
3.4	Sample	37
3.4.1	Value Properties of Sample	38
3.5	Event	40
3.5.1	Value Properties of Event	40
3.6	Condition	42
3.6.1	Value Properties of Condition	42
4	Representations	47
4.1	Representation	47
4.2	Value	47
4.3	TimeSeries	48
4.3.1	Value Properties of TimeSeries	49
4.4	<<deprecated>>Discrete	49
4.5	DataSet	49
4.5.1	Value Properties of DataSet	52
4.5.2	Part Properties of DataSet	52
4.6	Table	53
4.6.1	Value Properties of Table	54
4.6.2	Part Properties of Table	55
4.7	Entry	55

4.7.1	Value Properties of Entry	56
4.8	TableEntry	56
4.8.1	Value Properties of TableEntry	56
4.8.2	Part Properties of TableEntry	57
4.9	Cell	57
4.9.1	Value Properties of Cell	58
5	Observation Types	59
5.1	Condition Types	59
5.1.1	Actuator	59
5.1.2	Communications	59
5.1.3	DataRange	59
5.1.4	LogicProgram	60
5.1.5	MotionProgram	60
5.1.6	System	60
5.2	Event Types	60
5.2.1	ActivationCount	60
5.2.2	ActiveAxes	61
5.2.3	ActuatorState	61
5.2.4	AdapterSoftwareVersion	62
5.2.5	AdapterURI	62
5.2.6	<<deprecated>>Alarm	62
5.2.7	AlarmLimit	63
5.2.8	Application	64
5.2.9	AssetChanged	64
5.2.10	AssetCount	65
5.2.11	AssetRemoved	65
5.2.12	Availability	66
5.2.13	AxisCoupling	66
5.2.14	AxisFeedrateOverride	66
5.2.15	AxisInterlock	67
5.2.16	AxisState	67
5.2.17	Block	68
5.2.18	BlockCount	68
5.2.19	ChuckInterlock	68
5.2.20	ChuckState	69
5.2.21	ClockTime	69
5.2.22	<<deprecated>>Code	70
5.2.23	CompositionState	70
5.2.24	ConnectionStatus	72
5.2.25	ControlLimit	72
5.2.26	ControllerMode	73

5.2.27	ControllerModeOverride	74
5.2.28	CoupledAxes	75
5.2.29	CycleCount	75
5.2.30	DateCode	76
5.2.31	DeactivationCount	77
5.2.32	DeviceAdded	78
5.2.33	DeviceChanged	78
5.2.34	DeviceRemoved	78
5.2.35	DeviceUuid	78
5.2.36	Direction	78
5.2.37	DoorState	80
5.2.38	EmergencyStop	80
5.2.39	EndOfBar	80
5.2.40	EquipmentMode	81
5.2.41	Execution	82
5.2.42	Firmware	83
5.2.43	FixtureId	84
5.2.44	FunctionalMode	84
5.2.45	Hardness	85
5.2.46	Hardware	86
5.2.47	Library	87
5.2.48	<<deprecated>>Line	87
5.2.49	LineLabel	88
5.2.50	LineNumber	88
5.2.51	LoadCount	88
5.2.52	LockState	89
5.2.53	MTConnectVersion	90
5.2.54	MaintenanceList	90
5.2.55	Material	92
5.2.56	MaterialLayer	92
5.2.57	Message	92
5.2.58	Network	92
5.2.59	OperatingMode	93
5.2.60	OperatingSystem	94
5.2.61	OperatorId	95
5.2.62	PalletId	95
5.2.63	PartCount	95
5.2.64	PartCountType	96
5.2.65	PartDetect	96
5.2.66	PartGroupId	97
5.2.67	PartId	97
5.2.68	PartKindId	97

5.2.69	<<deprecated>>PartNumber	98
5.2.70	PartProcessingState	98
5.2.71	PartStatus	99
5.2.72	PartUniqueId	100
5.2.73	PathFeedrateOverride	100
5.2.74	PathMode	101
5.2.75	PowerState	102
5.2.76	<<deprecated>>PowerStatus	102
5.2.77	ProcessAggregateId	103
5.2.78	ProcessKindId	103
5.2.79	ProcessOccurrenceId	104
5.2.80	ProcessState	104
5.2.81	ProcessTime	105
5.2.82	Program	105
5.2.83	ProgramComment	106
5.2.84	ProgramEdit	106
5.2.85	ProgramEditName	107
5.2.86	ProgramHeader	107
5.2.87	ProgramLocation	107
5.2.88	ProgramLocationType	108
5.2.89	ProgramNestLevel	109
5.2.90	RotaryMode	109
5.2.91	RotaryVelocityOverride	109
5.2.92	Rotation	109
5.2.93	SensorAttachment	110
5.2.94	SerialNumber	110
5.2.95	SpecificationLimit	110
5.2.96	SpindleInterlock	111
5.2.97	ToolAssetId	111
5.2.98	ToolGroup	111
5.2.99	<<deprecated>>ToolId	111
5.2.100	ToolNumber	112
5.2.101	ToolOffset	112
5.2.102	TransferCount	112
5.2.103	Translation	113
5.2.104	UnloadCount	113
5.2.105	User	114
5.2.106	ValveState	115
5.2.107	Variable	116
5.2.108	WaitState	116
5.2.109	Wire	117
5.2.110	WorkOffset	117

5.2.111	WorkholdingId	117
5.3	Sample Types	117
5.3.1	Acceleration	118
5.3.2	AccumulatedTime	118
5.3.3	<<deprecated>>Amperage	118
5.3.4	AmperageAC	119
5.3.5	AmperageDC	120
5.3.6	Angle	120
5.3.7	AngularAcceleration	121
5.3.8	AngularDeceleration	121
5.3.9	AngularVelocity	122
5.3.10	AssetUpdateRate	122
5.3.11	AxisFeedrate	122
5.3.12	CapacityFluid	123
5.3.13	CapacitySpatial	123
5.3.14	Concentration	123
5.3.15	Conductivity	123
5.3.16	CuttingSpeed	123
5.3.17	Deceleration	124
5.3.18	Density	124
5.3.19	DepositionAccelerationVolumetric	125
5.3.20	DepositionDensity	125
5.3.21	DepositionMass	125
5.3.22	DepositionRateVolumetric	126
5.3.23	DepositionVolume	126
5.3.24	Diameter	127
5.3.25	Displacement	127
5.3.26	ElectricalEnergy	127
5.3.27	EquipmentTimer	127
5.3.28	FillLevel	128
5.3.29	Flow	128
5.3.30	Frequency	128
5.3.31	<<deprecated>>GlobalPosition	129
5.3.32	HumidityAbsolute	129
5.3.33	HumidityRelative	129
5.3.34	HumiditySpecific	130
5.3.35	Length	130
5.3.36	<<deprecated>>Level	131
5.3.37	LinearForce	131
5.3.38	Load	131
5.3.39	Mass	131
5.3.40	ObservationUpdateRate	132

5.3.41	Openness	132
5.3.42	Orientation	132
5.3.43	PH	132
5.3.44	PathFeedrate	133
5.3.45	PathFeedratePerRevolution	133
5.3.46	PathPosition	134
5.3.47	Position	135
5.3.48	PowerFactor	135
5.3.49	Pressure	135
5.3.50	PressureAbsolute	136
5.3.51	PressurizationRate	136
5.3.52	ProcessTimer	136
5.3.53	Resistance	137
5.3.54	RotaryVelocity	137
5.3.55	SoundLevel	138
5.3.56	<<deprecated>>SpindleSpeed	138
5.3.57	Strain	139
5.3.58	Temperature	139
5.3.59	Tension	139
5.3.60	Tilt	139
5.3.61	Torque	140
5.3.62	Velocity	140
5.3.63	Viscosity	140
5.3.64	VoltAmpere	140
5.3.65	VoltAmpereReactive	140
5.3.66	<<deprecated>>Voltage	140
5.3.67	VoltageAC	141
5.3.68	VoltageDC	142
5.3.69	VolumeFluid	142
5.3.70	VolumeSpatial	143
5.3.71	Wattage	144
5.3.72	XDimension	144
5.3.73	YDimension	144
5.3.74	ZDimension	144
6	Profile	145
6.1	DataTypes	145
6.1.1	boolean	145
6.1.2	ID	145
6.1.3	string	145
6.1.4	float	145
6.1.5	dateTime	146

6.1.6	integer	146
6.1.7	xlinktype	146
6.1.8	xslang	146
6.1.9	SECOND	146
6.1.10	IDREF	146
6.1.11	xlinkhref	146
6.1.12	x509	147
6.1.13	int32	147
6.1.14	int64	147
6.1.15	version	147
6.1.16	uInt32	147
6.1.17	uInt64	147
6.2	Stereotypes	147
6.2.1	organizer	147
6.2.2	deprecated	148
6.2.3	extensible	148
6.2.4	informative	148
6.2.5	valueType	148
6.2.6	normative	148
6.2.7	observes	148
Appendices		150
A	Bibliography	150
B	XML Schema Diagrams	152
B.1	Observations Schema Diagrams	152
B.2	Representation Schema Diagrams	152
C	XML Examples	152
C.1	DeviceStream Example	153
C.2	Observations made for DataItem Example	154
C.3	Sample Example	155
C.4	Event Example	155
C.5	Condition Example	155
C.6	DataSet Example	156
C.7	Table Example	156

Table of Figures

Figure 1: Observations	34
Figure 2: DeviceStream Example	34
Figure 3: Observations made for DataItem Example	35
Figure 4: Sample Example	38
Figure 5: Event Example	41
Figure 6: Condition Example	44
Figure 7: Representation	47
Figure 8: TemperatureTimeSeries	48
Figure 9: VariableDataSet	50
Figure 10:DataSet Example	51
Figure 11:WorkOffsetTable	53
Figure 12:Table Example	54
Figure 13:DataTypes	145
Figure 14:Stereotypes	149
Figure 15:Streams Schema	152
Figure 16:DeviceStream Schema	152
Figure 17:ComponentStream Schema	157
Figure 18:Sample Schema	158
Figure 19:Event Schema	159
Figure 20:Condition Schema	160
Figure 21:AbsTimeSeries Schema	161
Figure 22:DataSet Schema	162
Figure 23:Entry Schema	163
Figure 24:Table Schema	164

List of Tables

Table 1: Value Properties of DeviceStream	31
Table 2: Part Properties of DeviceStream	31
Table 3: Value Properties of ComponentStream	32
Table 4: Reference Properties of ComponentStream	33
Table 5: Value Properties of Observation	36
Table 6: Value Properties of Sample	39
Table 7: Value Properties of Event	41
Table 8: Value Properties of Condition	43
Table 9: Value Properties of TimeSeries	49
Table 10: Value Properties of DataSet	52
Table 11: Part Properties of DataSet	52
Table 12: Value Properties of Table	54
Table 13: Part Properties of Table	55
Table 14: Value Properties of Entry	56
Table 15: Value Properties of TableEntry	56
Table 16: Part Properties of TableEntry	57
Table 17: Value Properties of Cell	58
Table 18: Value Properties of Alarm	62
Table 19: Value Properties of AssetChanged	65
Table 20: Value Properties of AssetRemoved	65

1 1 Purpose of This Document

2 This document, *MTConnect Standard: Part 3.0 - Observation Information Model* of the
3 MTConnect Standard, establishes the rules and terminology that describes the informa-
4 tion returned by an *MTConnect Agent* from a piece of equipment. The term (Observation
5 Information Model also defines, in *MTConnect Standard: Part 3.0 - Observation Infor-*
6 *mation Model*, the structure for the *response documents* that are returned from an *agent* in
7 response to a *sample request* or *current request*.

8 *MTConnect Standard: Part 3.0 - Observation Information Model* is not a stand-alone doc-
9 ument. This document is used in conjunction with *MTConnect Standard Part 1.0 - Fun-*
10 *damentals* which defines the fundamentals of the operation of the MTConnect Standard
11 and *MTConnect Standard: Part 2.0 - Device Information Model* that defines the semantic
12 model representing the information that may be returned from a piece of equipment.

13 Note: *MTConnect Standard: Part 5.0 - Interface Interaction Model* provides
14 details on extensions to the *Observation Information Model* required to de-
15 scribe the interactions between pieces of equipment.

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

19 2 Terminology and Conventions

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

22 2.1 MTConnect References

23 [MTConnect Part 1.0] *MTConnect Standard Part 1.0 - Fundamentals*. Version 2.0.

24 [MTConnect Part 2.0] *MTConnect Standard: Part 2.0 - Device Information Model*. Ver-
25 sion 2.0.

26 [MTConnect Part 3.0] *MTConnect Standard: Part 3.0 - Observation Information Model*.
27 Version 2.0.

28 [MTConnect Part 4.0] *MTConnect Standard: Part 4.0 - Asset Information Model*. Ver-
29 sion 2.0.

30 [MTConnect Part 5.0] *MTConnect Standard: Part 5.0 - Interface Interaction Model*. Ver-
31 sion 2.0.

32

33 3 Observation Information Model

34 The *Observation Information Model* provides a representation of the data reported by a
 35 piece of equipment used for a manufacturing process, or used for any other purpose. Ad-
 36 ditional descriptive information associated with the reported data is defined by the MT-
 37 ConnectDevices entity, which is described in *MTConnect Standard: Part 2.0 - Device*
 38 *Information Model*.

39 Information defined in the *Observation Information Model* allows a software application
 40 to (1) determine the *Observations* for *DataItems* returned from a piece of equipment and
 41 (2) interpret the data associated with those *Observations* with the same meaning, value,
 42 and context that it had at its original source. To do this, the software application issues one
 43 of two HTTP requests to an *agent* associated with a piece of equipment. They are:

- 44 • *sample*: Returns a designated number of time stamped *Observations* from an *agent*
 45 associated with a piece of equipment; subject to any HTTP filtering associated with
 46 the request. See *Agent* in *MTConnect Standard Part 1.0 - Fundamentals* for details
 47 on the *sample* HTTP request.
- 48 • *current*: Returns a snapshot of either the most recent values or the values at a
 49 given sequence number for all *Observations* associated with a piece of equipment
 50 from an *agent*; subject to any HTTP filtering associated with the request. See *Agent*
 51 in *MTConnect Standard Part 1.0 - Fundamentals* for details on the *current* HTTP
 52 request.

53 An *agent* responds to either the *sample* or *current* HTTP request with an *MTCon-*
 54 *nectStreams Response Document*. This document contains information describing *Ob-*
 55 *servations* reported by an *agent* associated with a piece of equipment. A client software
 56 application may correlate the information provided in the *MTConnectStreams Response*
 57 *Document* with the physical and logical structure for that piece of equipment defined in
 58 the *MTConnectDevices* entity to form a clear and unambiguous understanding of the
 59 information provided. (See details on the structure for a piece of equipment described in
 60 *MTConnect Standard: Part 2.0 - Device Information Model*).

61 *Streams* for an *MTConnectStreams* entity contains a *DeviceStream* entity for
 62 each piece of equipment represented in the document. Each *DeviceStream* is com-
 63 prised of two primary types of entities – *Components* and *Observations*. The contents
 64 of the *DeviceStream* container are described in detail in this document, *MTConnect*
 65 *Standard: Part 3.0 - Observation Information Model* of the *MTConnect Standard*.

66 *Components* are defined for both the *MTConnectDevices* and the *MTConnectStreams*

67 entities. These *Components* are used to provide a logical organization of the information
 68 provided in each entity.

69 • For an MTConnectDevices entity: *Components* organize information that rep-
 70 resents the physical and logical parts and sub-parts of a piece of equipment. (See
 71 Component in *MTConnect Standard: Part 2.0 - Device Information Model* for
 72 more details on *Components* used in the MTConnectDevices entity).

73 • For an MTConnectStreams entity: *Components* provide the structure to orga-
 74 nize the data returned from a piece of equipment and establishes the proper context
 75 for that data. The *Components* specifically defined for MTConnectStreams are
 76 DeviceStream (see *Section 3.1 - DeviceStream*) and ComponentStream (see
 77 *Section 3.2 - ComponentStream*).

78 DeviceStream and ComponentStream entities have a direct correlation to each of
 79 the *Component* defined in the MTConnectDevices entity.

80 Within each ComponentStream entity in the MTConnectStreams entity, *Observa-*
 81 *tions* are modeled as Observation entities. The three types of Observation entity
 82 are Sample, Event, and Condition. (See *Section 5 - Observation Types* for more
 83 information on these entities.)

84 3.1 DeviceStream

85 *organizes* data reported from a Device.

86 DeviceStream **MUST** be provided for each Device reporting data in an *MTConnect-*
 87 *Streams Response Document*.

88 If the response to the request for data from an *agent* does not contain any data for a specific
 89 Device, an empty DeviceStream entity **MAY** be created to indicate that the Device
 90 exists, but there was no data available.

91 3.1.1 Value Properties of DeviceStream

92 *Table 1* lists the Value Properties of DeviceStream.

Value Property name	Value Property type	Multiplicity
name	ID	1
uuid	ID	1

Table 1: Value Properties of DeviceStream

93 Descriptions for Value Properties of DeviceStream:

- 94 • name

95 name of the Device.

96 The value reported for name, DeviceStream **MUST** be the same as the value
 97 defined for the name, Device attribute of the same Device in the *MTConnect-*
 98 *Devices Response Document*.

- 99 • uuid

100 uuid of the Device.

101 The value reported for uuid, DeviceStream **MUST** be the same as the value
 102 defined for the uuid, Device attribute of the same Device in the *MTConnect-*
 103 *Devices Response Document*.

104 3.1.2 Part Properties of DeviceStream

105 Table 2 lists the Part Properties of DeviceStream.

Part Property name	Multiplicity
ComponentStream	1..*

Table 2: Part Properties of DeviceStream

106 Descriptions for Part Properties of DeviceStream:

- 107 • ComponentStream

108 organizes the data associated with each Component entity defined for a Device
 109 in the associated *MTConnectDevices Response Document*.

110 See Section 3.2 - ComponentStream.

111 3.2 ComponentStream

112 organizes the data associated with each Component entity defined for a Device in the
113 associated *MTConnectDevices Response Document*.

114 At least one of Sample, Event, or Condition **MUST** be organized by a Compo-
115 nentStream entity.

116 3.2.1 Value Properties of ComponentStream

117 Table 3 lists the Value Properties of ComponentStream.

Value Property name	Value Property type	Multiplicity
component	string	1
componentId	ID	1
name	string	0..1
nativeName	string	0..1
uuid	ID	0..1

Table 3: Value Properties of ComponentStream

118 Descriptions for Value Properties of ComponentStream:

- 119 • component
120 identifies the Component type associated with the ComponentStream.
121 Examples of component are Device, Controller, Linear and Loader.
- 122 • componentId
123 identifier of the Component as defined by the id attribute in the *MTConnectDe-*
124 *vices Response Document*.
- 125 • name
126 name of the Component associated with the ComponentStream.
- 127 • nativeName
128 common name of the Component associated with the ComponentStream.
- 129 • uuid
130 uuid of the Component associated with the ComponentStream.

131 3.2.2 Reference Properties of ComponentStream

132 *Table 4* lists the Reference Properties of ComponentStream.

Reference Property name	Multiplicity
Event (organized by Events)	0..*
Sample (organized by Samples)	0..*
Condition (organized by Conditions)	0..*

Table 4: Reference Properties of ComponentStream

133 Descriptions for Reference Properties of ComponentStream:

- 134 • Event
- 135 Observation that is a discrete piece of information from a piece of equipment.
- 136 Events groups one or more Event entities. See *Section 3.5 - Event*.
- 137 • Sample
- 138 Observation that is continuously changing or analog data value.
- 139 Samples groups one or more Sample entities. See *Section 3.4 - Sample*.
- 140 • Condition
- 141 Observation that provides the condition of a piece of equipment or a *Component*.
- 142 Conditions groups one or more Condition entities. See *Section 3.6 - Condition*.

145 Note: In the Extensible Markup Language (XML) representation, Con-
 146 ditions **MUST** appear as Condition element in the *MTConnect-*
 147 *Streams Response Document*.

148 3.3 Observation

149 abstract entity that provides telemetry data for a DataItem at a point in time.

150 Note: See *Section B.1 - Observations Schema Diagrams* for XML schema.
 151 The XML schema also shows differences in XML representation of certain
 152 MTConnect entities.

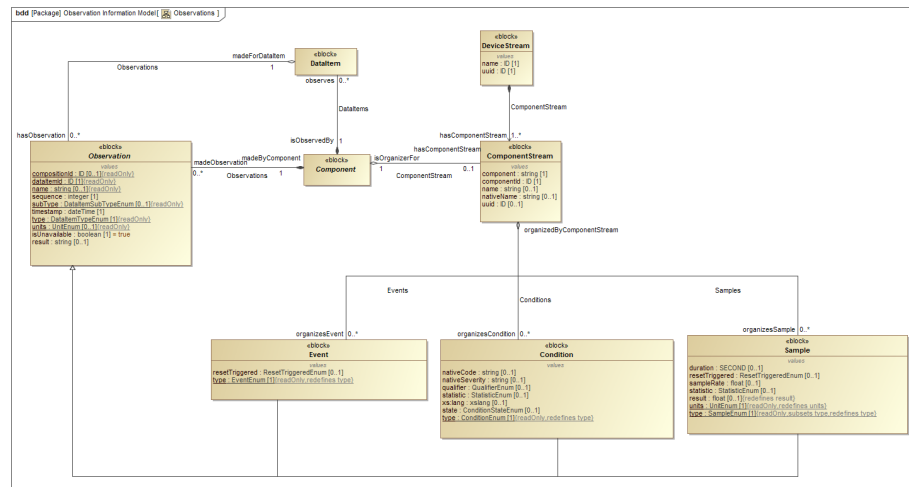


Figure 1: Observations

Figure 2 shows a complete example of DeviceStream for the Device shown in *Figure 2: Component Example in MTConnect Standard: Part 2.0 - Device Information Model*.

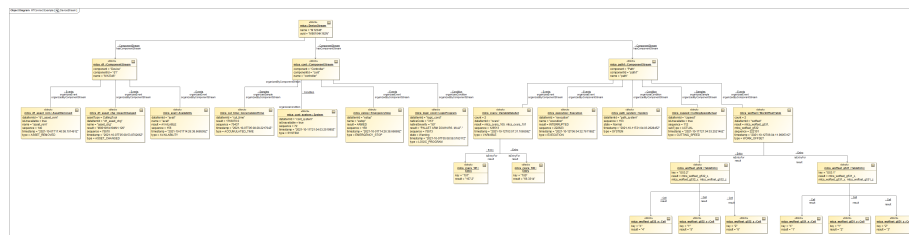


Figure 2: DeviceStream Example

155 Note: See *Example 1* for the XML representation of the same example.

156 This section provides semantic information for the `Observation` model.

157 Note: See *Section B.1 - Observations Schema Diagrams* for XML schema of
158 Observation types.

159 3.3.0.1 Observations made for DataItem

160 Component **observes** DataItem entities and creates Observation entities for the
161 DataItem entities. See Figure 1.

162 Observation entities made by a Component are organized by a ComponentStream
163 which is specifically created for that Component.

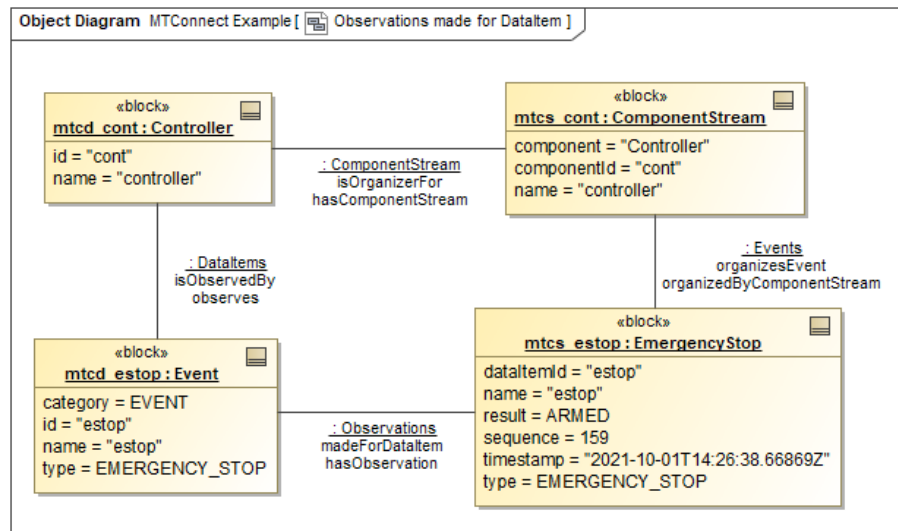


Figure 3: Observations made for DataItem Example

Note: See Section C.2 - *Observations made for DataItem Example* for how XML representation of the same example is separated into *MTConnectDevices Response Document* and *MTConnectStreams Response Document*.

Figure 3 is a subset of Figure 2. It shows an example of the association between a DataItemEvent type (EMERGENCY_STOP) and an Observation Event type (EmergencyStop). See Section 3.3.0.2 - *Naming Convention for Observation types*.

Figure 3 also shows example of the association between a Component type (Controller) and related ComponentStream.

3.3.0.2 Naming Convention for Observation types

The name of an Observation type **MUST** derive from the DataItem property type converted to Pascal-Case by removing underscores (_) and capitalizing each word. The conversion **MUST NOT** apply to the following abbreviated words: PH, AC, DC and URI. MTCONNECT **MUST** be converted to MTConnect. See Figure 3 for an example.

The name of an Observation type reported in the *MTConnectStreams Response Document* is extended when the representation property is used to further describe that DataItem in the *MTConnectDevices Response Document*. See Section 4 - *Representations* for more details.

181 3.3.1 Value Properties of Observation

182 *Table 5* lists the Value Properties of Observation.

Value Property name	Value Property type	Multiplicity
compositionId	ID	0..1
dataItemId	ID	1
name	string	0..1
sequence	integer	1
subType	DataItemSubTypeEnum	0..1
timestamp	dateTime	1
type	DataItemTypeEnum	1
units	UnitEnum	0..1

Table 5: Value Properties of Observation

183 Descriptions for Value Properties of Observation:

- 184 • compositionId
185 identifier of the `Composition` entity defined in the *MTConnectDevices Response*
186 *Document* associated with the data reported for the Observation.
- 187 • dataItemId
188 unique identifier of the `DataItem` associated with this Observation.
189 dataItemId **MUST** match the `id` attribute of the `DataItem` defined in the *MT-*
190 *ConnectDevices Response Document*.
- 191 • name
192 name of the `DataItem` associated with this Observation.
193 name **MUST** match the `name` attribute of the `DataItem` defined in the *MTCon-*
194 *nectDevices Response Document*.
- 195 • sequence
196 number representing the sequential position of an occurrence of an observation in
197 the data buffer of an *agent*.
198 sequence **MUST** have a value represented as an unsigned 64-bit value from 1 to
199 $2^{64} - 1$.
- 200 • subType
201 subtype of the `DataItem` associated with this Observation.

202 subType **MUST** match the subType attribute of the DataItem defined in the
 203 *MTConnectDevices Response Document*.

204 The value of subType **MUST** be one of the DataItemSubTypeEnum enumer-
 205 ation.

206 • timestamp

207 most accurate time available to a piece of equipment that represents the point in time
 208 that the data reported was measured.

209 • type

210 type of the DataItem associated with this Observation.

211 type **MUST** match the type attribute of the DataItem defined in the *MTCon-*
 212 *nectDevices Response Document*.

213 The value of type **MUST** be one of the DataItemTypeEnum enumeration.

214 • units

215 units of the DataItem associated with this Observation.

216 units **MUST** match the units attribute of the DataItem defined in the *MT-*
 217 *ConnectDevices Response Document*.

218 The value of units **MUST** be one of the UnitEnum enumeration.

219 • result

220 observation of the Observation entity.

221 The default value type for result is string.

222 3.4 Sample

223 Observation that is continuously changing or analog data value.

224 It provides the information and data reported from a piece of equipment for those DataItem
 225 entities defined with a category, DataItem property of SAMPLE in the *MTConnect-*
 226 *Devices Response Document*.

227 Sample **MUST** always be reported in float.

228 Figure 4 shows Sample type examples. It also shows an example for when the result
 229 is not available (dataItemId=cspeed).

230 Note: See *Example 4* for the XML representation of the same example.

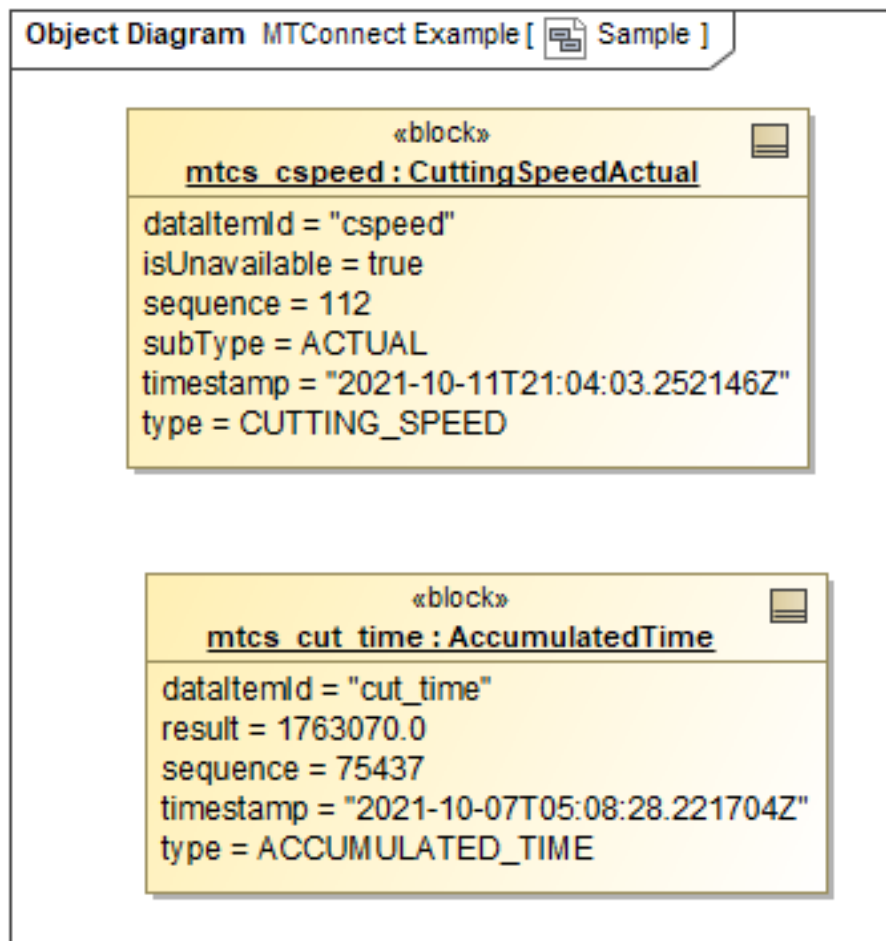


Figure 4: Sample Example

231 The following *Section 3.4.1 - Value Properties of Sample* lists the additional and/or up-
 232 dated attributes for Sample.

233 The value of Sample **MUST** be float.

234 The units for Sample **MUST** always be specified.

235 3.4.1 Value Properties of Sample

236 *Table 6* lists the Value Properties of Sample.

Value Property name	Value Property type	Multiplicity
duration	SECOND	0..1
resetTriggered	ResetTriggeredEnum	0..1
sampleRate	float	0..1
statistic	StatisticEnum	0..1

Table 6: Value Properties of Sample

237 Descriptions for Value Properties of Sample:

- 238 • duration
- 239 time-period over which the data was collected.
- 240 duration **MUST** be provided when the statistic attribute of the DataItem
- 241 is defined in the *MTConnectDevices Response Document*.
- 242 • resetTriggered
- 243 identifies when a reported value has been reset and what has caused that reset to
- 244 occur for those DataItem entities that may be periodically reset to an initial value.
- 245 resetTriggered **MUST** only be provided for the specific occurrence of a DataItem
- 246 reported in the *MTConnectStreams Response Document* when the reset occurred.
- 247 ResetTriggeredEnum Enumeration:
- 248 – ACTION_COMPLETE
- 249 result of the Observation that is measuring an action or operation was
- 250 reset upon completion of that action or operation.
- 251 – ANNUAL
- 252 result of the Observation was reset at the end of a 12-month period.
- 253 – DAY
- 254 result of the Observation was reset at the end of a 24-hour period.
- 255 – MAINTENANCE
- 256 result of the Observation was reset upon completion of a maintenance
- 257 event.
- 258 – MANUAL
- 259 result of the Observation was reset based on a physical reset action.
- 260 – MONTH
- 261 result of the Observation was reset at the end of a monthly period.

- 262 – POWER_ON
- 263 result of the Observation was reset when power was applied to the piece
- 264 of equipment after a planned or unplanned interruption of power has occurred.
- 265 – SHIFT
- 266 result of the Observation was reset at the end of a work shift.
- 267 – WEEK
- 268 result of the Observation was reset at the end of a 7-day period.
- 269 • sampleRate
- 270 rate at which successive samples of the value are recorded.
- 271 • statistic
- 272 type of statistical calculation defined by the statistic attribute of the DataItem
- 273 defined in the *MTConnectDevices Response Document*.
- 274 The value of statistic **MUST** be one of the StatisticEnum enumeration.

275 3.5 Event

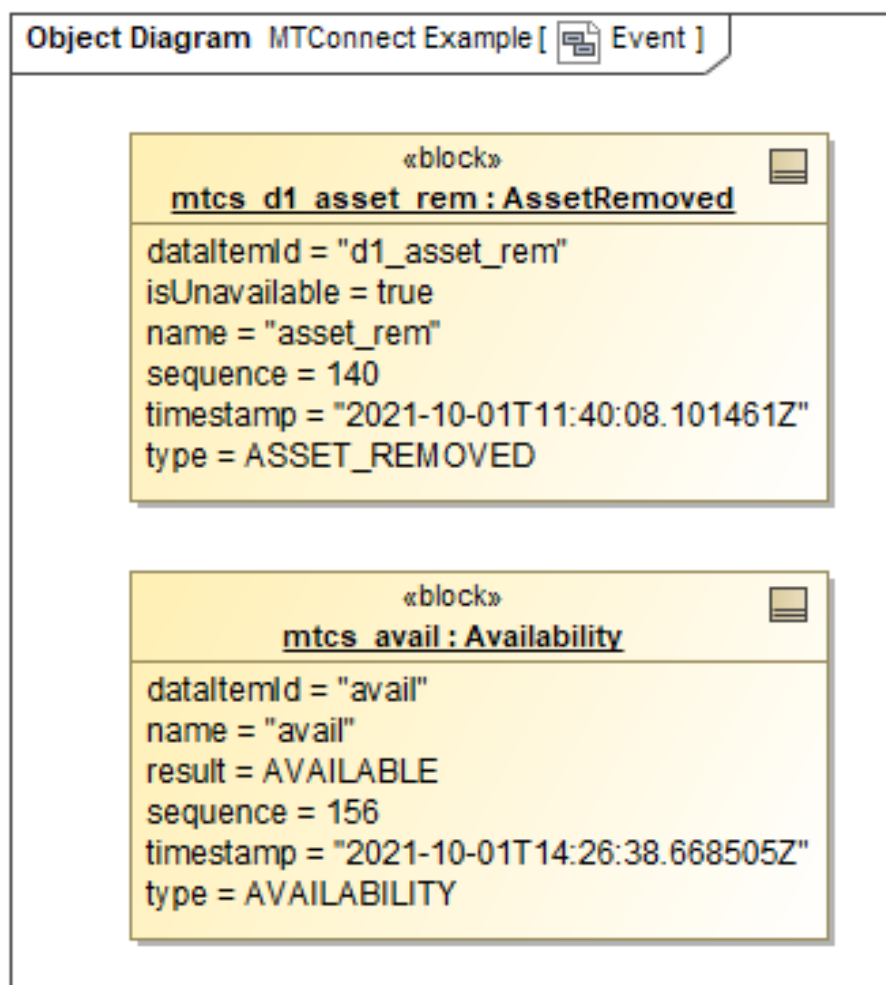
- 276 Observation that is a discrete piece of information from a piece of equipment.
- 277 It provides the information and data reported from a piece of equipment for those DataItem
- 278 entities defined with a category, DataItem property of EVENT in the *MTConnectDe-*
- 279 *vices Response Document*.
- 280 Figure 5 shows Event type examples. It also shows an example for when the result is
- 281 not available (dataItemId=dl_asset_rem).

282 Note: See *Example 5* for the XML representation of the same example.

- 283 The following *Section 3.5.1 - Value Properties of Event* lists the additional and/or updated
- 284 attributes for Event.

285 3.5.1 Value Properties of Event

286 *Table 7* lists the Value Properties of Event.

**Figure 5:** Event Example

Value Property name	Value Property type	Multiplicity
resetTriggered	ResetTriggeredEnum	0..1

Table 7: Value Properties of Event

287 Descriptions for Value Properties of Event:

- 288 • `resetTriggered`
- 289 identifies when a reported value has been reset and what has caused that reset to
- 290 occur for those `DataItem` entities that may be periodically reset to an initial value.
- 291 `resetTriggered` **MUST** only be provided for the specific occurrence of a `DataItem`
- 292 reported in the *MTConnectStreams Response Document* when the reset occurred.
- 293 The value of `resetTriggered` **MUST** be one of the `ResetTriggeredEnum`
- 294 enumeration.

295 3.6 Condition

296 Observation that provides the condition of a piece of equipment or a *Component*.

297 It provides the information and data reported from a piece of equipment for those `DataItem`
 298 entities defined with a `category`, `DataItem` property of `CONDITION` in the *MTCon-*
 299 *nectDevices Response Document*.

300 Figure 6 shows `Condition` type examples for various state: `Normal` (`dataItemId`
 301 `= path_system`) and `Warning` (`dataItemId = logic_cond`). It also shows an
 302 example for when the state is not available (`dataItemId = cont_system`).

303 Note: See *Example 6* for the XML representation of the same example.

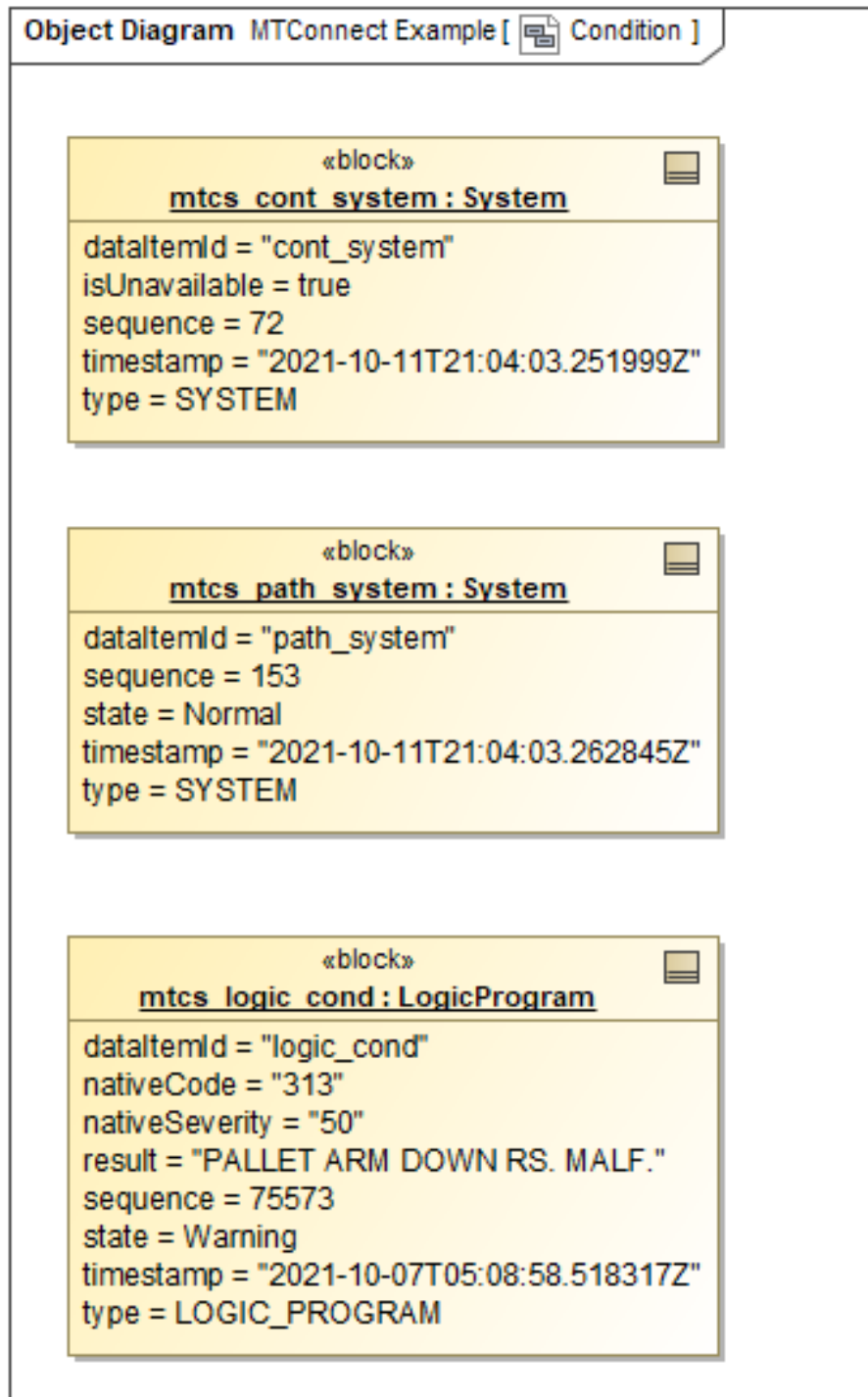
304 The following *Section 3.6.1 - Value Properties of Condition* lists the additional and/or
 305 updated attributes for `Condition`.

306 3.6.1 Value Properties of Condition

307 *Table 8* lists the Value Properties of `Condition`.

Value Property name	Value Property type	Multiplicity
nativeCode	string	0..1
nativeSeverity	string	0..1
qualifier	QualifierEnum	0..1
statistic	StatisticEnum	0..1
xs:lang	xslang	0..1
state	ConditionStateEnum	0..1

Table 8: Value Properties of Condition

**Figure 6:** Condition Example

308 Descriptions for Value Properties of Condition:

- 309 • `nativeCode`
310 native code (usually an alpha-numeric value) generated by the controller of a piece
311 of equipment providing a reference identifier for a `Condition`.
312 This is the same information an operator or maintenance personnel may see as a
313 reference code designating a specific fault code provided by the piece of equipment.
- 314 • `nativeSeverity`
315 severity information to a client software application if the piece of equipment designates a severity level to a fault.
- 317 • `qualifier`
318 additional information regarding a condition state associated with the measured
319 value of a process variable.
320 `qualifier` defines whether the condition state represented indicates a measured
321 value that is above or below an expected value of a process variable.
322 QualifierEnum Enumeration:
323 – HIGH
324 measured value is greater than the expected value for a process variable.
325 – LOW
326 measured value is less than the expected value for a process variable.
- 327 • `statistic`
328 `statistic` provides additional information describing the meaning of the Condition element.
329 `statistic` **MUST** match the `statistic` attribute of the `DataItem` entity defined in the *MTConnectDevices Response Document*.
330 The value of `statistic` **MUST** be one of the `StatisticEnum` enumeration.
- 333 • `xs:lang`
334 specifies the language of the `result` returned for the `Condition`.
335 See *Ref IETF RFC 4646* (<http://www.ietf.org/rfc/rfc4646.txt>).
- 336 • `state`
337 condition state of the piece of equipment or `Component`.
338 ConditionStateEnum Enumeration:

- 339 – Fault
- 340 condition state that requires intervention to continue operation to function prop-
- 341 erly.
- 342 – Normal
- 343 condition state that indicates operation within specified limits.
- 344 – Warning
- 345 condition state that requires concern and supervision and may become haz-
- 346 ardous if no action is taken.

347 4 Representations

348 This section provides semantic information for the Representation model.

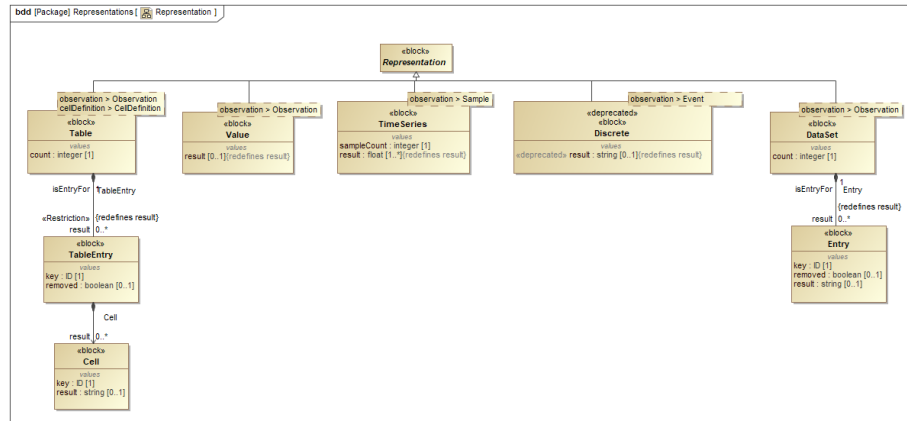


Figure 7: Representation

349 Note: See Section B.2 - Representation Schema Diagrams for XML schema.

350 4.1 Representation

351 specifies the format and structure of the result for an Observation.

352 The Representation type for an Observation is defined by the associated DataItem's
353 property representation in the *MTConnectDevices Response Document*.

354 Value is the default Representation type for all Observation types.

355 The name of the Observation type is modified for all Representation types other
356 than Value by appending the pascal case of the Representation type.

357 Example: The name for Sample Observation type Temperaturewith Repr-
358 sentation type of TimeSeries becomes TemperatureTimeSeries.

359 4.2 Value

360 default Representation type for all Observation types where result of the
361 Observation types is an MTConnect data type. See Section 6.1 - DataTypes.

362 4.3 TimeSeries

363 Representation for an Observation composed of a series of sampled data.

364 TimeSeries for an Observation is defined by the associated DataItem's property
365 representation as TIME_SERIES.

366 DataItem with TIME_SERIES representation **MUST** have a category of
367 SAMPLE.

368 Figure 8 shows the model for Temperature (Sample type) with a Representation
369 type of TimeSeries.

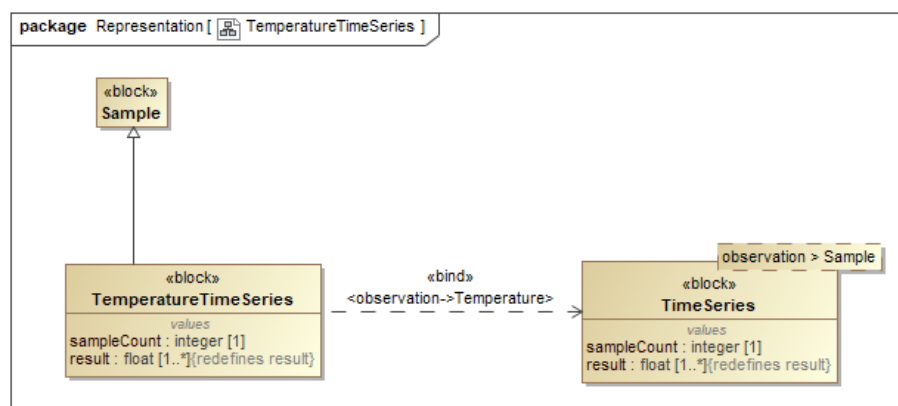


Figure 8: TemperatureTimeSeries

370 Note: See Figure 21 for XML schema.

371 TimeSeries **MUST** report multiple values at fixed intervals in a single Observation.
372 At minimum, one of DataItem or Observation **MUST** specify the sampleRate in
373 Hertz(values/second); fractional rates are permitted. When the Observation and the
374 DataItem specify the sampleRate, the Observation sampleRate supersedes
375 the DataItem.

376 The Observation **MUST** set the timestamp to the time the last value was observed.
377 The duration **MAY** indicate the time interval from the first to the last value in the series.

378 Section 4.3.1 - Value Properties of TimeSeries defines additional attributes for an Obser-
379 vation with TimeSeries Representation type.

380 4.3.1 Value Properties of TimeSeries

381 *Table 9* lists the Value Properties of TimeSeries.

Value Property name	Value Property type	Multiplicity
sampleCount	integer	1

Table 9: Value Properties of TimeSeries

382 Descriptions for Value Properties of TimeSeries:

- 383 • sampleCount
- 384 number of values given for the Observation.

385 4.4 <<deprecated>>Discrete

386 **DEPRECATED** Representation for an Observation where each discrete occur-
 387 rence of the data may have the same value as the previous occurrence of the data.

388 Discrete for an Observation is defined by the associated DataItem's property
 389 representation as DISCRETE.

390 DataItem with DISCRETE representation **MUST** have a category of EVENT.

391 *MTConnect Version 1.5* replaced representation DISCRETE with a discrete
 392 property for DataItem.

393 Each occurrence of the Observation **MAY** have the same value as the previous occur-
 394 rence, and **MUST NOT** suppress duplicates.

395 Examples of Discrete: A PartCount reporting the completion of each part using a 1
 396 to indicate completion of a single part, a Message that occurs each time a door opens.

397 4.5 DataSet

398 Representation for an Observation composed of value(s) represented as a set of
 399 *key-value pairs*.

400 DataSet for an Observation is defined by the associated DataItem's property
 401 representation as DATA_SET.

402 DataItem with DATA_SET representation **MUST** have a category of SAM-
 403 PLE or EVENT.

404 Figure 9 shows the model for Variable (Event type) with a Representation type
 405 of DataSet.

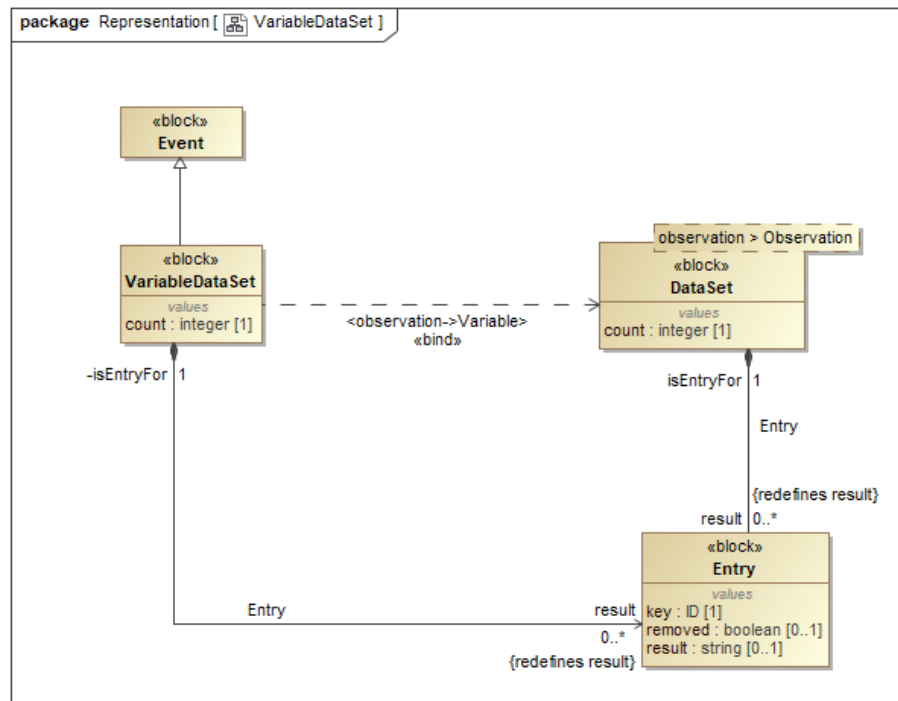


Figure 9: VariableDataSet

406 Note: See Figure 22 for XML schema.

407 DataSet reports multiple values as a set of *key-value pair* where each *key* **MUST** be
 408 unique. The representation of the *key-value pair* is an Entry. The value of each En-
 409 try **MUST** have the same constraints and format as the Observation defined for the
 410 VALUE representation for the DataItem type (See Value).

411 The meaning of each Entry **MAY** be provided as the DataItem EntryDefinition.

412 Figure 10 shows Event Observation type Variable with a Representation
 413 type of DataSet.

414 Note: See Example 7 for the XML representation of the same example.

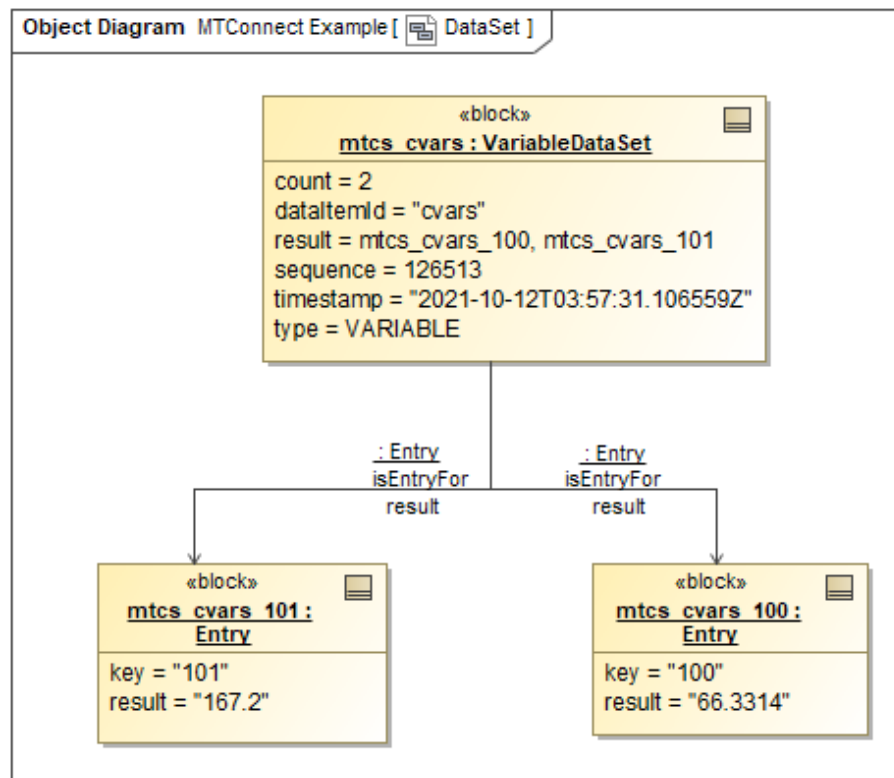


Figure 10: DataSet Example

4.5.0.1 Management of Data Set Observations

An *agent* **MUST** maintain the current state of the DataSet as described in *MTConnect Standard Part 1.0 - Fundamentals*.

One or more *key-value pairs* **MAY** be added, removed, or changed in an Observation. An *agent* **MUST** publish the changes to one or more *key-value pairs* as a single Observation. An *agent* **MUST** indicate the removal of a *key-value pair* from a DataSet using the `removed` attribute equal `true`.

When the `DataItem discrete, DataItem` attribute is `false` or is not present, an *agent* in response to a *sample request* **MUST** only publish the changed *key-value pair* since the previous state of the DataSet.

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

When an *agent* responds to a *current request*, the *response document* **MUST** include the full set of *key-value pairs*. If the *current request* includes an `at query` parameter, the *agent*

429 **MUST** provide the set of *key-value pairs* at the *sequence number*.

430 When an Observation *reset* occurs, the DataSet **MUST** remove all *key-value pairs*
 431 making the set empty. The Observation **MAY** simultaneously populate the DataSet
 432 with new *key-value pairs*. The previous entries **MUST NOT** be included and **MUST NOT**
 433 have removed attribute equal `true`.

434 When the Observation is UNAVAILABLE the DataSet **MUST** remove all *key-value*
 435 *pairs* making the set empty.

436 4.5.1 Value Properties of DataSet

437 Table 10 lists the Value Properties of DataSet.

Value Property name	Value Property type	Multiplicity
count	integer	1

Table 10: Value Properties of DataSet

438 Descriptions for Value Properties of DataSet:

- 439 • count
- 440 number of Entry elements for the Observation.

441 4.5.2 Part Properties of DataSet

442 Table 11 lists the Part Properties of DataSet.

Part Property name	Multiplicity
Entry	0..*

Table 11: Part Properties of DataSet

443 Descriptions for Part Properties of DataSet:

- 444 • Entry
- 445 *key-value pair* published as part of a DataSet.
- 446 See Section 4.7 - Entry.

447 4.6 Table

448 Representation for an Observation composed of two-dimensional sets of *key-*
 449 *value pairs* where the Entry represents rows containing sets of *key-value pairs* given by
 450 Cell entities.

451 Table for an Observation is defined by the associated DataItem's property rep-
 452 resentation as TABLE.

453 DataItem with TABLE representation **MUST** have a category of SAMPLE or
 454 EVENT.

455 Figure 11 shows the model for WorkOffset (Event type) with a Representation
 456 type of Table.

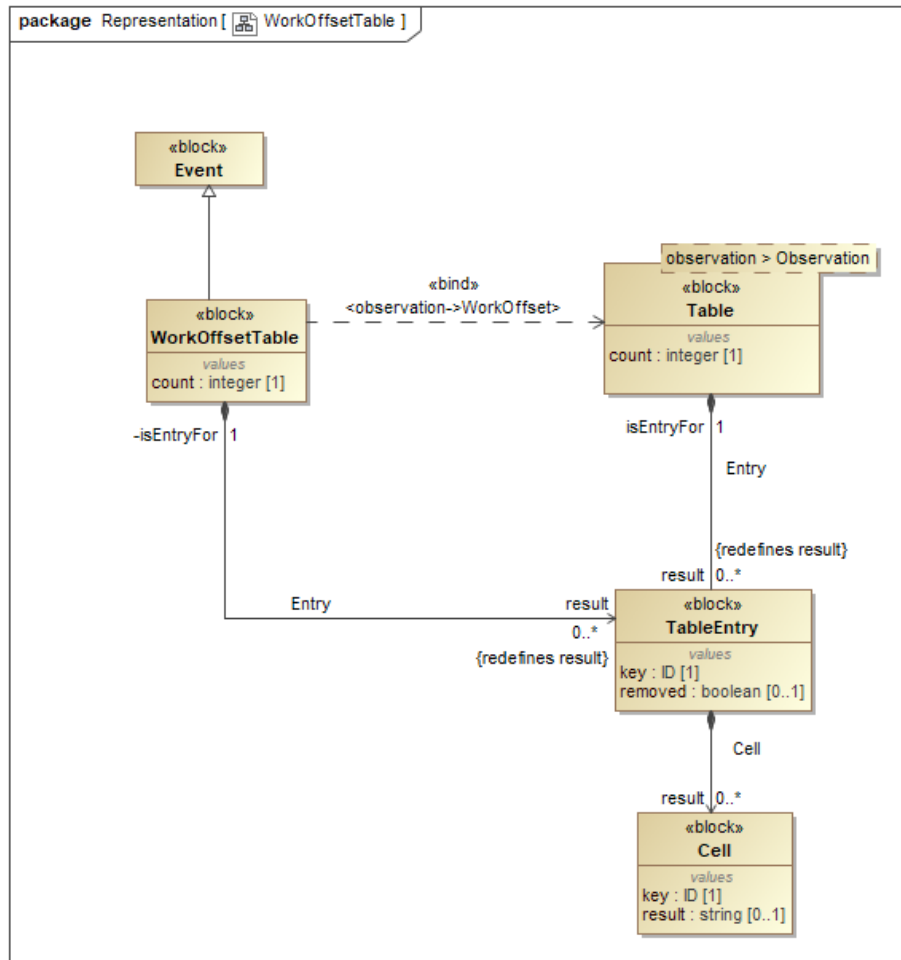


Figure 11: WorkOffsetTable

457 Note: See Figure 24 for XML schema.

458 Table has the same behavior as the DataSet for change tracking, clearing, and history.
 459 When an Entry changes, all Cell entities update at the same time; they are not tracked
 460 separately like Entry.

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

463 The Entry key attribute **MUST** be the unique identity of the Entry within an Obser-
 464 vation. The Cell key attribute **MUST** be the unique identity of the Cell within an
 465 Entry.

466 Figure 12 shows Event Observation type WorkOffset with a Representation
 467 type of Table.

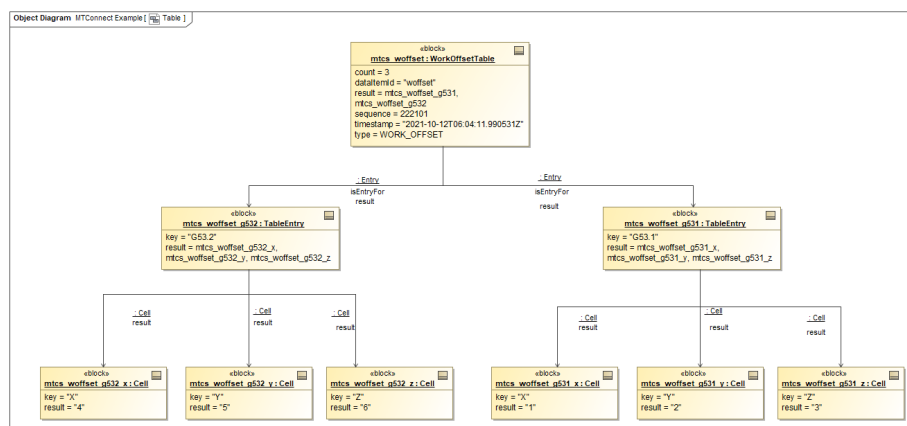


Figure 12: Table Example

468 Note: See Example 8 for the XML representation of the same example.

469 4.6.1 Value Properties of Table

470 Table 12 lists the Value Properties of Table.

Value Property name	Value Property type	Multiplicity
count	integer	1

Table 12: Value Properties of Table

471 Descriptions for Value Properties of Table:

- 472 • count
- 473 number of *key-value pairs* represented as `Entry` entities.

474 4.6.2 Part Properties of Table

475 *Table 13* lists the Part Properties of `Table`.

Part Property name	Multiplicity
<code>TableEntry</code>	0..*

Table 13: Part Properties of `Table`

476 Descriptions for Part Properties of `Table`:

- 477 • `TableEntry`
- 478 *key-value pair* published as part of a `Table`.

479 Note: In the XML representation, `TableEntry` **MUST** appear as `En-`
 480 `try`.

481 See *Section 4.8 - TableEntry*.

482 4.7 Entry

483 *key-value pair* published as part of a `DataSet`.

484 4.7.0.1 Constraints for Entry Values

485 The value of each `Entry` **MUST** have the same restrictions as the value of an observation
 486 with representation of `VALUE`.

487 An `Entry` **MAY** be further constrained by the `DataItem` definition (see *MTConnect*
 488 *Standard: Part 2.0 - Device Information Model*), for example a `VariableDataSet`
 489 having a string value **MAY** have a floating-point `Temperature` value. A restriction
 490 **MUST NOT** be broadened or removed, for example, the value “READY” **MUST NOT**
 491 occur with a `TemperatureDataSet` constrained to floating-point numbers.

492 The *MTConnect Standard: Part 2.0 - Device Information Model* `DataItem` Defini-
 493 tion **MAY** provide the type and units of an `Entry` for a key.

494 4.7.1 Value Properties of Entry

495 *Table 14* lists the Value Properties of Entry.

Value Property name	Value Property type	Multiplicity
key	ID	1
removed	boolean	0..1

Table 14: Value Properties of Entry

496 Descriptions for Value Properties of Entry:

- 497 • key
- 498 unique identifier for each *key-value pair*.
- 499 • removed
- 500 removal indicator of a *key-value pair*.

501 4.8 TableEntry

502 *key-value pair* published as part of a Table.

503 Note: In the XML representation, TableEntry **MUST** appear as Entry.

504 4.8.1 Value Properties of TableEntry

505 *Table 15* lists the Value Properties of TableEntry.

Value Property name	Value Property type	Multiplicity
key	ID	1
removed	boolean	0..1

Table 15: Value Properties of TableEntry

506 Descriptions for Value Properties of TableEntry:

- 507 • key
- 508 unique identifier for each *key-value pair*.

- 509 • removed
- 510 removal indicator of a *key-value pair*.

511 4.8.2 Part Properties of TableEntry

512 *Table 16* lists the Part Properties of TableEntry.

Part Property name	Multiplicity
Cell	0..*

Table 16: Part Properties of TableEntry

513 Descriptions for Part Properties of TableEntry:

- 514 • Cell
- 515 *key-value pair* published as part of a TableEntry.
- 516 See *Section 4.9 - Cell*.

517 4.9 Cell

518 *key-value pair* published as part of a TableEntry.

519 4.9.0.1 Constraints for Cell Values

520 The value of each Cell **MUST** have the same restrictions as the value of an observation
521 with representation of VALUE.

522 An Cell **MAY** be further constrained by the DataItem definition (see *MTConnect Stan-*
523 *dard: Part 2.0 - Device Information Model*), for example a VariableDataSet having
524 a string value **MAY** have a floating-point Temperature value. A restriction **MUST**
525 **NOT** be broadened or removed, for example, the value READY **MUST NOT** occur with a
526 TemperatureDataSet constrained limited to floating-point numbers.

527 The *MTConnect Standard: Part 2.0 - Device Information Model* DataItem Defini-
528 tion **MAY** provide the type and units of a Cell for a key.

529 4.9.1 Value Properties of Cell

530 *Table 17* lists the Value Properties of `Cell`.

Value Property name	Value Property type	Multiplicity
key	ID	1

Table 17: Value Properties of Cell

531 Descriptions for Value Properties of `Cell`:

- 532 • key
- 533 unique identifier for each *key-value pair*.

534 **5 Observation Types**

535 This section provides semantic information for the `Observation` types.

536 `Observation` entities are instantiated as `Sample`, `Event` or `Condition` entities
537 based upon the `category` and `type` attributes defined for the corresponding `DataItem`.

538 See Figure 2 for examples on how the `Observation` types are organized within `Com-`
539 `ponentStream`.

540 **5.1 Condition Types**

541 This section provides semantic information for `Condition` types.

542 `Condition` types are reported differently from other `Observation` types. They are
543 reported based on the condition state for each `Condition`.

544 The `type` and `subType` (where applicable) properties for a `Condition` **MAY** be any
545 of the `type` and `subType` attributes defined for `SAMPLE` category or `EVENT` cat-
546 egory `DataItem` listed in the *Device Information Model*.

547 This section lists additional `Condition` types that have been defined to represent the
548 health and fault status of *Components*.

549 **5.1.1 Actuator**

550 indication of a fault associated with an actuator.

551 **5.1.2 Communications**

552 indication that the piece of equipment has experienced a communications failure.

553 **5.1.3 DataRange**

554 indication that the value of the data associated with a measured value or a calculation is
555 outside of an expected range.

556 **5.1.4 LogicProgram**

557 indication that an error occurred in the logic program or programmable logic controller
558 (PLC) associated with a piece of equipment.

559 **5.1.5 MotionProgram**

560 indication that an error occurred in the motion program associated with a piece of equip-
561 ment.

562 **5.1.6 System**

563 general purpose indication associated with an electronic component of a piece of equip-
564 ment or a controller that represents a fault that is not associated with the operator, program,
565 or hardware.

566 **5.2 Event Types**

567 This section provides semantic information for `Event` types.

568 **5.2.1 ActivationCount**

569 accumulation of the number of times a function has attempted to, or is planned to attempt
570 to, activate or be performed.

571 The default `subType` of `ActivationCount` is `ALL`.

572 The value of `ActivationCount` **MUST** be integer.

573 **5.2.1.1 Subtypes of ActivationCount**

574 • `ABORTED`

575 accumulation of actions or activities that were attempted, but terminated before they
576 could be completed.

- 577 • ALL
- 578 accumulation of all actions, items, or activities being counted independent of the
- 579 outcome.
- 580 • BAD
- 581 accumulation of actions, items, or activities being counted that do not conform to
- 582 specification or expectation.
- 583 • COMPLETE
- 584 accumulation of actions, items, or activities that have been completed, independent
- 585 of the outcome.
- 586 • FAILED
- 587 accumulation of actions or activities that were attempted, but failed to complete or
- 588 resulted in an unexpected or unacceptable outcome.
- 589 • GOOD
- 590 accumulation of actions, items, or activities being counted that conform to specifi-
- 591 cation or expectation.
- 592 • REMAINING
- 593 accumulation of actions, items, or activities yet to be counted.
- 594 • TARGET
- 595 goal of the operation or process.

596 5.2.2 ActiveAxes

- 597 set of axes currently associated with a `Path` or `Controller`.
- 598 The value of `ActiveAxes` **MUST** be a list of string of size 0 . . *.

599 5.2.3 ActuatorState

- 600 operational state of an apparatus for moving or controlling a mechanism or system.
- 601 `ActuatorStateEnum` Enumeration:

- 602 • ACTIVE
- 603 Actuator is operating.
- 604 • INACTIVE
- 605 Actuator is not operating.

606 **5.2.4 AdapterSoftwareVersion**

607 originator's software version of the *adapter*.

608 **5.2.5 AdapterURI**

609 Uniform Resource Identifier (URI) of the *adapter*.

610 **5.2.6 <<deprecated>>Alarm**

611 **DEPRECATED:** Replaced with CONDITION category data items in Version 1.1.0.

612 **5.2.6.1 Value Properties of Alarm**

613 *Table 18* lists the Value Properties of Alarm.

Value Property name	Value Property type	Multiplicity
<<deprecated>> code	AlarmCodeEnum	1
<<deprecated>> severity	AlarmSeverityEnum	1
<<deprecated>> nativeCode	string	1
<<deprecated>> state	AlarmStateEnum	1
<<deprecated>> lang	xslang	0..1

Table 18: Value Properties of Alarm

614 Descriptions for Value Properties of Alarm:

- 615 • <<deprecated>> code
- 616 type of alarm.

- 617 • <<deprecated>> severity
- 618 severity of the alarm.
- 619 • <<deprecated>> nativeCode
- 620 native code for the piece of equipment.
- 621 • <<deprecated>> state
- 622 state of the alarm.
- 623 • <<deprecated>> lang
- 624 specifies the language of the alarm text.
- 625 See *Ref IETF RFC 4646* (<http://www.ietf.org/rfc/rfc4646.txt>).

626 5.2.7 AlarmLimit

627 set of limits used to trigger warning or alarm indicators.

628 The Entry key **MUST** be one or more from the AlarmLimitResult keys.

629 AlarmLimitResult keys:

- 630 • UPPER_LIMIT
- 631 upper conformance boundary for a variable.
- 632 Note: immediate concern or action may be required.
- 633 The value of UPPER_LIMIT **MUST** be float.
- 634 • UPPER_WARNING
- 635 upper boundary indicating increased concern and supervision may be required.
- 636 The value of UPPER_WARNING **MUST** be float.
- 637 • LOWER_WARNING
- 638 lower boundary indicating increased concern and supervision may be required.
- 639 The value of LOWER_WARNING **MUST** be float.
- 640 • LOWER_LIMIT
- 641 lower conformance boundary for a variable.
- 642 Note: immediate concern or action may be required.
- 643 The value of LOWER_LIMIT **MUST** be float.

644 5.2.8 Application

645 application on a Component.

646 A subType **MUST** always be specified.

647 5.2.8.1 Subtypes of Application

648 • INSTALL_DATE

649 date the hardware or software was installed.

650 The value of Application **MUST** be dateTime. See *Section 6.1.5 - dateTime*.

651 dateTime Enumeration:

652 • LICENSE

653 license code to validate or activate the hardware or software.

654 • MANUFACTURER

655 corporate identity for the maker of the hardware or software.

656 • RELEASE_DATE

657 date the hardware or software was released for general use.

658 The value of Application **MUST** be dateTime. See *Section 6.1.5 - dateTime*.

659 dateTime Enumeration:

660 • VERSION

661 version of the hardware or software.

662 5.2.9 AssetChanged

663 assetId of the Asset that has been added or changed.

664 5.2.9.1 Value Properties of AssetChanged

665 Table 19 lists the Value Properties of AssetChanged.

Value Property name	Value Property type	Multiplicity
assetType	string	1

Table 19: Value Properties of AssetChanged

666 Descriptions for Value Properties of AssetChanged:

- 667 • assetType
- 668 type of Asset changed. See *MTConnect Standard: Part 4.0 - Asset Information*
- 669 *Model* for details on the Asset model.

670 5.2.10 AssetCount

671 *data set* of the number of *Assets* of a given type for a *Device*.

672 The value of AssetCount **MUST** be integer.

673 5.2.11 AssetRemoved

674 assetId of the *Asset* that has been removed.

675 5.2.11.1 Value Properties of AssetRemoved

676 *Table 20* lists the Value Properties of AssetRemoved.

Value Property name	Value Property type	Multiplicity
assetType	string	1

Table 20: Value Properties of AssetRemoved

677 Descriptions for Value Properties of AssetRemoved:

- 678 • assetType
- 679 type of Asset removed. See *MTConnect Standard: Part 4.0 - Asset Information*
- 680 *Model* for details on the Asset model.

681 5.2.12 Availability

682 *agent's* ability to communicate with the data source.

683 AvailabilityEnum Enumeration:

- 684 • AVAILABLE
- 685 data source is active and capable of providing data.
- 686 • UNAVAILABLE
- 687 data source is either inactive or not capable of providing data.

688 5.2.13 AxisCoupling

689 describes the way the axes will be associated to each other.

690 This is used in conjunction with COUPLED_AXES to indicate the way they are interacting.

691 AxisCouplingEnum Enumeration:

- 692 • MASTER
- 693 axis is the master of the CoupledAxes.
- 694 • SLAVE
- 695 axis is a slave to the CoupledAxes.
- 696 • SYNCHRONOUS
- 697 axes are not physically connected to each other but are operating together in lock-
- 698 step.
- 699 • TANDEM
- 700 axes are physically connected to each other and operate as a single unit.

701 5.2.14 AxisFeedrateOverride

702 value of a signal or calculation issued to adjust the feedrate of an individual linear type
703 axis.

704 The value of AxisFeedrateOverride **MUST** be float.

705 **5.2.14.1 Subtypes of AxisFeedrateOverride**

- 706 • JOG
- 707 relating to momentary activation of a function or a movement.
- 708 **DEPRECATION WARNING:** May be deprecated in the future.
- 709 When the JOG subtype of AxisFeedrateOverride is applied, the resulting
- 710 commanded feedrate for the axis is limited to the value of the original JOG subtype
- 711 of the AxisFeedrate multiplied by the value of the JOG subtype of AxisFeedrateOverride.
- 712
- 713 • PROGRAMMED
- 714 directive value without offsets and adjustments.
- 715 • RAPID
- 716 performing an operation faster or in less time than nominal rate.

717 **5.2.15 AxisInterlock**

718 state of the axis lockout function when power has been removed and the axis is allowed to
 719 move freely.

720 AxisInterlockEnum Enumeration:

- 721 • ACTIVE
- 722 axis lockout function is activated, power has been removed from the axis, and the
- 723 axis is allowed to move freely.
- 724 • INACTIVE
- 725 axis lockout function has not been activated, the axis may be powered, and the axis
- 726 is capable of being controlled by another component.

727 **5.2.16 AxisState**

728 state of a Linear or Rotary component representing an axis.

729 AxisStateEnum Enumeration:

- 730 • HOME
- 731 axis is in its home position.
- 732 • PARKED
- 733 axis has been moved to a fixed position and is being maintained in that position
- 734 either electrically or mechanically.
- 735 Action is required to release the axis from this position.
- 736 • STOPPED
- 737 axis is stopped.
- 738 • TRAVEL
- 739 axis is in motion.

740 **5.2.17 Block**

741 line of code or command being executed by a `Controller` entity.

742 **5.2.18 BlockCount**

743 total count of the number of blocks of program code that have been executed since execu-
744 tion started.

745 The value of `BlockCount` **MUST** be integer.

746 **5.2.19 ChuckInterlock**

747 state of an interlock function or control logic state intended to prevent the associated
748 `Chuck` component from being operated.

749 `ChuckInterlockEnum` Enumeration:

- 750 • ACTIVE
- 751 chuck cannot be unclamped.
- 752 • INACTIVE
- 753 chuck can be unclamped.

754 **5.2.19.1 Subtypes of ChuckInterlock**

- 755 • MANUAL_UNCLAMP
- 756 indication of the state of an operator controlled interlock that can inhibit the ability
- 757 to initiate an unclamp action of an electronically controlled chuck.
- 758 When `ChuckInterlockManualUnclamp` is `ACTIVE`, it is expected that a chuck
- 759 cannot be unclamped until `ChuckInterlockManualUnclamp` is set to `INAC-`
- 760 `TIVE`.

761 **5.2.20 ChuckState**

762 operating state of a mechanism that holds a part or stock material during a manufacturing
763 process.

764 It may also represent a mechanism that holds any other mechanism in place within a piece
765 of equipment.

766 `ChuckStateEnum` Enumeration:

- 767 • CLOSED
- 768 Chuck is closed to the point of a positive confirmation.
- 769 • OPEN
- 770 Chuck is open to the point of a positive confirmation.
- 771 • UNLATCHED
- 772 Chuck is not closed to the point of a positive confirmation and not open to the point
- 773 of a positive confirmation.
- 774 It is in an intermediate position.

775 **5.2.21 ClockTime**

776 time provided by a timing device at a specific point in time.

777 The value of `ClockTime` **MUST** be `dateTime`. See *Section 6.1.5 - dateTime*.

778 5.2.22 <<deprecated>>Code

779 programmatic code being executed.

780 **DEPRECATED** in *Version 1.1*.

781 5.2.23 CompositionState

782 operating state of a mechanism represented by a `Composition` entity.

783 A `subType` **MUST** always be specified.

784 5.2.23.1 Subtypes of CompositionState

785 • ACTION

786 indication of the operating state of a mechanism.

787 `CompositionStateActionEnum` Enumeration:

788 – ACTIVE

789 `Composition` is operating.

790 – INACTIVE

791 `Composition` is not operating.

792 • LATERAL

793 indication of the position of a mechanism that may move in a lateral direction.

794 `CompositionStateLateralEnum` Enumeration:

795 – LEFT

796 position of the `Composition` is oriented to the left to the point of a positive
797 confirmation.

798 – RIGHT

799 position of the `Composition` is oriented to the right to the point of a positive
800 confirmation.

801 – TRANSITIONING

802 position of the `Composition` is not oriented to the right to the point of a
803 positive confirmation and is not oriented to the left to the point of a positive
804 confirmation.

805 It is in an intermediate position.

- 806 • MOTION
- 807 indication of the open or closed state of a mechanism.
- 808 CompositionStateMotionEnum Enumeration:
- 809 – CLOSED
- 810 position of the Composition is closed to the point of a positive confirmation.
- 811 – OPEN
- 812 position of the Composition is open to the point of a positive confirmation.
- 813 – UNLATCHED
- 814 position of the Composition is not open to the point of a positive confirma-
- 815 tion and is not closed to the point of a positive confirmation.
- 816 It is in an intermediate position.
- 817 • SWITCHED
- 818 indication of the activation state of a mechanism.
- 819 CompositionStateSwitchedEnum Enumeration:
- 820 – OFF
- 821 activation state of the Composition is in an OFF condition, it is not operat-
- 822 ing, or it is not powered.
- 823 – ON
- 824 activation state of the Composition is in an ON condition, it is operating, or
- 825 it is powered.
- 826 • VERTICAL
- 827 indication of the position of a mechanism that may move in a vertical direction.
- 828 CompositionStateVerticalEnum Enumeration:
- 829 – DOWN
- 830 position of the Composition element is oriented in a downward direction to
- 831 the point of a positive confirmation.
- 832 – TRANSITIONING
- 833 position of the Composition element is not oriented in an upward direc-
- 834 tion to the point of a positive confirmation and is not oriented in a downward
- 835 direction to the point of a positive confirmation.
- 836 It is in an intermediate position.
- 837 – UP
- 838 position of the Composition element is oriented in an upward direction to
- 839 the point of a positive confirmation.

840 5.2.24 ConnectionStatus

841 status of the connection between an *adapter* and an *agent*.

842 ConnectionStatusEnum Enumeration:

- 843 • CLOSED
- 844 no connection at all.
- 845 • ESTABLISHED
- 846 open connection.
- 847 The normal state for the data transfer phase of the connection.
- 848 • LISTEN
- 849 *agent* is waiting for a connection request from an *adapter*.

850 5.2.25 ControlLimit

851 set of limits used to indicate whether a process variable is stable and in control.

852 The Entry key **MUST** be one or more from the ControlLimitResult keys.

853 ControlLimitResult keys:

- 854 • UPPER_LIMIT
- 855 upper conformance boundary for a variable.
- 856 Note: immediate concern or action may be required.
- 857 The value of UPPER_LIMIT **MUST** be float.
- 858 • UPPER_WARNING
- 859 upper boundary indicating increased concern and supervision may be required.
- 860 The value of UPPER_WARNING **MUST** be float.
- 861 • NOMINAL
- 862 ideal or desired value for a variable.
- 863 The value of NOMINAL **MUST** be float.

- 864 • LOWER_WARNING
- 865 lower boundary indicating increased concern and supervision may be required.
- 866 The value of LOWER_WARNING **MUST** be float.
- 867 • LOWER_LIMIT
- 868 lower conformance boundary for a variable.
- 869 Note: immediate concern or action may be required.
- 870 The value of LOWER_LIMIT **MUST** be float.

871 5.2.26 ControllerMode

872 current mode of the Controller component.

873 ControllerModeEnum Enumeration:

- 874 • AUTOMATIC
- 875 Controller is configured to automatically execute a program.
- 876 • EDIT
- 877 Controller is currently functioning as a programming device and is not capable
- 878 of executing an active program.
- 879 • <<deprecated>> FEED_HOLD
- 880 axes of the device are commanded to stop, but the spindle continues to function.
- 881 • MANUAL
- 882 Controller is not executing an active program.
- 883 It is capable of receiving instructions from an external source – typically an operator.
- 884 The Controller executes operations based on the instructions received from the
- 885 external source.
- 886 • MANUAL_DATA_INPUT
- 887 operator can enter a series of operations for the Controller to perform.
- 888 The Controller will execute this specific series of operations and then stop.
- 889 • SEMI_AUTOMATIC
- 890 Controller is operating in a mode that restricts the active program from process-
- 891 ing its next process step without operator intervention.

892 5.2.27 ControllerModeOverride

893 setting or operator selection that changes the behavior of a piece of equipment.

894 ControllerModeOverrideEnum Enumeration:

895 • OFF

896 ControllerModeOverride is in the OFF state and the mode override is inactive.
897

898 • ON

899 ControllerModeOverride is in the ON state and the mode override is active.

900 A subType **MUST** always be specified.

901 5.2.27.1 Subtypes of ControllerModeOverride

902 • DRY_RUN

903 setting or operator selection used to execute a test mode to confirm the execution of
904 machine functions.

905 When DRY_RUN is ON, the equipment performs all of its normal functions, except
906 no part or product is produced. If the equipment has a spindle, spindle operation is
907 suspended.

908 • MACHINE_AXIS_LOCK

909 setting or operator selection that changes the behavior of the controller on a piece of
910 equipment.

911 When MACHINE_AXIS_LOCK is ON, program execution continues normally, but
912 no equipment motion occurs.

913 • OPTIONAL_STOP

914 setting or operator selection that changes the behavior of the controller on a piece of
915 equipment.

916 The program execution is stopped after a specific program block is executed when
917 OPTIONAL_STOP is ON.

918 In the case of a G-Code program, a program block containing a M01 code designates
919 the command for an OPTIONAL_STOP.

920 Execution **MUST** change to `OPTIONAL_STOP` after a program block speci-
 921 fying an optional stop is executed and the `ControllerModeOverride OP-`
 922 `TIONAL_STOP` selection is ON.

- 923 • `SINGLE_BLOCK`
 924 setting or operator selection that changes the behavior of the controller on a piece of
 925 equipment.
 926 Program execution is paused after each block of code is executed when `SINGLE_BLOCK`
 927 is ON.
 928 When `SINGLE_BLOCK` is ON, Execution **MUST** change to `INTERRUPTED`
 929 after completion of each block of code.
- 930 • `TOOL_CHANGE_STOP`
 931 setting or operator selection that changes the behavior of the controller on a piece of
 932 equipment.
 933 Program execution is paused when a command is executed requesting a cutting tool
 934 to be changed.
 935 Execution **MUST** change to `INTERRUPTED` after completion of the command
 936 requesting a cutting tool to be changed and `TOOL_CHANGE_STOP` is ON.

937 5.2.28 CoupledAxes

938 set of associated axes.
 939 The value of `CoupledAxes` **MUST** be a list of `string` of size `0..*`.

940 5.2.29 CycleCount

941 accumulation of the number of times a cyclic function has attempted to, or is planned to
 942 attempt to execute.
 943 The default `subType` of `CycleCount` is `ALL`.
 944 The value of `CycleCount` **MUST** be `integer`.

945 5.2.29.1 Subtypes of CycleCount

- 946 • ABORTED
- 947 accumulation of actions or activities that were attempted, but terminated before they
- 948 could be completed.

- 949 • ALL
- 950 accumulation of all actions, items, or activities being counted independent of the
- 951 outcome.

- 952 • BAD
- 953 accumulation of actions, items, or activities being counted that do not conform to
- 954 specification or expectation.

- 955 • COMPLETE
- 956 accumulation of actions, items, or activities that have been completed, independent
- 957 of the outcome.

- 958 • FAILED
- 959 accumulation of actions or activities that were attempted, but failed to complete or
- 960 resulted in an unexpected or unacceptable outcome.

- 961 • GOOD
- 962 accumulation of actions, items, or activities being counted that conform to specifi-
- 963 cation or expectation.

- 964 • REMAINING
- 965 accumulation of actions, items, or activities yet to be counted.

- 966 • TARGET
- 967 goal of the operation or process.

968 **5.2.30 DateCode**

- 969 time and date code associated with a material or other physical item.
- 970 The value of `DateCode` **MUST** be `dateTime`. See *Section 6.1.5 - dateTime*.

971 **5.2.30.1 Subtypes of DateCode**

- 972 • EXPIRATION
- 973 time and date code relating to the expiration or end of useful life for a material or
- 974 other physical item.
- 975 • FIRST_USE
- 976 time and date code relating the first use of a material or other physical item.
- 977 • MANUFACTURE
- 978 time and date code relating to the production of a material or other physical item.

979 **5.2.31 DeactivationCount**

980 accumulation of the number of times a function has attempted to, or is planned to attempt

981 to, deactivate or cease.

982 The default subType of DeactivationCount is ALL.

983 The value of DeactivationCount **MUST** be integer.

984 **5.2.31.1 Subtypes of DeactivationCount**

- 985 • ABORTED
- 986 accumulation of actions or activities that were attempted, but terminated before they
- 987 could be completed.
- 988 • ALL
- 989 accumulation of all actions, items, or activities being counted independent of the
- 990 outcome.
- 991 • BAD
- 992 accumulation of actions, items, or activities being counted that do not conform to
- 993 specification or expectation.
- 994 • COMPLETE
- 995 accumulation of actions, items, or activities that have been completed, independent
- 996 of the outcome.

- 997 • FAILED
- 998 accumulation of actions or activities that were attempted, but failed to complete or
- 999 resulted in an unexpected or unacceptable outcome.
- 1000 • GOOD
- 1001 accumulation of actions, items, or activities being counted that conform to specifi-
- 1002 cation or expectation.
- 1003 • REMAINING
- 1004 accumulation of actions, items, or activities yet to be counted.
- 1005 • TARGET
- 1006 goal of the operation or process.

1007 **5.2.32 DeviceAdded**

1008 Universally Unique Identifier (UUID) of new device added to an *MTConnect Agent*.

1009 **5.2.33 DeviceChanged**

1010 UUID of the device whose *metadata* has changed.

1011 **5.2.34 DeviceRemoved**

1012 UUID of a device removed from an *MTConnect Agent*.

1013 **5.2.35 DeviceUuid**

1014 identifier of another piece of equipment that is temporarily associated with a component

1015 of this piece of equipment to perform a particular function.

1016 **5.2.36 Direction**

1017 direction of motion.

1018 <<deprecated>> DirectionEnum Enumeration:

- 1019 • <<deprecated>> CLOCKWISE
- 1020 clockwise rotation using the right-hand rule.
- 1021 • <<deprecated>> COUNTER_CLOCKWISE
- 1022 counter-clockwise rotation using the right-hand rule.
- 1023 • <<deprecated>> NEGATIVE
- 1024 • <<deprecated>> POSITIVE

1025 A subType **MUST** always be specified.

1026 5.2.36.1 Subtypes of Direction

- 1027 • LINEAR
- 1028 direction of motion of a linear motion.
- 1029 DirectionLinearEnum Enumeration:
- 1030 – NEGATIVE
- 1031 linear position is decreasing.
- 1032 – NONE
- 1033 no direction.
- 1034 – POSITIVE
- 1035 linear position is increasing.
- 1036 • ROTARY
- 1037 rotational direction of a rotary motion using the right hand rule convention.
- 1038 DirectionRotaryEnum Enumeration:
- 1039 – CLOCKWISE
- 1040 clockwise rotation using the right-hand rule.
- 1041 – COUNTER_CLOCKWISE
- 1042 counter-clockwise rotation using the right-hand rule.
- 1043 – NONE
- 1044 no direction.

1045 **5.2.37 DoorState**

1046 operational state of a `Door` component or composition element.

1047 `DoorStateEnum` Enumeration:

1048 • `CLOSED`

1049 `Door` is closed to the point of a positive confirmation.

1050 • `OPEN`

1051 `Door` is open to the point of a positive confirmation.

1052 • `UNLATCHED`

1053 `Door` is not closed to the point of a positive confirmation and not open to the point
1054 of a positive confirmation.

1055 It is in an intermediate position.

1056 **5.2.38 EmergencyStop**

1057 state of the emergency stop signal for a piece of equipment, controller path, or any other
1058 component or subsystem of a piece of equipment.

1059 `EmergencyStopEnum` Enumeration:

1060 • `ARMED`

1061 emergency stop circuit is complete and the piece of equipment, component, or com-
1062 position is allowed to operate.

1063 • `TRIGGERED`

1064 operation of the piece of equipment, component, or composition is inhibited.

1065 **5.2.39 EndOfBar**

1066 indication of whether the end of a piece of bar stock being feed by a bar feeder has been
1067 reached.

1068 `EndOfBarEnum` Enumeration:

1069 • NO
 1070 EndOfBar has not been reached.

1071 • YES
 1072 EndOfBar has been reached.

1073 The default subType of EndOfBar is PRIMARY.

1074 **5.2.39.1 Subtypes of EndOfBar**

1075 • AUXILIARY
 1076 when multiple locations on a piece of bar stock are referenced as the indication for
 1077 the EndOfBar, the additional location(s) **MUST** be designated as AUXILIARY
 1078 indication(s) for the EndOfBar.

1079 • PRIMARY
 1080 specific applications **MAY** reference one or more locations on a piece of bar stock
 1081 as the indication for the EndOfBar.

1082 The main or most important location **MUST** be designated as the PRIMARY indica-
 1083 tion for the EndOfBar.

1084 If no subType is specified, PRIMARY **MUST** be the default EndOfBar indica-
 1085 tion.

1086 **5.2.40 EquipmentMode**

1087 indication that a piece of equipment, or a sub-part of a piece of equipment, is performing
 1088 specific types of activities.

1089 EquipmentModeEnum Enumeration:

1090 • OFF
 1091 equipment is not functioning in the mode designated by the subType.

1092 • ON
 1093 equipment is functioning in the mode designated by the subType.

1094 A subType **MUST** always be specified.

1095 **5.2.40.1 Subtypes of EquipmentMode**

- 1096 • **DELAY**
 1097 elapsed time of a temporary halt of action.
- 1098 • **LOADED**
 1099 indication that the sub-parts of a piece of equipment are under load.
 1100 Example: For traditional machine tools, this is an indication that the cutting tool is
 1101 assumed to be engaged with the part.
- 1102 • **OPERATING**
 1103 indication that the major sub-parts of a piece of equipment are powered or perform-
 1104 ing any activity whether producing a part or product or not.
 1105 Example: For traditional machine tools, this includes when the piece of equipment
 1106 is **WORKING** or it is idle.
- 1107 • **POWERED**
 1108 indication that primary power is applied to the piece of equipment and, as a min-
 1109 imum, the controller or logic portion of the piece of equipment is powered and
 1110 functioning or components that are required to remain on are powered.
 1111 Example: Heaters for an extrusion machine that required to be powered even when
 1112 the equipment is turned off.
- 1113 • **WORKING**
 1114 indication that a piece of equipment is performing any activity, the equipment is
 1115 active and performing a function under load or not.
 1116 Example: For traditional machine tools, this includes when the piece of equipment
 1117 is **LOADED**, making rapid moves, executing a tool change, etc.

1118 **5.2.41 Execution**

1119 execution status of the `Component`.

1120 `ExecutionEnum` Enumeration:

- 1121 • **ACTIVE**
 1122 `Component` is actively executing an instruction.

- 1123 • FEED_HOLD
- 1124 motion of the active axes are commanded to stop at their current position.
- 1125 • INTERRUPTED
- 1126 Component suspends the execution of the program due to an external signal.
- 1127 Action is required to resume execution.
- 1128 • OPTIONAL_STOP
- 1129 command from the program has intentionally interrupted execution.
- 1130 The Component **MAY** have another state that indicates if the execution is inter-
- 1131 rupted or the execution ignores the interrupt instruction.
- 1132 • PROGRAM_COMPLETED
- 1133 program completed execution.
- 1134 • PROGRAM_STOPPED
- 1135 command from the program has intentionally interrupted execution.
- 1136 Action is required to resume execution.
- 1137 • READY
- 1138 Component is ready to execute instructions.
- 1139 It is currently idle.
- 1140 • STOPPED
- 1141 Component program is not READY to execute.
- 1142 • WAIT
- 1143 Component suspends execution while a secondary operation executes.
- 1144 Execution resumes automatically once the secondary operation completes.

1145 5.2.42 Firmware

- 1146 embedded software of a Component .
- 1147 A subType **MUST** always be specified.

1148 **5.2.42.1 Subtypes of Firmware**

- 1149 • `INSTALL_DATE`
 1150 date the hardware or software was installed.
 1151 The value of `Firmware` **MUST** be `dateTime`. See *Section 6.1.5 - dateTime*.
 1152 `dateTime` Enumeration:
- 1153 • `LICENSE`
 1154 license code to validate or activate the hardware or software.
- 1155 • `MANUFACTURER`
 1156 corporate identity for the maker of the hardware or software.
- 1157 • `RELEASE_DATE`
 1158 date the hardware or software was released for general use.
 1159 The value of `Firmware` **MUST** be `dateTime`. See *Section 6.1.5 - dateTime*.
 1160 `dateTime` Enumeration:
- 1161 • `VERSION`
 1162 version of the hardware or software.

1163 **5.2.43 FixtureId**

1164 identifier for a fixture.

1165 **5.2.44 FunctionalMode**

- 1166 current intended production status of the `Component`.
- 1167 `FunctionalModeEnum` Enumeration:
- 1168 • `MAINTENANCE`
 1169 `Component` is not currently producing product.
 1170 It is currently being repaired, waiting to be repaired, or has not yet been returned to
 1171 a normal production status after maintenance has been performed.

- 1172 • PROCESS_DEVELOPMENT
- 1173 Component is being used to prove-out a new process, testing of equipment or
- 1174 processes, or any other active use that does not result in the production of product.
- 1175 • PRODUCTION
- 1176 Component is currently producing product, ready to produce product, or its current
- 1177 intended use is to be producing product.
- 1178 • SETUP
- 1179 Component is not currently producing product.
- 1180 It is being prepared or modified to begin production of product.
- 1181 • TEARDOWN
- 1182 Component is not currently producing product.
- 1183 Typically, it has completed the production of a product and is being modified or
- 1184 returned to a neutral state such that it may then be prepared to begin production of a
- 1185 different product.

1186 5.2.45 Hardness

- 1187 hardness of a material.
- 1188 The value of Hardness **MUST** be float.
- 1189 A subType **MUST** always be specified.

1190 5.2.45.1 Subtypes of Hardness

- 1191 • BRINELL
- 1192 scale to measure the resistance to deformation of a surface.
- 1193 • LEEB
- 1194 scale to measure the elasticity of a surface.
- 1195 • MOHS
- 1196 scale to measure the resistance to scratching of a surface.

- 1197 • ROCKWELL
- 1198 scale to measure the resistance to deformation of a surface.
- 1199 • SHORE
- 1200 scale to measure the resistance to deformation of a surface.
- 1201 • VICKERS
- 1202 scale to measure the resistance to deformation of a surface.

1203 **5.2.46 Hardware**

- 1204 hardware of a Component.
- 1205 A subType **MUST** always be specified.

1206 **5.2.46.1 Subtypes of Hardware**

- 1207 • INSTALL_DATE
- 1208 date the hardware or software was installed.
- 1209 The value of Hardware **MUST** be dateTime. See *Section 6.1.5 - dateTime*.
- 1210 dateTime Enumeration:
- 1211 • LICENSE
- 1212 license code to validate or activate the hardware or software.
- 1213 • MANUFACTURER
- 1214 corporate identity for the maker of the hardware or software.
- 1215 • RELEASE_DATE
- 1216 date the hardware or software was released for general use.
- 1217 The value of Hardware **MUST** be dateTime. See *Section 6.1.5 - dateTime*.
- 1218 dateTime Enumeration:
- 1219 • VERSION
- 1220 version of the hardware or software.

1221 **5.2.47 Library**

1222 software library on a Component

1223 A subType **MUST** always be specified.

1224 **5.2.47.1 Subtypes of Library**

1225 • INSTALL_DATE

1226 date the hardware or software was installed.

1227 The value of Library **MUST** be dateTime. See *Section 6.1.5 - dateTime*.

1228 dateTime Enumeration:

1229 • LICENSE

1230 license code to validate or activate the hardware or software.

1231 • MANUFACTURER

1232 corporate identity for the maker of the hardware or software.

1233 • RELEASE_DATE

1234 date the hardware or software was released for general use.

1235 The value of Library **MUST** be dateTime. See *Section 6.1.5 - dateTime*.

1236 dateTime Enumeration:

1237 • VERSION

1238 version of the hardware or software.

1239 **5.2.48 <<deprecated>>Line**

1240 current line of code being executed.

1241 **DEPRECATED** in *Version 1.4.0*.

1242 **5.2.48.1 Subtypes of Line**

- 1243 • MAXIMUM
- 1244 maximum line number of the code being executed.
- 1245 • MINIMUM
- 1246 minimum line number of the code being executed.

1247 **5.2.49 LineLabel**

1248 identifier for a Block of code in a Program.

1249 **5.2.50 LineNumber**

1250 position of a block of program code within a control program.

1251 The value of LineNumber **MUST** be integer.

1252 **5.2.50.1 Subtypes of LineNumber**

- 1253 • ABSOLUTE
- 1254 position of a block of program code relative to the beginning of the control program.
- 1255 • INCREMENTAL
- 1256 position of a block of program code relative to the occurrence of the last LineLa-
- 1257 bel encountered in the control program.

1258 **5.2.51 LoadCount**

1259 accumulation of the number of times an operation has attempted to, or is planned to attempt
1260 to, load materials, parts, or other items.

1261 The default subType of LoadCount is ALL.

1262 The value of LoadCount **MUST** be integer.

1263 **5.2.51.1 Subtypes of LoadCount**

- 1264 • ABORTED
- 1265 accumulation of actions or activities that were attempted, but terminated before they
- 1266 could be completed.
- 1267 • ALL
- 1268 accumulation of all actions, items, or activities being counted independent of the
- 1269 outcome.
- 1270 • BAD
- 1271 accumulation of actions, items, or activities being counted that do not conform to
- 1272 specification or expectation.
- 1273 • COMPLETE
- 1274 accumulation of actions, items, or activities that have been completed, independent
- 1275 of the outcome.
- 1276 • FAILED
- 1277 accumulation of actions or activities that were attempted, but failed to complete or
- 1278 resulted in an unexpected or unacceptable outcome.
- 1279 • GOOD
- 1280 accumulation of actions, items, or activities being counted that conform to specifi-
- 1281 cation or expectation.
- 1282 • REMAINING
- 1283 accumulation of actions, items, or activities yet to be counted.
- 1284 • TARGET
- 1285 goal of the operation or process.

1286 **5.2.52 LockState**

- 1287 state or operating mode of a Lock.
- 1288 LockStateEnum Enumeration:

- 1289 • LOCKED
- 1290 mechanism is engaged and preventing the associated Component from being opened
- 1291 or operated.
- 1292 • UNLOCKED
- 1293 mechanism is disengaged and the associated Component is able to be opened or
- 1294 operated.

1295 **5.2.53 MTConnectVersion**

1296 reference version of the MTConnect Standard supported by the *adapter*.

1297 **5.2.54 MaintenanceList**

- 1298 actions or activities to be performed in support of a piece of equipment.
- 1299 If the INTERVAL key is not provided, it is assumed ABSOLUTE.
- 1300 If the DIRECTION key is not provided, it is assumed UP.
- 1301 If the UNITS key is not provided, it is assumed to be COUNT.
- 1302 The Entry key **MUST** be one or more from the MaintenanceListResult keys.
- 1303 MaintenanceListResult keys:
- 1304 • VALUE
 - 1305 current interval value of the activity.
 - 1306 The value of VALUE **MUST** be float.
 - 1307 • INTERVAL
 - 1308 interval of the value observed.
 - 1309 MaintenanceListIntervalEnum Enumeration:
 - 1310 – ABSOLUTE
 - 1311 – INCREMENTAL

- 1312 • NEXT_SERVICE_DATE
- 1313 next date/time stamp that maintenance should be performed.
- 1314 The value of NEXT_SERVICE_DATE **MUST** be dateTime. See *Section 6.1.5 -*
- 1315 *dateTime*.
- 1316 • RESET
- 1317 last date/time stamp of the observation was reset.
- 1318 The value of RESET **MUST** be dateTime. See *Section 6.1.5 - dateTime*.
- 1319 • SEVERITY
- 1320 level of severity on a scale of 1-10.
- 1321 The value of SEVERITY **MUST** be integer.
- 1322 • DIRECTION
- 1323 direction of the value observed.
- 1324 MaintenanceListDirectionEnum Enumeration:
- 1325 – DOWN
- 1326 – UP
- 1327 • NAME
- 1328 identifier of the maintenance activity.
- 1329 The value of NAME **MUST** be string.
- 1330 • LAST_SERVICE_DATE
- 1331 last date/time stamp that maintenance was performed.
- 1332 The value of LAST_SERVICE_DATE **MUST** be dateTime. See *Section 6.1.5 -*
- 1333 *dateTime*.
- 1334 • UNITS
- 1335 same as DataItem units. See *MTConnect Standard: Part 2.0 - Device Informa-*
- 1336 *tion Model*.
- 1337 The value of UNITS **MUST** be one of the UnitEnum enumeration.
- 1338 • TARGET
- 1339 target value of the next maintenance.
- 1340 The value of TARGET **MUST** be float.

1341 5.2.55 Material

1342 identifier of a material used or consumed in the manufacturing process.

1343 5.2.56 MaterialLayer

1344 identifies the layers of material applied to a part or product as part of an additive manufac-
1345 turing process.

1346 The value of `MaterialLayer` **MUST** be integer.

1347 5.2.56.1 Subtypes of MaterialLayer

- 1348 • ACTUAL
- 1349 measured or reported value of an observation.
- 1350 • TARGET
- 1351 goal of the operation or process.

1352 5.2.57 Message

1353 information to be transferred from a piece of equipment to a client software application.

1354 5.2.58 Network

1355 network details of a `Component`.

1356 A `subType` **MUST** always be specified.

1357 5.2.58.1 Subtypes of Network

- 1358 • GATEWAY
- 1359 Gateway for the component network.

- 1360 • IPV4_ADDRESS
- 1361 IPV4 network address of the component.
- 1362 • IPV6_ADDRESS
- 1363 IPV6 network address of the component.
- 1364 • MAC_ADDRESS
- 1365 Media Access Control Address.
- 1366 The unique physical address of the network hardware.
- 1367 • SUBNET_MASK
- 1368 SubNet mask for the component network.
- 1369 • VLAN_ID
- 1370 layer2 Virtual Local Network (VLAN) ID for the component network.
- 1371 • WIRELESS
- 1372 identifies whether the connection type is wireless.
- 1373 NetworkWirelessEnum Enumeration:
- 1374 – NO
- 1375 – YES

1376 5.2.59 OperatingMode

1377 state of Component or Composition that describes the automatic or manual operation
1378 of the entity.

1379 OperatingModeEnum Enumeration:

- 1380 • AUTOMATIC
- 1381 automatically execute instructions from a recipe or program.
- 1382 Note: Setpoint comes from a recipe.
- 1383 • MANUAL
- 1384 execute instructions from an external agent or person.
- 1385 Note 1 to entry: Valve or switch is manipulated by an agent/person.

1386 Note 2 to entry: Direct control of the PID output. % of the range: A user
1387 manually sets the % output, not the setpoint.

1388 • SEMI_AUTOMATIC
1389 executes a single instruction from a recipe or program.

1390 Note 1 to entry: Setpoint is entered and fixed, but the PID is controlling.

1391 Note 2 to entry: Still goes through the PID control system.

1392 Note 3 to entry: Manual fixed entry from a recipe.

1393 5.2.60 OperatingSystem

1394 Operating System (OS) of a Component.

1395 A subType **MUST** always be specified.

1396 5.2.60.1 Subtypes of OperatingSystem

1397 • INSTALL_DATE
1398 date the hardware or software was installed.
1399 The value of OperatingSystem **MUST** be dateTime. See *Section 6.1.5 -*
1400 *dateTime*.

1401 dateTime Enumeration:

1402 • LICENSE
1403 license code to validate or activate the hardware or software.

1404 • MANUFACTURER
1405 corporate identity for the maker of the hardware or software.

1406 • RELEASE_DATE
1407 date the hardware or software was released for general use.
1408 The value of OperatingSystem **MUST** be dateTime. See *Section 6.1.5 -*
1409 *dateTime*.

1410 dateTime Enumeration:

1411 • VERSION
1412 version of the hardware or software.

1413 5.2.61 OperatorId

1414 identifier of the person currently responsible for operating the piece of equipment.

1415 5.2.62 PalletId

1416 identifier for a pallet.

1417 5.2.63 PartCount

1418 aggregate count of parts.

1419 The value of PartCount **MUST** be integer.

1420 The default subType of PartCount is ALL.

1421 5.2.63.1 Subtypes of PartCount

- 1422 • ABORTED
- 1423 accumulation of actions or activities that were attempted, but terminated before they
- 1424 could be completed.
- 1425 • ALL
- 1426 accumulation of all actions, items, or activities being counted independent of the
- 1427 outcome.
- 1428 • BAD
- 1429 accumulation of actions, items, or activities being counted that do not conform to
- 1430 specification or expectation.
- 1431 • COMPLETE
- 1432 accumulation of actions, items, or activities that have been completed, independent
- 1433 of the outcome.
- 1434 • FAILED
- 1435 accumulation of actions or activities that were attempted, but failed to complete or
- 1436 resulted in an unexpected or unacceptable outcome.

- 1437 • GOOD
- 1438 accumulation of actions, items, or activities being counted that conform to specifi-
- 1439 cation or expectation.
- 1440 • REMAINING
- 1441 accumulation of actions, items, or activities yet to be counted.
- 1442 • TARGET
- 1443 goal of the operation or process.

1444 5.2.64 PartCountType

1445 interpretation of PART_COUNT.

1446 PartCountTypeEnum Enumeration:

- 1447 • BATCH
- 1448 pre-specified group of items.
- 1449 • EACH
- 1450 count is of individual items.

1451 5.2.65 PartDetect

1452 indication designating whether a part or work piece has been detected or is present.

1453 PartDetectEnum Enumeration:

- 1454 • NOT_PRESENT
- 1455 part or work piece is not detected or is not present.
- 1456 • PRESENT
- 1457 part or work piece is detected or is present.

1458 **5.2.66 PartGroupId**

1459 identifier given to a collection of individual parts.

1460 If no subType is specified, UUID is default.

1461 The default subType of PartGroupId is UUID.

1462 **5.2.66.1 Subtypes of PartGroupId**

1463 • BATCH

1464 identifier that references a group of parts produced in a batch.

1465 • HEAT_TREAT

1466 identifier used to reference a material heat number.

1467 • LOT

1468 identifier that references a group of parts tracked as a lot.

1469 • RAW_MATERIAL

1470 material that is used to produce parts.

1471 • UUID

1472 universally unique identifier as specified in ISO 11578 or RFC 4122.

1473 **5.2.67 PartId**

1474 identifier of a part in a manufacturing operation.

1475 **5.2.68 PartKindId**

1476 identifier given to link the individual occurrence to a class of parts, typically distinguished
1477 by a particular part design.

1478 If no subType is specified, UUID is default.

1479 The default subType of PartKindId is UUID.

1480 **5.2.68.1 Subtypes of PartKindId**

- 1481 • PART_FAMILY
- 1482 identifier given to a group of parts having similarities in geometry, manufacturing
- 1483 process, and/or functions.
- 1484 • PART_NAME
- 1485 word or set of words by which a part is known, addressed, or referred to.
- 1486 • PART_NUMBER
- 1487 identifier of a particular part design or model.
- 1488 • UUID
- 1489 universally unique identifier as specified in ISO 11578 or RFC 4122.

1490 **5.2.69 <<deprecated>>PartNumber**

- 1491 identifier of a part or product moving through the manufacturing process.
- 1492 **DEPRECATED** in *Version 1.7*. PART_NUMBER is now a subType of PART_KIND_ID.

1493 **5.2.70 PartProcessingState**

- 1494 particular condition of the part occurrence at a specific time.
- 1495 PartProcessingStateEnum Enumeration:
- 1496 • IN_PROCESS
 - 1497 part occurrence is actively being processed.
 - 1498 • IN_TRANSIT
 - 1499 part occurrence is being transported to its destination.
 - 1500 • NEEDS_PROCESSING
 - 1501 part occurrence is not actively being processed, but the processing has not ended.
 - 1502 Processing requirements exist that have not yet been fulfilled. This is the default
 - 1503 entry state when the part occurrence is originally received. In some cases, the part

- 1504 occurrence may return to this state while it waits for additional processing to be
 1505 performed.
- 1506 • PROCESSING_ENDED
- 1507 part occurrence is no longer being processed.
- 1508 A general state when the reason for termination is unknown.
- 1509 • PROCESSING_ENDED_ABORTED
- 1510 processing of the part occurrence has come to a premature end.
- 1511 • PROCESSING_ENDED_COMPLETE
- 1512 part occurrence has completed processing successfully.
- 1513 • PROCESSING_ENDED_LOST
- 1514 terminal state when the part occurrence has been removed from the equipment by
 1515 an external entity and it no longer exists at the equipment.
- 1516 • PROCESSING_ENDED_REJECTED
- 1517 part occurrence has been processed completely. However, the processing may have
 1518 a problem.
- 1519 • PROCESSING_ENDED_SKIPPED
- 1520 part occurrence has been skipped for processing on the piece of equipment.
- 1521 • PROCESSING_ENDED_STOPPED
- 1522 process has been stopped during the processing.
- 1523 The part occurrence will require special treatment.
- 1524 • TRANSIT_COMPLETE
- 1525 part occurrence has been placed at its designated destination.
- 1526 • WAITING_FOR_TRANSIT
- 1527 part occurrence is waiting for transit.

1528 5.2.71 PartStatus

1529 state or condition of a part.

1530 If unique identifier is given, part status is for that individual. If group identifier is given
 1531 without a unique identifier, then the status is assumed to be for the whole group.

1532 `PartStatusEnum` Enumeration:

- 1533 • FAIL
- 1534 part does not conform to some given requirements.
- 1535 • PASS
- 1536 part conforms to given requirements.

1537 **5.2.72 PartUniqueId**

1538 identifier given to a distinguishable, individual part.

1539 If no `subType` is specified, `UUID` is default.

1540 The default `subType` of `PartUniqueId` is `UUID`.

1541 **5.2.72.1 Subtypes of PartUniqueId**

- 1542 • `RAW_MATERIAL`
- 1543 material that is used to produce parts.
- 1544 • `SERIAL_NUMBER`
- 1545 serial number that uniquely identifies a specific part.
- 1546 • `UUID`
- 1547 universally unique identifier as specified in ISO 11578 or RFC 4122.

1548 **5.2.73 PathFeedrateOverride**

1549 value of a signal or calculation issued to adjust the feedrate for the axes associated with a
1550 `Path` component that may represent a single axis or the coordinated movement of multiple
1551 axes.

1552 The value of `PathFeedrateOverride` **MUST** be `float`.

1553 **5.2.73.1 Subtypes of PathFeedrateOverride**

- 1554 • JOG
- 1555 relating to momentary activation of a function or a movement.
- 1556 **DEPRECATION WARNING:** May be deprecated in the future.
- 1557 • PROGRAMMED
- 1558 directive value without offsets and adjustments.
- 1559 • RAPID
- 1560 performing an operation faster or in less time than nominal rate.

1561 **5.2.74 PathMode**

1562 describes the operational relationship between a `Path` entity and another `Path` entity for
 1563 pieces of equipment comprised of multiple logical groupings of controlled axes or other
 1564 logical operations.

1565 `PathModeEnum` Enumeration:

- 1566 • INDEPENDENT
- 1567 path is operating independently and without the influence of another path.
- 1568 • MASTER
- 1569 path provides information or state values that influences the operation of other `DataItem`
 1570 of similar type.
- 1571 • MIRROR
- 1572 axes associated with the path are mirroring the motion of the `MASTER` path.
- 1573 • SYNCHRONOUS
- 1574 physical or logical parts which are not physically connected to each other but are
 1575 operating together.

1576 5.2.75 PowerState

1577 indication of the status of the source of energy for an entity to allow it to perform its
 1578 intended function or the state of an enabling signal providing permission for the entity to
 1579 perform its functions.

1580 PowerStateEnum Enumeration:

- 1581 • OFF
- 1582 source of energy for an entity or the enabling signal providing permission for the
- 1583 entity to perform its function(s) is not present or is disconnected.
- 1584 • ON
- 1585 source of energy for an entity or the enabling signal providing permission for the
- 1586 entity to perform its function(s) is present and active.

1587 5.2.75.1 Subtypes of PowerState

- 1588 • CONTROL
- 1589 state of the enabling signal or control logic that enables or disables the function or
- 1590 operation of the entity.
- 1591 • LINE
- 1592 state of the power source for the entity.

1593 5.2.76 <<deprecated>>PowerStatus

1594 status of the Component.

1595 **DEPRECATED** in *Version 1.1.0*.

1596 <<deprecated>> PowerStatusEnum Enumeration:

- 1597 • <<deprecated>> OFF
- 1598 • <<deprecated>> ON

1599 5.2.77 ProcessAggregateId

1600 identifier given to link the individual occurrence to a group of related occurrences, such as
 1601 a process step in a process plan.

1602 5.2.77.1 Subtypes of ProcessAggregateId

1603 • ORDER_NUMBER
 1604 identifier of the authorization of the process occurrence. Synonyms include “job id”,
 1605 “work order”.

1606 • PROCESS_PLAN
 1607 identifier of the process plan that this occurrence belongs to. Synonyms include
 1608 “routing id”, “job id”.

1609 • PROCESS_STEP
 1610 identifier of the step in the process plan that this occurrence corresponds to. Syn-
 1611 onyms include “operation id”.

1612 5.2.78 ProcessKindId

1613 identifier given to link the individual occurrence to a class of processes or process defini-
 1614 tion.

1615 5.2.78.1 Subtypes of ProcessKindId

1616 • ISO_STEP_EXECUTABLE
 1617 reference to a ISO 10303 Executable.

1618 • PROCESS_NAME
 1619 word or set of words by which a process being executed (process occurrence) by the
 1620 device is known, addressed, or referred to.

1621 • UUID
 1622 universally unique identifier as specified in ISO 11578 or RFC 4122.

1623 **5.2.79 ProcessOccurrenceId**

1624 identifier of a process being executed by the device.

1625 **5.2.79.1 Subtypes of ProcessOccurrenceId**

- 1626 • ACTIVITY
- 1627 phase or segment of a recipe or program.
- 1628 • OPERATION
- 1629 step of a discrete manufacturing process.
- 1630 • RECIPE
- 1631 process as part of product production; can be a subprocess of a larger process.
- 1632 • SEGMENT
- 1633 phase of a recipe process.

1634 **5.2.80 ProcessState**

1635 particular condition of the process occurrence at a specific time.

1636 `ProcessStateEnum` Enumeration:

- 1637 • ABORTED
- 1638 process occurrence has come to a premature end and cannot be resumed.
- 1639 • ACTIVE
- 1640 process occurrence is actively executing.
- 1641 • COMPLETE
- 1642 process occurrence is now finished.
- 1643 • INITIALIZING
- 1644 device is preparing to execute the process occurrence.
- 1645 • INTERRUPTED
- 1646 process occurrence has been stopped and may be resumed.

- 1647 • READY
- 1648 process occurrence is ready to be executed.

1649 **5.2.81 ProcessTime**

1650 time and date associated with an activity or event.

1651 **5.2.81.1 Subtypes of ProcessTime**

- 1652 • COMPLETE
- 1653 time and date associated with the completion of an activity or event.
- 1654 • START
- 1655 boundary when an activity or an event commences.
- 1656 • TARGET_COMPLETION
- 1657 projected time and date associated with the end or completion of an activity or event.

1658 **5.2.82 Program**

1659 name of the logic or motion program being executed by the Controller component.

1660 **5.2.82.1 Subtypes of Program**

- 1661 • ACTIVE
- 1662 identity of the logic or motion program currently executing.
- 1663 • ACTIVITY
- 1664 phase or segment of a recipe or program.
- 1665 • MAIN
- 1666 identity of the primary logic or motion program currently being executed.
- 1667 It is the starting nest level in a call structure and may contain calls to sub programs.

- 1668 • OPERATION
- 1669 step of a discrete manufacturing process.
- 1670 • RECIPE
- 1671 process as part of product production; can be a subprocess of a larger process.
- 1672 • SCHEDULE
- 1673 identity of a control program that is used to specify the order of execution of other
- 1674 programs.
- 1675 • SEGMENT
- 1676 phase of a recipe process.

1677 **5.2.83 ProgramComment**

1678 comment or non-executable statement in the control program.

1679 **5.2.84 ProgramEdit**

1680 indication of the status of the `Controller` components program editing mode.

1681 A program may be edited while another is executed.

1682 `ProgramEditEnum` Enumeration:

- 1683 • ACTIVE
- 1684 `Controller` is in the program edit mode.
- 1685 • NOT_READY
- 1686 `Controller` is being inhibited by a function from entering the program edit mode.
- 1687 • READY
- 1688 `Controller` is capable of entering the program edit mode and no function is
- 1689 inhibiting a change to that mode.

1690 **5.2.85 ProgramEditName**

1691 name of the program being edited.

1692 This is used in conjunction with `ProgramEdit` when in `ACTIVE` state.

1693 **5.2.86 ProgramHeader**

1694 non-executable header section of the control program.

1695 The default `subType` of `ProgramHeader` is `MAIN`.

1696 **5.2.86.1 Subtypes of ProgramHeader**

1697 • `ACTIVE`

1698 identity of the logic or motion program currently executing.

1699 • `MAIN`

1700 identity of the primary logic or motion program currently being executed.

1701 It is the starting nest level in a call structure and may contain calls to sub programs.

1702 • `SCHEDULE`

1703 identity of a control program that is used to specify the order of execution of other
1704 programs.

1705 **5.2.87 ProgramLocation**

1706 URI for the source file associated with `Program`.

1707 **5.2.87.1 Subtypes of ProgramLocation**

1708 • `ACTIVE`

1709 identity of the logic or motion program currently executing.

- 1710 • MAIN
- 1711 identity of the primary logic or motion program currently being executed.
- 1712 It is the starting nest level in a call structure and may contain calls to sub programs.
- 1713 • SCHEDULE
- 1714 identity of a control program that is used to specify the order of execution of other
- 1715 programs.

1716 **5.2.88 ProgramLocationType**

- 1717 defines whether the logic or motion program defined by `Program` is being executed from
 1718 the local memory of the controller or from an outside source.

1719 `ProgramLocationTypeEnum` Enumeration:

- 1720 • EXTERNAL
- 1721 not managed by the controller.
- 1722 • LOCAL
- 1723 managed by the controller.

1724 **5.2.88.1 Subtypes of ProgramLocationType**

- 1725 • ACTIVE
- 1726 identity of the logic or motion program currently executing.
- 1727 • MAIN
- 1728 identity of the primary logic or motion program currently being executed.
- 1729 It is the starting nest level in a call structure and may contain calls to sub programs.
- 1730 • SCHEDULE
- 1731 identity of a control program that is used to specify the order of execution of other
- 1732 programs.

1733 5.2.89 ProgramNestLevel

1734 indication of the nesting level within a control program that is associated with the code or
 1735 instructions that is currently being executed.

1736 If an initial value is not defined, the nesting level associated with the highest or initial
 1737 nesting level of the program **MUST** default to zero (0).

1738 The value of ProgramNestLevel **MUST** be integer.

1739 5.2.90 RotaryMode

1740 current operating mode for a Rotary type axis.

1741 RotaryModeEnum Enumeration:

- 1742 • CONTOUR
 1743 position of the axis is being interpolated.
- 1744 • INDEX
 1745 axis is configured to index.
- 1746 • SPINDLE
 1747 axis is functioning as a spindle.

1748 5.2.91 RotaryVelocityOverride

1749 percentage change to the velocity of the programmed velocity for a Rotary axis.

1750 This command represents a percentage change to the velocity calculated by a logic or
 1751 motion program or set by a switch for a Rotary type axis.

1752 The value of RotaryVelocityOverride **MUST** be float.

1753 5.2.92 Rotation

1754 three space angular rotation relative to a coordinate system.

1755 The units of Rotation **MUST** be DEGREE_3D.

1756 5.2.93 SensorAttachment

1757 *attachment* between a sensor and an entity.

1758 The Entry key **MUST** be one or more from the SensorAttachmentResult keys.

1759 SensorAttachmentResult keys:

1760 • SENSOR_ID

1761 The identity of a sensor used to observe some measurement of an item.

1762 The value of SENSOR_ID **MUST** be string.

1763 5.2.94 SerialNumber

1764 serial number associated with a Component, Asset, or Device.

1765 5.2.95 SpecificationLimit

1766 set of limits defining a range of values designating acceptable performance for a variable.

1767 The Entry key **MUST** be one or more from the SpecificationLimitResult
1768 keys.

1769 SpecificationLimitResult keys:

1770 • UPPER_LIMIT

1771 upper conformance boundary for a variable.

1772 Note: immediate concern or action may be required.

1773 The value of UPPER_LIMIT **MUST** be float.

1774 • NOMINAL

1775 ideal or desired value for a variable.

1776 The value of NOMINAL **MUST** be float.

- 1777 • LOWER_LIMIT
- 1778 lower conformance boundary for a variable.
- 1779 Note: immediate concern or action may be required.
- 1780 The value of LOWER_LIMIT **MUST** be float.

1781 **5.2.96 SpindleInterlock**

1782 indication of the status of the spindle for a piece of equipment when power has been
1783 removed and it is free to rotate.

1784 SpindleInterlockEnum Enumeration:

- 1785 • ACTIVE
- 1786 power has been removed and the spindle cannot be operated.
- 1787 • INACTIVE
- 1788 spindle has not been deactivated.

1789 **5.2.97 ToolAssetId**

1790 identifier of an individual tool asset.

1791 **5.2.98 ToolGroup**

1792 identifier for the tool group associated with a specific tool. Commonly used to designate
1793 spare tools.

1794 **5.2.99 <<deprecated>>ToolId**

1795 identifier of the tool currently in use for a given Path.

1796 **DEPRECATED** in *Version 1.2.0*. See TOOL_ASSET_ID.

1797 **5.2.100 ToolNumber**

1798 identifier assigned by the Controller component to a cutting tool when in use by a
1799 piece of equipment.

1800 **5.2.101 ToolOffset**

1801 reference to the tool offset variables applied to the active cutting tool associated with a
1802 Path in a Controller type component.

1803 The value of ToolOffset **MUST** be float.

1804 A subType **MUST** always be specified.

1805 **5.2.101.1 Subtypes of ToolOffset**

- 1806 • LENGTH
1807 reference to a length type tool offset variable.
- 1808 • RADIAL
1809 reference to a radial type tool offset variable.

1810 **5.2.102 TransferCount**

1811 accumulation of the number of times an operation has attempted to, or is planned to attempt
1812 to, transfer materials, parts, or other items from one location to another.

1813 The default subType of TransferCount is ALL.

1814 The value of TransferCount **MUST** be integer.

1815 **5.2.102.1 Subtypes of TransferCount**

- 1816 • ABORTED
1817 accumulation of actions or activities that were attempted, but terminated before they
1818 could be completed.

- 1819 • ALL
- 1820 accumulation of all actions, items, or activities being counted independent of the
- 1821 outcome.
- 1822 • BAD
- 1823 accumulation of actions, items, or activities being counted that do not conform to
- 1824 specification or expectation.
- 1825 • COMPLETE
- 1826 accumulation of actions, items, or activities that have been completed, independent
- 1827 of the outcome.
- 1828 • FAILED
- 1829 accumulation of actions, items, or activities being counted that do not conform to
- 1830 specification or expectation.
- 1831 • GOOD
- 1832 accumulation of actions, items, or activities being counted that conform to specifi-
- 1833 cation or expectation.
- 1834 • REMAINING
- 1835 accumulation of actions, items, or activities yet to be counted.
- 1836 • TARGET
- 1837 goal of the operation or process.

1838 **5.2.103 Translation**

- 1839 three space linear translation relative to a coordinate system.
- 1840 The units of Translation **MUST** be MILLIMETER_3D.

1841 **5.2.104 UnloadCount**

- 1842 accumulation of the number of times an operation has attempted to, or is planned to attempt
- 1843 to, unload materials, parts, or other items.
- 1844 The default subType of UnloadCount is ALL.
- 1845 The value of UnloadCount **MUST** be integer.

1846 **5.2.104.1 Subtypes of UnloadCount**

- 1847 • ABORTED
- 1848 accumulation of actions or activities that were attempted, but terminated before they
- 1849 could be completed.
- 1850 • ALL
- 1851 accumulation of all actions, items, or activities being counted independent of the
- 1852 outcome.
- 1853 • BAD
- 1854 accumulation of actions, items, or activities being counted that do not conform to
- 1855 specification or expectation.
- 1856 • COMPLETE
- 1857 accumulation of actions, items, or activities that have been completed, independent
- 1858 of the outcome.
- 1859 • FAILED
- 1860 accumulation of actions, items, or activities being counted that do not conform to
- 1861 specification or expectation.
- 1862 • GOOD
- 1863 accumulation of actions, items, or activities being counted that conform to specifi-
- 1864 cation or expectation.
- 1865 • REMAINING
- 1866 accumulation of actions, items, or activities yet to be counted.
- 1867 • TARGET
- 1868 goal of the operation or process.

1869 **5.2.105 User**

- 1870 identifier of the person currently responsible for operating the piece of equipment.
- 1871 A subType **MUST** always be specified.

1872 **5.2.105.1 Subtypes of User**

- 1873 • MAINTENANCE
- 1874 identifier of the person currently responsible for performing maintenance on the
- 1875 piece of equipment.
- 1876 • OPERATOR
- 1877 identifier of the person currently responsible for operating the piece of equipment.
- 1878 • SET_UP
- 1879 identifier of the person currently responsible for preparing a piece of equipment for
- 1880 production or restoring the piece of equipment to a neutral state after production.

1881 **5.2.106 ValveState**

1882 state of a valve is one of open, closed, or transitioning between the states.

1883 `ValveStateEnum` Enumeration:

- 1884 • CLOSED
- 1885 `ValveState` where flow is not possible, the aperture is static, and the valve is
- 1886 completely shut.
- 1887 • CLOSING
- 1888 valve is transitioning from an OPEN state to a CLOSED state.
- 1889 • OPEN
- 1890 `ValveState` where flow is allowed and the aperture is static.
- 1891 Note: For a binary value, OPEN indicates the valve has the maximum
- 1892 possible aperture.
- 1893 • OPENING
- 1894 valve is transitioning from a CLOSED state to an OPEN state.

1895 **5.2.106.1 Subtypes of ValveState**

- 1896 • ACTUAL
- 1897 measured or reported value of an observation.
- 1898 • PROGRAMMED
- 1899 directive value without offsets and adjustments.

1900 **5.2.107 Variable**

1901 data whose meaning may change over time due to changes in the operation of a piece of
 1902 equipment or the process being executed on that piece of equipment.

1903 **5.2.108 WaitState**

1904 indication of the reason that `Execution` is reporting a value of `WAIT`.

1905 `WaitStateEnum` Enumeration:

- 1906 • MATERIAL_LOAD
- 1907 execution is waiting while material is being loaded.
- 1908 • MATERIAL_UNLOAD
- 1909 execution is waiting while material is being unloaded.
- 1910 • PART_LOAD
- 1911 execution is waiting while one or more discrete workpieces are being loaded.
- 1912 • PART_UNLOAD
- 1913 execution is waiting while one or more discrete workpieces are being unloaded.
- 1914 • PAUSING
- 1915 execution is waiting while the equipment is pausing but the piece of equipment has
- 1916 not yet reached a fully paused state.
- 1917 • POWERING_DOWN
- 1918 execution is waiting while the equipment is powering down but has not fully reached
- 1919 a stopped state.

- 1920 • POWERING_UP
- 1921 execution is waiting while the equipment is powering up and is not currently avail-
- 1922 able to begin producing parts or products.
- 1923 • RESUMING
- 1924 execution is waiting while the equipment is resuming the production cycle but has
- 1925 not yet resumed execution.
- 1926 • SECONDARY_PROCESS
- 1927 execution is waiting while another process is completed before the execution can
- 1928 resume.
- 1929 • TOOL_LOAD
- 1930 execution is waiting while a tool or tooling is being loaded.
- 1931 • TOOL_UNLOAD
- 1932 execution is waiting while a tool or tooling is being unloaded.

1933 5.2.109 Wire

- 1934 identifier for the type of wire used as the cutting mechanism in Electrical Discharge Ma-
- 1935 chining or similar processes.

1936 5.2.110 WorkOffset

- 1937 offset variables for a work piece or part associated with a Path in a Controller type
- 1938 component.

1939 5.2.111 WorkholdingId

- 1940 identifier for the current workholding or part clamp in use by a piece of equipment.

1941 5.3 Sample Types

- 1942 This section provides semantic information for Sample types.

1943 5.3.1 Acceleration

1944 positive rate of change of velocity.

1945 The units of Acceleration **MUST** be MILLIMETER/SECOND².

1946 The default subType of Acceleration is ACTUAL.

1947 5.3.1.1 Subtypes of Acceleration

1948 • ACTUAL

1949 measured or reported value of an observation.

1950 • COMMANDED

1951 directive value including adjustments such as an offset or overrides.

1952 • PROGRAMMED

1953 directive value without offsets and adjustments.

1954 5.3.2 AccumulatedTime

1955 accumulated time for an activity or event.

1956 The units of AccumulatedTime **MUST** be SECOND.

1957 5.3.3 <<deprecated>>Amperage

1958 strength of electrical current.

1959 **DEPRECATED** in *Version 1.6*. Replaced by AMPERAGE_AC and AMPERAGE_DC.

1960 The units of Amperage **MUST** be AMPERE.

1961 5.3.3.1 Subtypes of Amperage

- 1962 • ACTUAL
- 1963 measured or reported value of an observation.
- 1964 **DEPRECATED** in *Version 1.6*.
- 1965 • ALTERNATING
- 1966 measurement of alternating voltage or current.
- 1967 If not specified further in statistic, defaults to RMS voltage.
- 1968 **DEPRECATED** in *Version 1.6*.
- 1969 • DIRECT
- 1970 measurement of DC current or voltage.
- 1971 **DEPRECATED** in *Version 1.6*.
- 1972 • TARGET
- 1973 goal of the operation or process.
- 1974 **DEPRECATED** in *Version 1.6*.

1975 5.3.4 AmperageAC

- 1976 electrical current that reverses direction at regular short intervals.
- 1977 The units of AmperageAC **MUST** be AMPERE.
- 1978 A subType **MUST** always be specified.

1979 5.3.4.1 Subtypes of AmperageAC

- 1980 • ACTUAL
- 1981 measured or reported value of an observation.
- 1982 • COMMANDED
- 1983 directive value including adjustments such as an offset or overrides.
- 1984 • PROGRAMMED
- 1985 directive value without offsets and adjustments.

1986 **5.3.5 AmperageDC**

1987 electric current flowing in one direction only.

1988 The units of AmperageDC **MUST** be AMPERE.

1989 A subType **MUST** always be specified.

1990 **5.3.5.1 Subtypes of AmperageDC**

1991 • ACTUAL

1992 measured or reported value of an observation.

1993 • COMMANDED

1994 directive value including adjustments such as an offset or overrides.

1995 • PROGRAMMED

1996 directive value without offsets and adjustments.

1997 **5.3.6 Angle**

1998 angular position.

1999 The units of Angle **MUST** be DEGREE.

2000 **5.3.6.1 Subtypes of Angle**

2001 • ACTUAL

2002 measured or reported value of an observation.

2003 • COMMANDED

2004 directive value including adjustments such as an offset or overrides.

2005 **5.3.7 AngularAcceleration**

2006 positive rate of change of angular velocity.

2007 The units of AngularAcceleration **MUST** be DEGREE/SECOND².

2008 The default subType of AngularAcceleration is ACTUAL.

2009 **5.3.7.1 Subtypes of AngularAcceleration**

2010 • ACTUAL

2011 measured or reported value of an observation.

2012 • COMMANDED

2013 directive value including adjustments such as an offset or overrides.

2014 • PROGRAMMED

2015 directive value without offsets and adjustments.

2016 **5.3.8 AngularDeceleration**

2017 negative rate of change of angular velocity.

2018 The units of AngularDeceleration **MUST** be DEGREE/SECOND².

2019 The default subType of AngularDeceleration is ACTUAL.

2020 **5.3.8.1 Subtypes of AngularDeceleration**

2021 • ACTUAL

2022 measured or reported value of an observation.

2023 • COMMANDED

2024 directive value including adjustments such as an offset or overrides.

2025 • PROGRAMMED

2026 directive value without offsets and adjustments.

2027 **5.3.9 AngularVelocity**

2028 rate of change of angular position.

2029 The units of AngularVelocity **MUST** be DEGREE/SECOND.

2030 **5.3.10 AssetUpdateRate**

2031 average rate of change of values for assets in the MTConnect streams.

2032 The average is computed over a rolling window defined by the implementation.

2033 The units of AssetUpdateRate **MUST** be COUNT/SECOND.

2034 **5.3.11 AxisFeedrate**

2035 feedrate of a linear axis.

2036 The units of AxisFeedrate **MUST** be MILLIMETER/SECOND.

2037 **5.3.11.1 Subtypes of AxisFeedrate**

- 2038 • ACTUAL
- 2039 measured or reported value of an observation.
- 2040 • COMMANDED
- 2041 directive value including adjustments such as an offset or overrides.
- 2042 • JOG
- 2043 relating to momentary activation of a function or a movement.
- 2044 **DEPRECATION WARNING:** May be deprecated in the future.
- 2045 • OVERRIDE
- 2046 operator's overridden value.
- 2047 • PROGRAMMED
- 2048 directive value without offsets and adjustments.

- 2049 • `RAPID`
- 2050 performing an operation faster or in less time than nominal rate.

2051 **5.3.12 CapacityFluid**

- 2052 fluid capacity of an object or container.
- 2053 The units of `CapacityFluid` **MUST** be `MILLILITER`.

2054 **5.3.13 CapacitySpatial**

- 2055 geometric capacity of an object or container.
- 2056 The units of `CapacitySpatial` **MUST** be `CUBIC_MILLIMETER`.

2057 **5.3.14 Concentration**

- 2058 percentage of one component within a mixture of components.
- 2059 The units of `Concentration` **MUST** be `PERCENT`.

2060 **5.3.15 Conductivity**

- 2061 ability of a material to conduct electricity.
- 2062 The units of `Conductivity` **MUST** be `SIEMENS/METER`.

2063 **5.3.16 CuttingSpeed**

- 2064 speed difference (relative velocity) between the cutting mechanism and the surface of the
- 2065 workpiece it is operating on.
- 2066 The units of `CuttingSpeed` **MUST** be `MILLIMETER/SECOND`.

2067 **5.3.16.1 Subtypes of CuttingSpeed**

- 2068 • ACTUAL
- 2069 measured or reported value of an observation.
- 2070 • COMMANDED
- 2071 directive value including adjustments such as an offset or overrides.
- 2072 • PROGRAMMED
- 2073 directive value without offsets and adjustments.

2074 **5.3.17 Deceleration**

- 2075 negative rate of change of velocity.
- 2076 The units of Deceleration **MUST** be MILLIMETER/SECOND².
- 2077 The default subType of Deceleration is ACTUAL.

2078 **5.3.17.1 Subtypes of Deceleration**

- 2079 • ACTUAL
- 2080 measured or reported value of an observation.
- 2081 • COMMANDED
- 2082 directive value including adjustments such as an offset or overrides.
- 2083 • PROGRAMMED
- 2084 directive value without offsets and adjustments.

2085 **5.3.18 Density**

- 2086 volumetric mass of a material per unit volume of that material.
- 2087 The units of Density **MUST** be MILLIGRAM/CUBIC_MILLIMETER.

2088 **5.3.19 DepositionAccelerationVolumetric**

2089 rate of change in spatial volume of material deposited in an additive manufacturing pro-
2090 cess.

2091 The units of DepositionAccelerationVolumetric **MUST** be CUBIC_MILLIMETER/SECOND²

2092 **5.3.19.1 Subtypes of DepositionAccelerationVolumetric**

- 2093 • ACTUAL
2094 measured or reported value of an observation.
- 2095 • COMMANDED
2096 directive value including adjustments such as an offset or overrides.

2097 **5.3.20 DepositionDensity**

2098 density of the material deposited in an additive manufacturing process per unit of volume.

2099 The units of DepositionDensity **MUST** be MILLIGRAM/CUBIC_MILLIMETER.

2100 **5.3.20.1 Subtypes of DepositionDensity**

- 2101 • ACTUAL
2102 measured or reported value of an observation.
- 2103 • COMMANDED
2104 directive value including adjustments such as an offset or overrides.

2105 **5.3.21 DepositionMass**

2106 mass of the material deposited in an additive manufacturing process.

2107 The units of DepositionMass **MUST** be MILLIGRAM.

2108 **5.3.21.1 Subtypes of DepositionMass**

- 2109 • ACTUAL
- 2110 measured or reported value of an observation.
- 2111 • COMMANDED
- 2112 directive value including adjustments such as an offset or overrides.

2113 **5.3.22 DepositionRateVolumetric**

2114 rate at which a spatial volume of material is deposited in an additive manufacturing pro-
2115 cess.

2116 The units of DepositionRateVolumetric **MUST** be CUBIC_MILLIMETER/SECOND.

2117 **5.3.22.1 Subtypes of DepositionRateVolumetric**

- 2118 • ACTUAL
- 2119 measured or reported value of an observation.
- 2120 • COMMANDED
- 2121 directive value including adjustments such as an offset or overrides.

2122 **5.3.23 DepositionVolume**

2123 spatial volume of material to be deposited in an additive manufacturing process.

2124 The units of DepositionVolume **MUST** be CUBIC_MILLIMETER.

2125 **5.3.23.1 Subtypes of DepositionVolume**

- 2126 • ACTUAL
- 2127 measured or reported value of an observation.
- 2128 • COMMANDED
- 2129 directive value including adjustments such as an offset or overrides.

2130 5.3.24 Diameter

2131 dimension of a diameter.

2132 The units of Diameter **MUST** be MILLIMETER.

2133 5.3.25 Displacement

2134 change in position of an object.

2135 The units of Displacement **MUST** be MILLIMETER.

2136 5.3.26 ElectricalEnergy

2137 Wattage used or generated by a component over an interval of time.

2138 The units of ElectricalEnergy **MUST** be WATT_SECOND.

2139 5.3.27 EquipmentTimer

2140 amount of time a piece of equipment or a sub-part of a piece of equipment has performed
2141 specific activities.

2142 The units of EquipmentTimer **MUST** be SECOND.

2143 A subType **MUST** always be specified.

2144 5.3.27.1 Subtypes of EquipmentTimer

2145 • DELAY

2146 elapsed time of a temporary halt of action.

2147 • LOADED

2148 time that the sub-parts of a piece of equipment are under load.

2149 Example: For traditional machine tools, this is a measurement of the time that the
2150 cutting tool is assumed to be engaged with the part.

- 2151 • OPERATING
- 2152 time that the major sub-parts of a piece of equipment are powered or performing any
- 2153 activity whether producing a part or product or not.
- 2154 Example: For traditional machine tools, this includes WORKING, plus idle time.
- 2155 • POWERED
- 2156 time that primary power is applied to the piece of equipment and, as a minimum, the
- 2157 controller or logic portion of the piece of equipment is powered and functioning or
- 2158 components that are required to remain on are powered.
- 2159 Example: Heaters for an extrusion machine that are required to be powered even
- 2160 when the equipment is turned off.
- 2161 • WORKING
- 2162 time that a piece of equipment is performing any activity the equipment is active and
- 2163 performing a function under load or not.
- 2164 Example: For traditional machine tools, this includes LOADED, plus rapid moves,
- 2165 tool changes, etc.

2166 **5.3.28 FillLevel**

- 2167 amount of a substance remaining compared to the planned maximum amount of that sub-
- 2168 stance.
- 2169 The units of FillLevel **MUST** be PERCENT.

2170 **5.3.29 Flow**

- 2171 rate of flow of a fluid.
- 2172 The units of Flow **MUST** be LITER/SECOND.

2173 **5.3.30 Frequency**

- 2174 number of occurrences of a repeating event per unit time.
- 2175 The units of Frequency **MUST** be HERTZ.

2176 **5.3.31 <<deprecated>>GlobalPosition**

2177 position in three-dimensional space.

2178 **DEPRECATED** in Version 1.1.

2179 The units of GlobalPosition **MUST** be MILLIMETER.

2180 **5.3.31.1 Subtypes of GlobalPosition**

2181 • ACTUAL

2182 measured or reported value of an observation.

2183 • COMMANDED

2184 directive value including adjustments such as an offset or overrides.

2185 **5.3.32 HumidityAbsolute**

2186 amount of water vapor expressed in grams per cubic meter.

2187 The units of HumidityAbsolute **MUST** be GRAM/CUBIC_METER.

2188 **5.3.32.1 Subtypes of HumidityAbsolute**

2189 • ACTUAL

2190 measured or reported value of an observation.

2191 • COMMANDED

2192 directive value including adjustments such as an offset or overrides.

2193 **5.3.33 HumidityRelative**

2194 amount of water vapor present expressed as a percent to reach saturation at the same tem-
2195 perature.

2196 The units of HumidityRelative **MUST** be PERCENT.

2197 **5.3.33.1 Subtypes of HumidityRelative**

- 2198 • ACTUAL
- 2199 measured or reported value of an observation.
- 2200 • COMMANDED
- 2201 directive value including adjustments such as an offset or overrides.

2202 **5.3.34 HumiditySpecific**

2203 ratio of the water vapor present over the total weight of the water vapor and air present
2204 expressed as a percent.

2205 The units of HumiditySpecific **MUST** be PERCENT.

2206 **5.3.34.1 Subtypes of HumiditySpecific**

- 2207 • ACTUAL
- 2208 measured or reported value of an observation.
- 2209 • COMMANDED
- 2210 directive value including adjustments such as an offset or overrides.

2211 **5.3.35 Length**

2212 length of an object.

2213 The units of Length **MUST** be MILLIMETER.

2214 **5.3.35.1 Subtypes of Length**

- 2215 • REMAINING
- 2216 remaining total length of an object.

- 2217 • STANDARD
- 2218 standard or original length of an object.
- 2219 • USEABLE
- 2220 remaining usable length of an object.

2221 **5.3.36 <<deprecated>>Level**

- 2222 level of a resource.
- 2223 **DEPRECATED** in *Version 1.2*. See `FILL_LEVEL`.
- 2224 The `units` of `Level` **MUST** be `PERCENT`.

2225 **5.3.37 LinearForce**

- 2226 *force* applied to a mass in one direction only.
- 2227 The `units` of `LinearForce` **MUST** be `NEWTON`.

2228 **5.3.38 Load**

- 2229 actual versus the standard rating of a piece of equipment.
- 2230 The `units` of `Load` **MUST** be `PERCENT`.

2231 **5.3.39 Mass**

- 2232 mass of an object(s) or an amount of material.
- 2233 The `units` of `Mass` **MUST** be `KILOGRAM`.

2234 **5.3.40 ObservationUpdateRate**

2235 average rate of change of values for data items in the MTConnect streams. The average is
2236 computed over a rolling window defined by the implementation.

2237 The units of ObservationUpdateRate **MUST** be COUNT/SECOND.

2238 **5.3.41 Openness**

2239 percentage open where 100% is fully open and 0% is fully closed.

2240 The units of Openness **MUST** be PERCENT.

2241 **5.3.42 Orientation**

2242 measured or calculated orientation of a plane or vector relative to a cartesian coordinate
2243 system.

2244 The units of Orientation **MUST** be DEGREE_3D.

2245 **5.3.42.1 Subtypes of Orientation**

2246 • ACTUAL

2247 measured or reported value of an observation.

2248 • COMMANDED

2249 directive value including adjustments such as an offset or overrides.

2250 **5.3.43 PH**

2251 acidity or alkalinity of a solution.

2252 The units of PH **MUST** be PH.

2253 5.3.44 PathFeedrate

2254 feedrate for the axes, or a single axis, associated with a `Path` component.

2255 The units of `PathFeedrate` **MUST** be `MILLIMETER/SECOND`.

2256 5.3.44.1 Subtypes of PathFeedrate

2257 • `ACTUAL`

2258 measured or reported value of an observation.

2259 • `COMMANDED`

2260 directive value including adjustments such as an offset or overrides.

2261 • `JOG`

2262 relating to momentary activation of a function or a movement.

2263 **DEPRECATION WARNING:** May be deprecated in the future.

2264 • `OVERRIDE`

2265 operator's overridden value.

2266 **DEPRECATED** in *Version 1.3*.

2267 • `PROGRAMMED`

2268 directive value without offsets and adjustments.

2269 • `RAPID`

2270 performing an operation faster or in less time than nominal rate.

2271 5.3.45 PathFeedratePerRevolution

2272 feedrate for the axes, or a single axis.

2273 The units of `PathFeedratePerRevolution` **MUST** be `MILLIMETER/REVO-`
2274 `LUTION`.

2275 **5.3.45.1 Subtypes of PathFeedratePerRevolution**

- 2276 • ACTUAL
- 2277 measured or reported value of an observation.
- 2278 • COMMANDED
- 2279 directive value including adjustments such as an offset or overrides.
- 2280 • PROGRAMMED
- 2281 directive value without offsets and adjustments.

2282 **5.3.46 PathPosition**

- 2283 measured or calculated position of a control point associated with a Controller entity,
 2284 or Path entity if provided, of a piece of equipment.
- 2285 The units of PathPosition **MUST** be MILLIMETER_3D.

2286 **5.3.46.1 Subtypes of PathPosition**

- 2287 • ACTUAL
- 2288 measured or reported value of an observation.
- 2289 • COMMANDED
- 2290 directive value including adjustments such as an offset or overrides.
- 2291 • PROBE
- 2292 position provided by a measurement probe.
- 2293 **DEPRECATION WARNING:** May be deprecated in the future.
- 2294 • TARGET
- 2295 goal of the operation or process.

2296 5.3.47 Position

2297 measured or calculated position of a Component element as reported by a piece of equip-
2298 ment.

2299 The units of Position **MUST** be MILLIMETER.

2300 5.3.47.1 Subtypes of Position

2301 • ACTUAL

2302 measured or reported value of an observation.

2303 • COMMANDED

2304 directive value including adjustments such as an offset or overrides.

2305 • PROGRAMMED

2306 directive value without offsets and adjustments.

2307 • TARGET

2308 goal of the operation or process.

2309 5.3.48 PowerFactor

2310 ratio of real power flowing to a load to the apparent power in that AC circuit.

2311 The units of PowerFactor **MUST** be PERCENT.

2312 5.3.49 Pressure

2313 force per unit area measured relative to atmospheric pressure.

2314 Commonly referred to as gauge pressure.

2315 The units of Pressure **MUST** be PASCAL.

2316 **5.3.50 PressureAbsolute**

2317 The force per unit area measured relative to a vacuum.

2318 The units of PressureAbsolute **MUST** be PASCAL.

2319 **5.3.51 PressurizationRate**

2320 change of pressure per unit time.

2321 The units of PressurizationRate **MUST** be PASCAL/SECOND.

2322 **5.3.51.1 Subtypes of PressurizationRate**

- 2323 • ACTUAL
- 2324 measured or reported value of an observation.
- 2325 • COMMANDED
- 2326 directive value including adjustments such as an offset or overrides.
- 2327 • PROGRAMMED
- 2328 directive value without offsets and adjustments.

2329 **5.3.52 ProcessTimer**

2330 amount of time a piece of equipment has performed different types of activities associated
2331 with the process being performed at that piece of equipment.

2332 The units of ProcessTimer **MUST** be SECOND.

2333 A subType **MUST** always be specified.

2334 **5.3.52.1 Subtypes of ProcessTimer**

- 2335 • DELAY
- 2336 elapsed time of a temporary halt of action.

- 2337 • PROCESS
- 2338 time from the beginning of production of a part or product on a piece of equipment
- 2339 until the time that production is complete for that part or product on that piece of
- 2340 equipment.
- 2341 This includes the time that the piece of equipment is running, producing parts or
- 2342 products, or in the process of producing parts.

2343 **5.3.53 Resistance**

- 2344 degree to which a substance opposes the passage of an electric current.
- 2345 The units of Resistance **MUST** be OHM.

2346 **5.3.54 RotaryVelocity**

- 2347 rotational speed of a rotary axis.
- 2348 The units of RotaryVelocity **MUST** be REVOLUTION/MINUTE.

2349 **5.3.54.1 Subtypes of RotaryVelocity**

- 2350 • ACTUAL
- 2351 measured or reported value of an observation.
- 2352 • COMMANDED
- 2353 directive value including adjustments such as an offset or overrides.
- 2354 • OVERRIDE
- 2355 The operators overridden value.
- 2356 • PROGRAMMED
- 2357 directive value without offsets and adjustments.

2358 **5.3.55 SoundLevel**

2359 sound level or sound pressure level relative to atmospheric pressure.

2360 The units of SoundLevel **MUST** be DECIBEL.

2361 The default subType of SoundLevel is A_SCALE.

2362 **5.3.55.1 Subtypes of SoundLevel**

2363 • A_SCALE

2364 A Scale weighting factor. This is the default weighting factor if no factor is specified

2365 • B_SCALE

2366 B Scale weighting factor

2367 • C_SCALE

2368 C Scale weighting factor

2369 • D_SCALE

2370 D Scale weighting factor

2371 • NO_SCALE

2372 No weighting factor on the frequency scale

2373 **5.3.56 <<deprecated>>SpindleSpeed**

2374 rotational speed of the rotary axis.

2375 **DEPRECATED** in *Version 1.2*. Replaced by ROTARY_VELOCITY.

2376 The units of SpindleSpeed **MUST** be REVOLUTION/MINUTE.

2377 **5.3.56.1 Subtypes of SpindleSpeed**

2378 • ACTUAL

2379 measured or reported value of an observation.

2380 **DEPRECATED** in *Version 1.3*.

- 2381 • COMMANDED
- 2382 directive value including adjustments such as an offset or overrides.
- 2383 **DEPRECATED** in *Version 1.3*.
- 2384 • OVERRIDE
- 2385 operator's overridden value.
- 2386 **DEPRECATED** in *Version 1.3*.

2387 **5.3.57 Strain**

- 2388 amount of deformation per unit length of an object when a load is applied.
- 2389 The units of Strain **MUST** be PERCENT.

2390 **5.3.58 Temperature**

- 2391 degree of hotness or coldness measured on a definite scale.
- 2392 The units of Temperature **MUST** be CELSIUS.

2393 **5.3.59 Tension**

- 2394 force that stretches or elongates an object.
- 2395 The units of Tension **MUST** be NEWTON.

2396 **5.3.60 Tilt**

- 2397 angular displacement.
- 2398 The units of Tilt **MUST** be MICRO_RADIAN.

2399 5.3.61 Torque

2400 turning force exerted on an object or by an object.

2401 The units of Torque **MUST** be NEWTON_METER.

2402 5.3.62 Velocity

2403 rate of change of position of a Component.

2404 The units of Velocity **MUST** be MILLIMETER/SECOND.

2405 5.3.63 Viscosity

2406 fluid's resistance to flow.

2407 The units of Viscosity **MUST** be PASCAL_SECOND.

2408 5.3.64 VoltAmpere

2409 apparent power in an electrical circuit, equal to the product of root-mean-square (RMS)
2410 voltage and RMS current (commonly referred to as VA).

2411 The units of VoltAmpere **MUST** be VOLT_AMPERE.

2412 5.3.65 VoltAmpereReactive

2413 reactive power in an AC electrical circuit (commonly referred to as VAR).

2414 The units of VoltAmpereReactive **MUST** be VOLT_AMPERE_REACTIVE.

2415 5.3.66 <<deprecated>>Voltage

2416 electrical potential between two points.

2417 **DEPRECATED** in *Version 1.6*. Replaced by VOLTAGE_AC and VOLTAGE_DC.

2418 The units of Voltage **MUST** be VOLT.

2419 **5.3.66.1 Subtypes of Voltage**

2420 • ACTUAL

2421 measured or reported value of an observation.

2422 **DEPRECATED** in *Version 1.6*.

2423 • ALTERNATING

2424 alternating voltage or current.

2425 If not specified further in statistic, defaults to RMS voltage.

2426 **DEPRECATED** in *Version 1.6*.

2427 • DIRECT

2428 DC current or voltage.

2429 **DEPRECATED** in *Version 1.6*.

2430 • TARGET

2431 goal of the operation or process.

2432 **DEPRECATED** in *Version 1.6*.

2433 **5.3.67 VoltageAC**

2434 electrical potential between two points in an electrical circuit in which the current period-
2435 ically reverses direction.

2436 The units of VoltageAC **MUST** be VOLT.

2437 A subType **MUST** always be specified.

2438 **5.3.67.1 Subtypes of VoltageAC**

2439 • ACTUAL

2440 measured or reported value of an observation.

- 2441 • COMMANDED
- 2442 directive value including adjustments such as an offset or overrides.
- 2443 • PROGRAMMED
- 2444 directive value without offsets and adjustments.

2445 **5.3.68 VoltageDC**

- 2446 electrical potential between two points in an electrical circuit in which the current is uni-
- 2447 directional.
- 2448 The units of VoltageDC **MUST** be VOLT.
- 2449 A subType **MUST** always be specified.

2450 **5.3.68.1 Subtypes of VoltageDC**

- 2451 • ACTUAL
- 2452 measured or reported value of an observation.
- 2453 • COMMANDED
- 2454 directive value including adjustments such as an offset or overrides.
- 2455 • PROGRAMMED
- 2456 directive value without offsets and adjustments.

2457 **5.3.69 VolumeFluid**

- 2458 fluid volume of an object or container.
- 2459 The units of VolumeFluid **MUST** be MILLILITER.

2460 **5.3.69.1 Subtypes of VolumeFluid**

- 2461 • ACTUAL
- 2462 measured or reported value of an observation.

- 2463 • CONSUMED
- 2464 reported or measured value of the amount used in the manufacturing process.
- 2465 • ENDED
- 2466 boundary when an activity or an event terminates.
- 2467 • PART
- 2468 reported or measured value of amount included in the *part*.
- 2469 • START
- 2470 boundary when an activity or an event commences.
- 2471 • WASTE
- 2472 reported or measured value of the amount discarded.

2473 **5.3.70 VolumeSpatial**

- 2474 geometric volume of an object or container.
- 2475 The units of VolumeSpatial **MUST** be CUBIC_MILLIMETER.

2476 **5.3.70.1 Subtypes of VolumeSpatial**

- 2477 • ACTUAL
- 2478 measured or reported value of an observation.
- 2479 • CONSUMED
- 2480 reported or measured value of the amount used in the manufacturing process.
- 2481 • ENDED
- 2482 boundary when an activity or an event terminates.
- 2483 • PART
- 2484 reported or measured value of amount included in the *part*.
- 2485 • START
- 2486 boundary when an activity or an event commences.
- 2487 • WASTE
- 2488 reported or measured value of the amount discarded

2489 **5.3.71 Wattage**

2490 power flowing through or dissipated by an electrical circuit or piece of equipment.

2491 The units of Wattage **MUST** be WATT.

2492 **5.3.71.1 Subtypes of Wattage**

- 2493 • ACTUAL
- 2494 measured or reported value of an observation.
- 2495 • TARGET
- 2496 goal of the operation or process.

2497 **5.3.72 XDimension**

2498 dimension of an entity relative to the X direction of the referenced coordinate system.

2499 The units of XDimension **MUST** be MILLIMETER.

2500 **5.3.73 YDimension**

2501 dimension of an entity relative to the Y direction of the referenced coordinate system.

2502 The units of YDimension **MUST** be MILLIMETER.

2503 **5.3.74 ZDimension**

2504 dimension of an entity relative to the Z direction of the referenced coordinate system.

2505 The units of ZDimension **MUST** be MILLIMETER.

2506 **6 Profile**

2507 MTConnect Profile is a *profile* that extends the Systems Modeling Language (SysML)
2508 metamodel for the MTConnect domain using additional data types and *stereotypes*.

2509 6.1 DataTypes

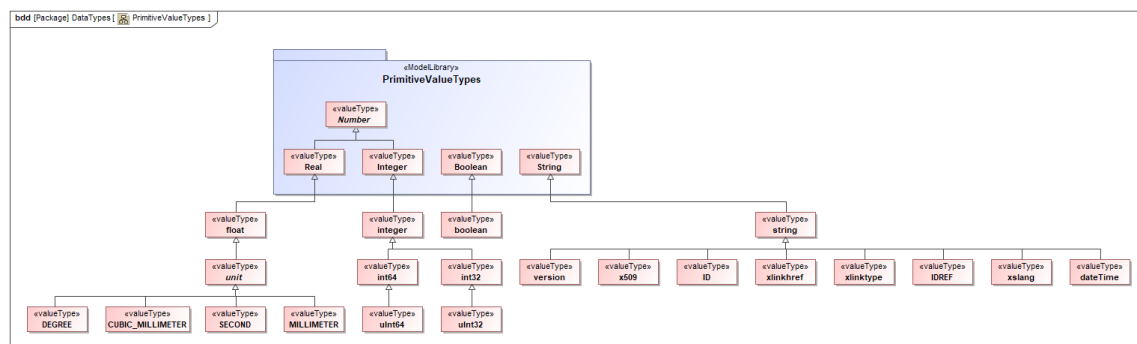


Figure 13: DataTypes

2510 6.1.1 boolean

```

2511 primitive type.

```

2512 **6.1.2 ID**

2513 string that represents an identifier (ID).

2514 **6.1.3 string**

2515 primitive type.

2516 **6.1.4 float**

2517 primitive type.

2518 6.1.5 dateTime

2519 string that represents timestamp in ISO 8601 format.

2520 6.1.6 integer

2521 primitive type.

2522 6.1.7 xlinktype

2523 string that represents the type of an XLink element. See [https://www.w3.org/TR/](https://www.w3.org/TR/xlink11/)
2524 [xlink11/](https://www.w3.org/TR/xlink11/).

2525 6.1.8 xslang

2526 string that represents a language tag. See [http://www.ietf.org/rfc/rfc4646.](http://www.ietf.org/rfc/rfc4646.txt)
2527 [txt](http://www.ietf.org/rfc/rfc4646.txt).

2528 6.1.9 SECOND

2529 float that represents time in seconds.

2530 6.1.10 IDREF

2531 string that represents a reference to an ID.

2532 6.1.11 xlinkhref

2533 string that represents the locator attribute of an XLink element. See [https://www.w3.](https://www.w3.org/TR/xlink11/)
2534 [org/TR/xlink11/](https://www.w3.org/TR/xlink11/).

2535 6.1.12 x509

2536 string that represents an `x509` data block. *Ref ISO/IEC 9594-8:2020.*

2537 6.1.13 int32

2538 32-bit integer.

2539 6.1.14 int64

2540 64-bit integer.

2541 6.1.15 version

2542 series of four numeric values, separated by a decimal point, representing a *major*, *minor*,
2543 and *revision* number of the MTConnect Standard and the revision number of a specific
2544 *schema*.

2545 6.1.16 uInt32

2546 32-bit unsigned integer.

2547 6.1.17 uInt64

2548 64-bit unsigned integer.

2549 6.2 Stereotypes

2550 6.2.1 organizer

2551 element that *organizes* other elements of a type.

2552 6.2.2 deprecated

2553 element that has been deprecated.

2554 6.2.3 extensible

2555 enumeration that can be extended.

2556 6.2.4 informative

2557 element that is descriptive and non-normative.

2558 6.2.5 valueType

2559 extends SysML <<ValueType>> to include `Class` as a value type.

2560 6.2.6 normative

2561 element that has been added to the standard.

2562 6.2.7 observes

2563 association in which a *Component* makes *Observations* about an observable *DataItem*.

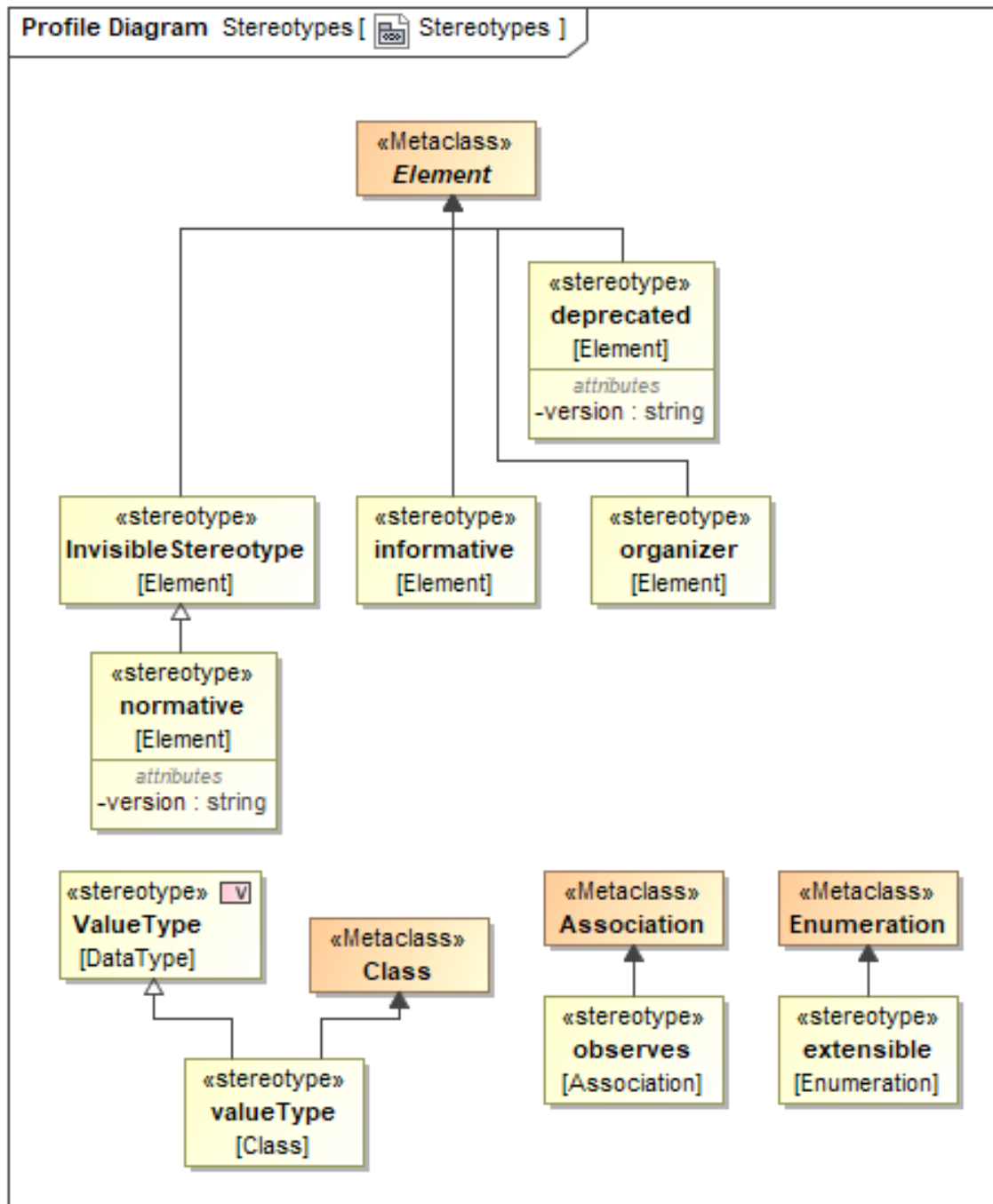


Figure 14: Stereotypes

2564 Appendices

2565 A Bibliography

2566 Engineering Industries Association. EIA Standard - EIA-274-D, Interchangeable Variable,
2567 Block Data Format for Positioning, Contouring, and Contouring/Positioning Numerically
2568 Controlled Machines. Washington, D.C. 1979.

2569 ISO TC 184/SC4/WG3 N1089. ISO/DIS 10303-238: Industrial automation systems and
2570 integration Product data representation and exchange Part 238: Application Protocols: Ap-
2571 plication interpreted model for computerized numerical controllers. Geneva, Switzerland,
2572 2004.

2573 International Organization for Standardization. ISO 14649: Industrial automation sys-
2574 tems and integration – Physical device control – Data model for computerized numerical
2575 controllers – Part 10: General process data. Geneva, Switzerland, 2004.

2576 International Organization for Standardization. ISO 14649: Industrial automation sys-
2577 tems and integration – Physical device control – Data model for computerized numerical
2578 controllers – Part 11: Process data for milling. Geneva, Switzerland, 2000.

2579 International Organization for Standardization. ISO 6983/1 – Numerical Control of ma-
2580 chines – Program format and definition of address words – Part 1: Data format for posi-
2581 tioning, line and contouring control systems. Geneva, Switzerland, 1982.

2582 Electronic Industries Association. ANSI/EIA-494-B-1992, 32 Bit Binary CL (BCL) and
2583 7 Bit ASCII CL (ACL) Exchange Input Format for Numerically Controlled Machines.
2584 Washington, D.C. 1992.

2585 National Aerospace Standard. Uniform Cutting Tests - NAS Series: Metal Cutting Equip-
2586 ment Specifications. Washington, D.C. 1969.

2587 International Organization for Standardization. ISO 10303-11: 1994, Industrial automa-
2588 tion systems and integration Product data representation and exchange Part 11: Descrip-
2589 tion methods: The EXPRESS language reference manual. Geneva, Switzerland, 1994.

2590 International Organization for Standardization. ISO 10303-21: 1996, Industrial automa-
2591 tion systems and integration – Product data representation and exchange – Part 21: Imple-
2592 mentation methods: Clear text encoding of the exchange structure. Geneva, Switzerland,
2593 1996.

2594 H.L. Horton, F.D. Jones, and E. Oberg. Machinery's Handbook. Industrial Press, Inc.

- 2595 New York, 1984.
- 2596 International Organization for Standardization. ISO 841-2001: Industrial automation sys-
2597 tems and integration - Numerical control of machines - Coordinate systems and motion
2598 nomenclature. Geneva, Switzerland, 2001.
- 2599 ASME B5.57: Methods for Performance Evaluation of Computer Numerically Controlled
2600 Lathes and Turning Centers, 1998.
- 2601 ASME/ANSI B5.54: Methods for Performance Evaluation of Computer Numerically Con-
2602 trolled Machining Centers. 2005.
- 2603 OPC Foundation. OPC Unified Architecture Specification, Part 1: Concepts Version 1.00.
2604 July 28, 2006.
- 2605 IEEE STD 1451.0-2007, Standard for a Smart Transducer Interface for Sensors and Ac-
2606 tuators – Common Functions, Communication Protocols, and Transducer Electronic Data
2607 Sheet (TEDS) Formats, IEEE Instrumentation and Measurement Society, TC-9, The In-
2608 stitute of Electrical and Electronics Engineers, Inc., New York, N.Y. 10016, SH99684,
2609 October 5, 2007.
- 2610 IEEE STD 1451.4-1994, Standard for a Smart Transducer Interface for Sensors and Ac-
2611 tuators – Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet
2612 (TEDS) Formats, IEEE Instrumentation and Measurement Society, TC-9, The Institute of
2613 Electrical and Electronics Engineers, Inc., New York, N.Y. 10016, SH95225, December
2614 15, 2004.

2615 B XML Schema Diagrams

2616 B.1 Observations Schema Diagrams

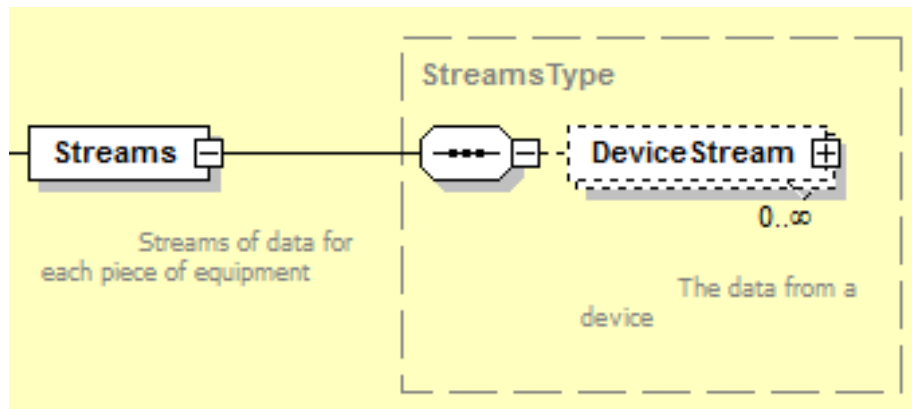


Figure 15: Streams Schema

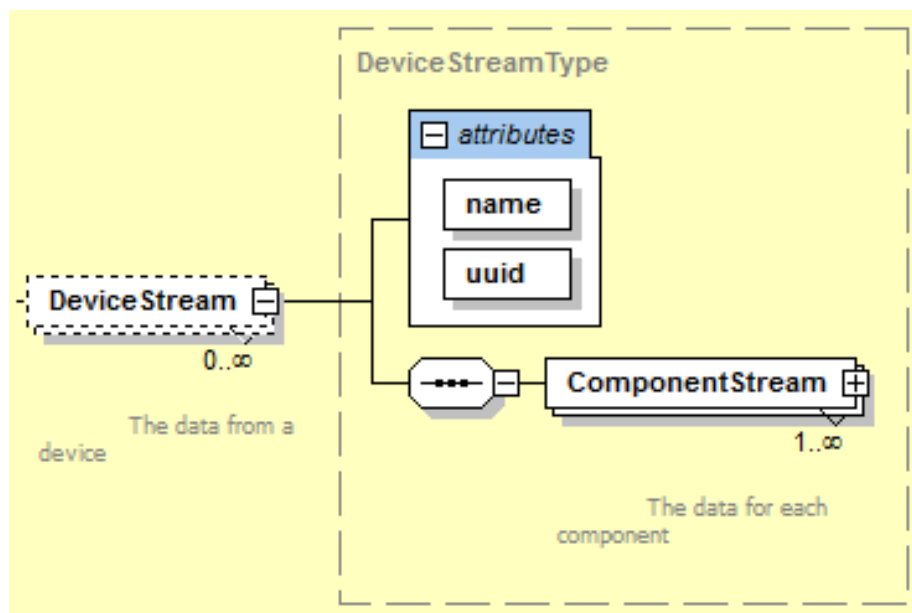


Figure 16: DeviceStream Schema

2617 B.2 Representation Schema Diagrams

2618 C XML Examples

2619 C.1 DeviceStream Example

Example 1: DeviceStream Example

```

2620 1 <Streams>
2621 2   <DeviceStream name="M12346" uuid="M8010W4194N">
2622 3     <ComponentStream component="Device" name="M12346" componentId="
2623 4       d1">
2624 5       <Events>
2625 6         <Availability dataItemId="avail" sequence="156" timestamp="
2626 7         2021-10-01T14:26:38.668505Z">AVAILABLE</Availability>
2627 8         <AssetChanged assetType="CuttingTool" dataItemId="d1\
2628 9         textunderscore_asset\textunderscore_chg" sequence="75570"
2629 10        timestamp="2021-10-07T05:08:53.870206Z">M8010W4194N1
2630 11        .120</AssetChanged>
2631 12        <AssetRemoved assetType="CuttingTool" dataItemId="d1\
2632 13        textunderscore_asset\textunderscore_rem" sequence="140"
2633 14        timestamp="2021-10-01T11:40:08.101461Z">UNAVAILABLE</
2634 15        AssetRemoved>
2635 16      </Events>
2636 17    </ComponentStream>
2637 18    <ComponentStream component="Controller" name="controller"
2638 19    componentId="cont">
2639 20      <Events>
2640 21        <EmergencyStop dataItemId="estop" sequence="159" timestamp="
2641 22        2021-10-01T14:26:38.66869Z">ARMED</EmergencyStop>
2642 23      </Events>
2643 24      <Samples>
2644 25        <AccumulatedTime dataItemId="cut\textunderscore_time"
2645 26        sequence="75437" timestamp="2021-10-07T05:08:28.221704Z">
2646 27        1763070.0</AccumulatedTime>
2647 28      </Samples>
2648 29      <Condition>
2649 30        <Unavailable dataItemId="cont\textunderscore_system"
2650 31        sequence="72" timestamp="2021-10-11T21:04:03.251999Z" type="
2651 32        SYSTEM"/>
2652 33        <Warning dataItemId="cont\textunderscore_system" nativecode=
2653 34        "313" nativeSeverity="50" sequence="75573" timestamp="
2654 35        2021-10-07T05:08:58.518317Z" type="LOGIC\textunderscore_
2655 36        PROGRAM">PALLET ARM DOWN RS. MALF.</Warning>
2656 37      </Condition>
2657 38    </ComponentStream>
2658 39    <ComponentStream component="Path" name="path" componentId="path1
2659 40    ">
2660 41      <Events>
2661 42        <Execution dataItemId="execution" name="execution" sequence=
2662 43        "222623" timestamp="2021-10-12T06:04:32.761198Z">INTERRUPTED</
2663 44        Execution>
2664 45        <VariableDataSet count="2" dataItemId="cvars" sequence="
2665 46        126513" timestamp="2021-10-12T03:57:31.106559Z">
2666 47        <Entry key="100">66.3314</Entry>

```

```

2667 27      <Entry key="101">167.2</Entry>
2668 28      </VariableDataSet>
2669 29      <WorkOffsetTable count="2" dataItemId="woffset" sequence="
2670      222101" timestamp="2021-10-12T06:04:11.990531Z">
2671 30      <Entry key="G53.1">
2672 31      <Cell key="X">1</Cell>
2673 32      <Cell key="Y">2</Cell>
2674 33      <Cell key="Z">3</Cell>
2675 34      </Entry>
2676 35      <Entry key="G53.2">
2677 36      <Cell key="X">4</Cell>
2678 37      <Cell key="Y">5</Cell>
2679 38      <Cell key="Z">6</Cell>
2680 39      </Entry>
2681 40      </WorkOffsetTable>
2682 41      </Events>
2683 42      <Samples>
2684 43      <CuttingSpeed dataItemId="cspeed" sequence="112" timestamp="
2685      2021-10-07T05:08:28.221704Z" subType="ACTUAL">UNAVAILABLE</
2686      CuttingSpeed>
2687 44      </Samples>
2688 45      <Condition>
2689 46      <Normal dataItemId="path\textunderscore_system" sequence="
2690      153" timestamp="2021-10-11T21:04:03.262845Z" type="SYSTEM"/>
2691 47      </Condition>
2692 48      </ComponentStream>
2693 49      </DeviceStream>
2694 50      </Streams>

```

2695 C.2 Observations made for DataItem Example

Example 2: MTConnectDevices Response Document

```

2696 1 <Components>
2697 2   <Controller id="cont" name="controller">
2698 3     <DataItems>
2699 4       <DataItem category="EVENT" id="estop" name="estop" type="
2700       EMERGENCY\textunderscore_STOP"/>
2701 5     </DataItems>
2702 6   </Controller>
2703 7 </Components>

```

Example 3: MTConnectStreams Response Document

```

2704 1 <ComponentStream component="Controller" name="controller"
2705   componentId="cont">
2706 2   <Events>
2707 3     <EmergencyStop dataItemId="estop" sequence="159" timestamp="
2708     2021-10-01T14:26:38.66869Z">ARMED</EmergencyStop>

```

```

2709 4    </Events>
2710 5    </ComponentStream>

```

2711 C.3 Sample Example

Example 4: Sample Example

```

2712 1    <Samples>
2713 2      <AccumulatedTime dataItemId="cut\textunderscore_time" sequence="
2714      75437" timestamp="2021-10-07T05:08:28.221704Z">1763070.0</
2715      AccumulatedTime>
2716 3      <CuttingSpeed dataItemId="cspeed" sequence="112" timestamp="
2717      2021-10-07T05:08:28.221704Z" subType="ACTUAL">UNAVAILABLE</
2718      CuttingSpeed>
2719 4    </Samples>

```

2720 C.4 Event Example

Example 5: Event Example

```

2721 1    <Events>
2722 2      <Availability dataItemId="avail" sequence="156" timestamp="
2723      2021-10-01T14:26:38.668505Z">AVAILABLE</Availability>
2724 3      <AssetRemoved assetType="CuttingTool" dataItemId="d1\
2725      textunderscore_asset\textunderscore_rem" sequence="140"
2726      timestamp="2021-10-01T11:40:08.101461Z">UNAVAILABLE</
2727      AssetRemoved>
2728 4    </Events>

```

2729 C.5 Condition Example

2730 Condition types are represented differently in XML when compared to Sample and
 2731 Event types. The element name is the condition state of the Condition type in Pascal
 2732 Case. The name of the Condition type is represented by the attribute type.

2733 If the condition state is unavailable then the element name is represented by Unavail-
 2734 able.

Example 6: Condition Example

```

2735 1    <Condition>
2736 2      <Unavailable dataItemId="cont\textunderscore_system" sequence="72"
2737      timestamp="2021-10-11T21:04:03.251999Z" type="SYSTEM"/>

```

```

2738 3    <Normal dataItemId="path\textunderscore_system" sequence="153"
2739         timestamp="2021-10-11T21:04:03.262845Z" type="SYSTEM"/>
2740 4    <Warning dataItemId="cont\textunderscore_system" nativecode="313"
2741         nativeSeverity="50" sequence="75573" timestamp="2021-10-07T05
2742         :08:58.518317Z" type="LOGIC\textunderscore_PROGRAM">PALLET ARM
2743         DOWN RS. MALF.</Warning>
2744 5    </Condition>

```

2745 C.6 DataSet Example

Example 7: DataSet Example

```

2746 1    <Events>
2747 2    <VariableDataSet count="2" dataItemId="cvars" sequence="126513"
2748         timestamp="2021-10-12T03:57:31.106559Z">
2749 3        <Entry key="100">66.3314</Entry>
2750 4        <Entry key="101">167.2</Entry>
2751 5    </VariableDataSet>
2752 6    </Events>

```

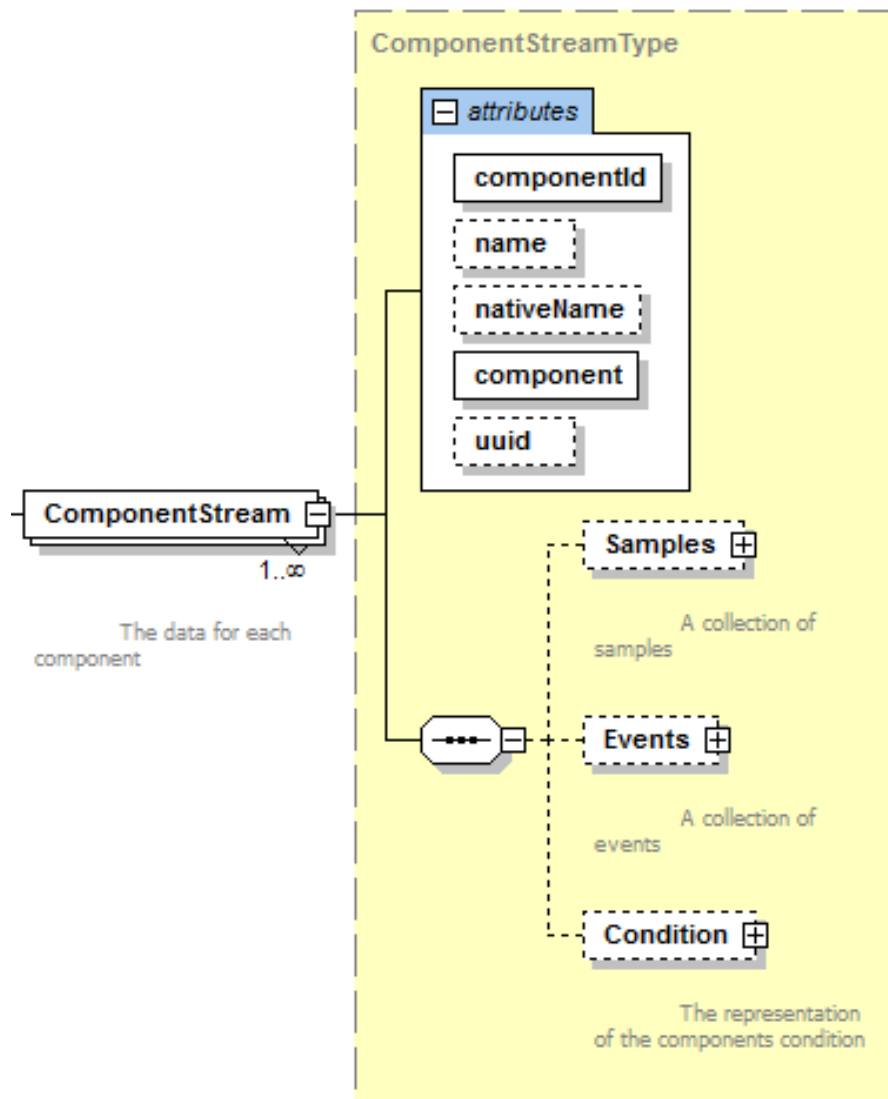
2753 C.7 Table Example

Example 8: Table Example

```

2754 1    <Events>
2755 2    <WorkOffsetTable count="2" dataItemId="woffset" sequence="222101"
2756         timestamp="2021-10-12T06:04:11.990531Z">
2757 3        <Entry key="G53.1">
2758 4            <Cell key="X">1</Cell>
2759 5            <Cell key="Y">2</Cell>
2760 6            <Cell key="Z">3</Cell>
2761 7        </Entry>
2762 8        <Entry key="G53.2">
2763 9            <Cell key="X">4</Cell>
2764 10           <Cell key="Y">5</Cell>
2765 11           <Cell key="Z">6</Cell>
2766 12        </Entry>
2767 13    </WorkOffsetTable>
2768 14    </Events>

```

**Figure 17:** ComponentStream Schema

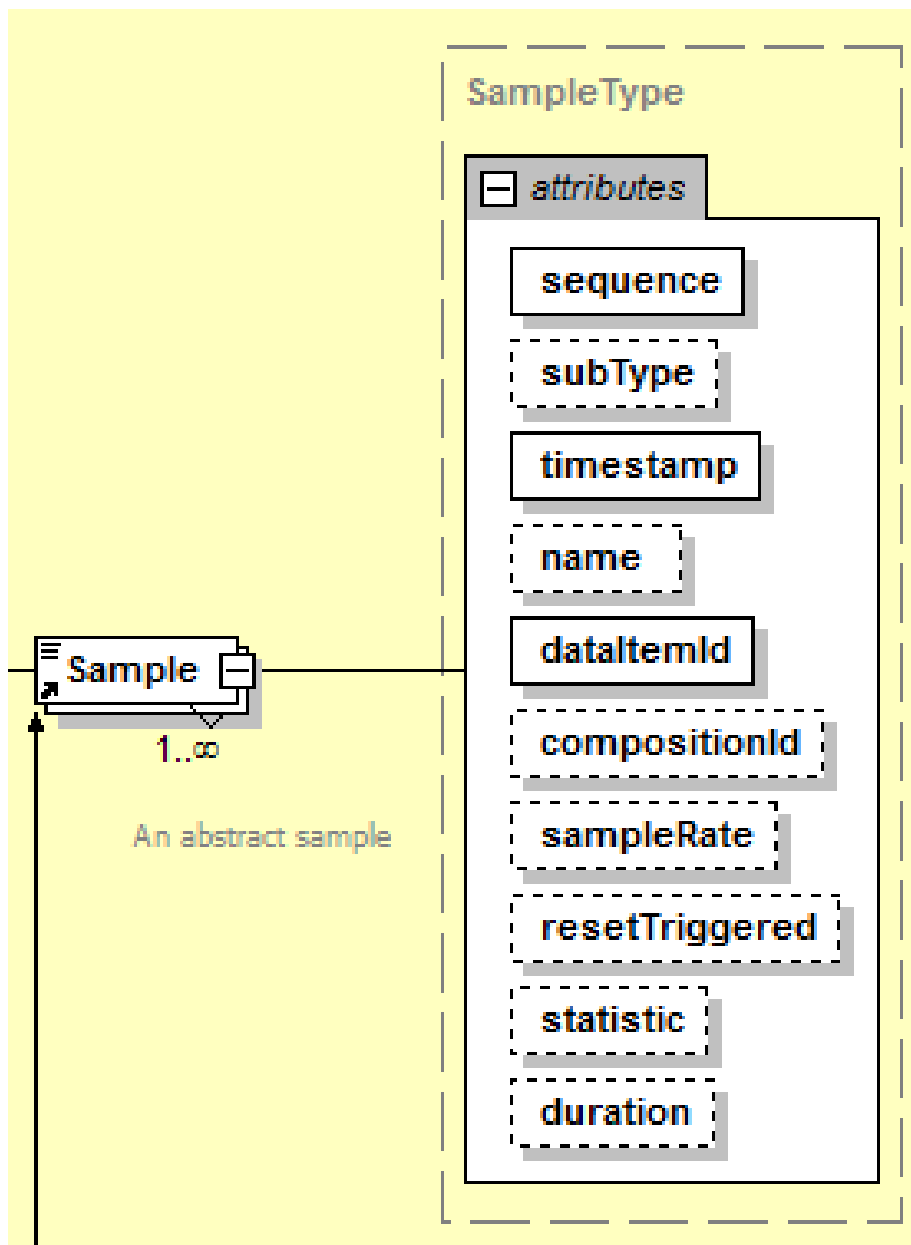


Figure 18: Sample Schema

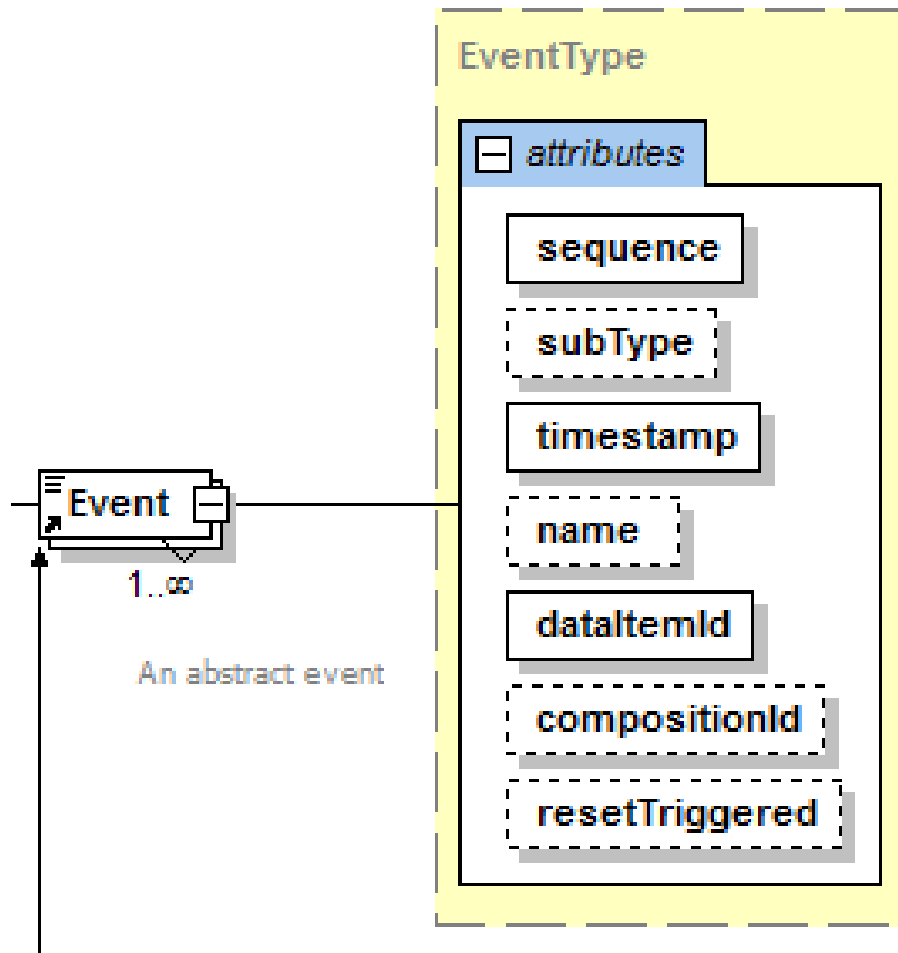


Figure 19: Event Schema

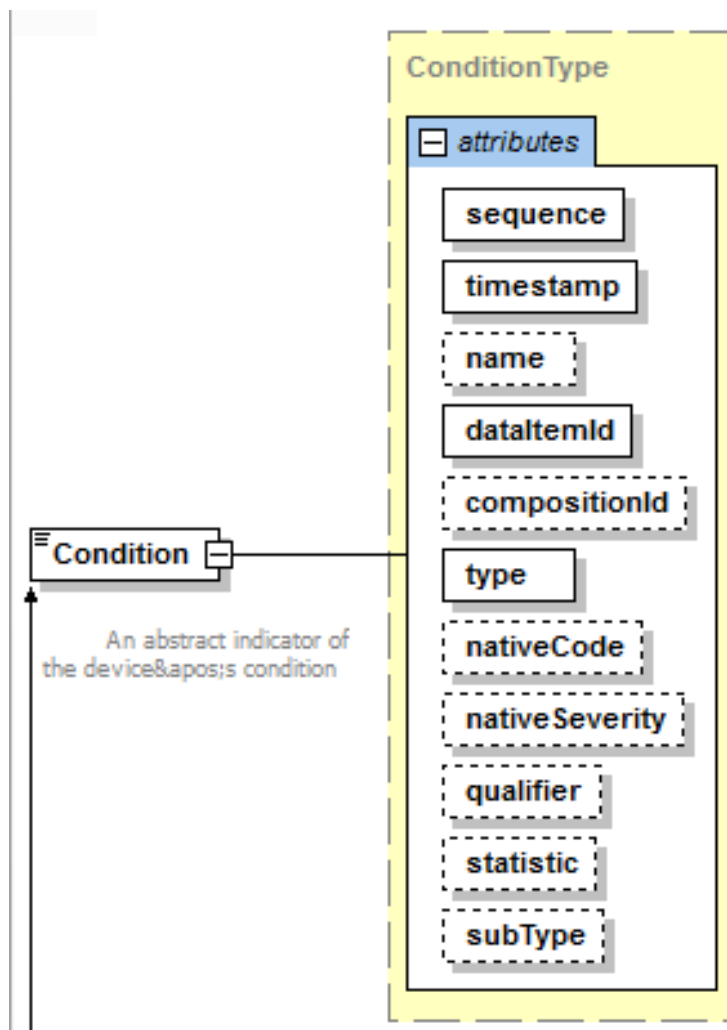


Figure 20: Condition Schema

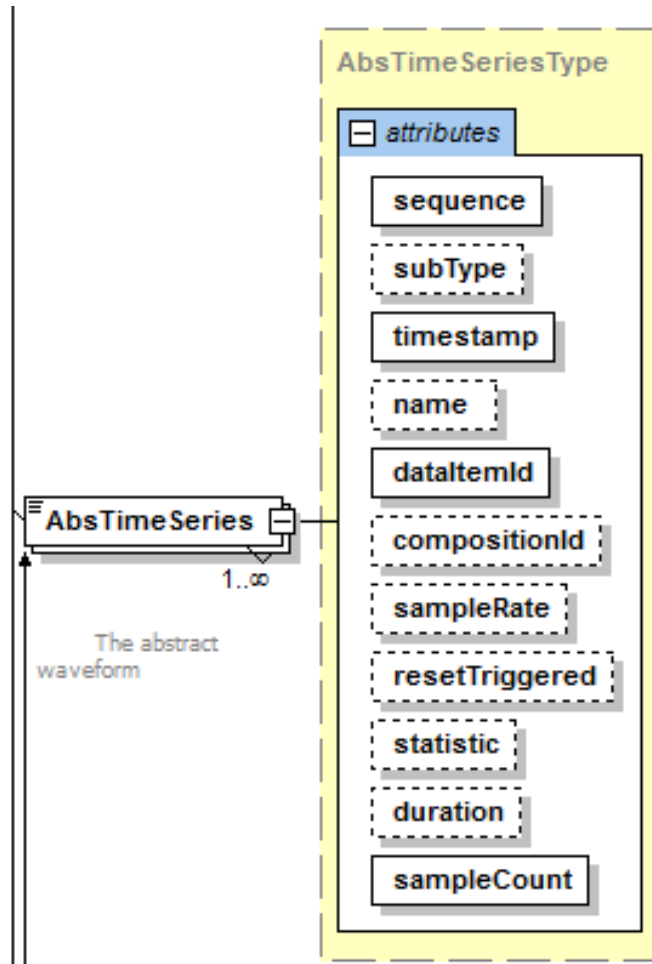


Figure 21: AbsTimeSeries Schema

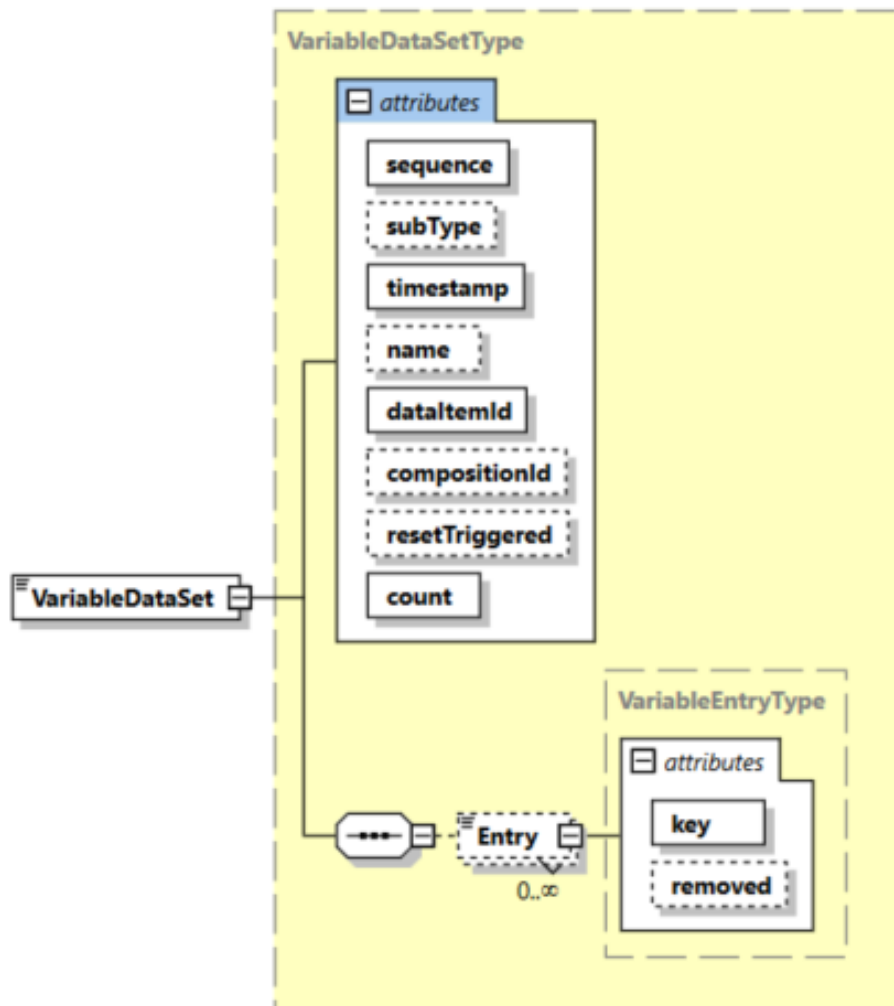


Figure 22: DataSet Schema

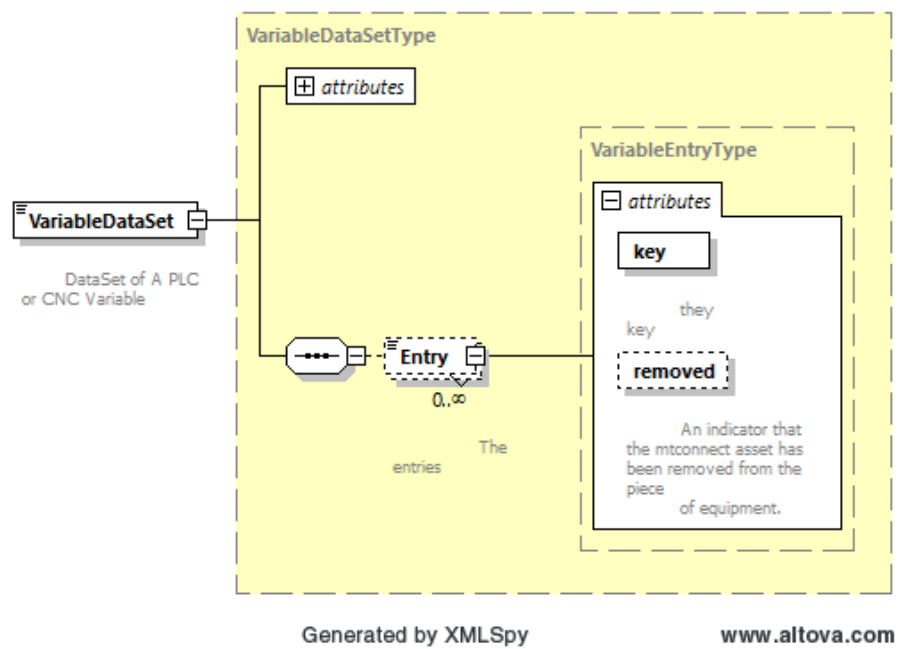


Figure 23: Entry Schema

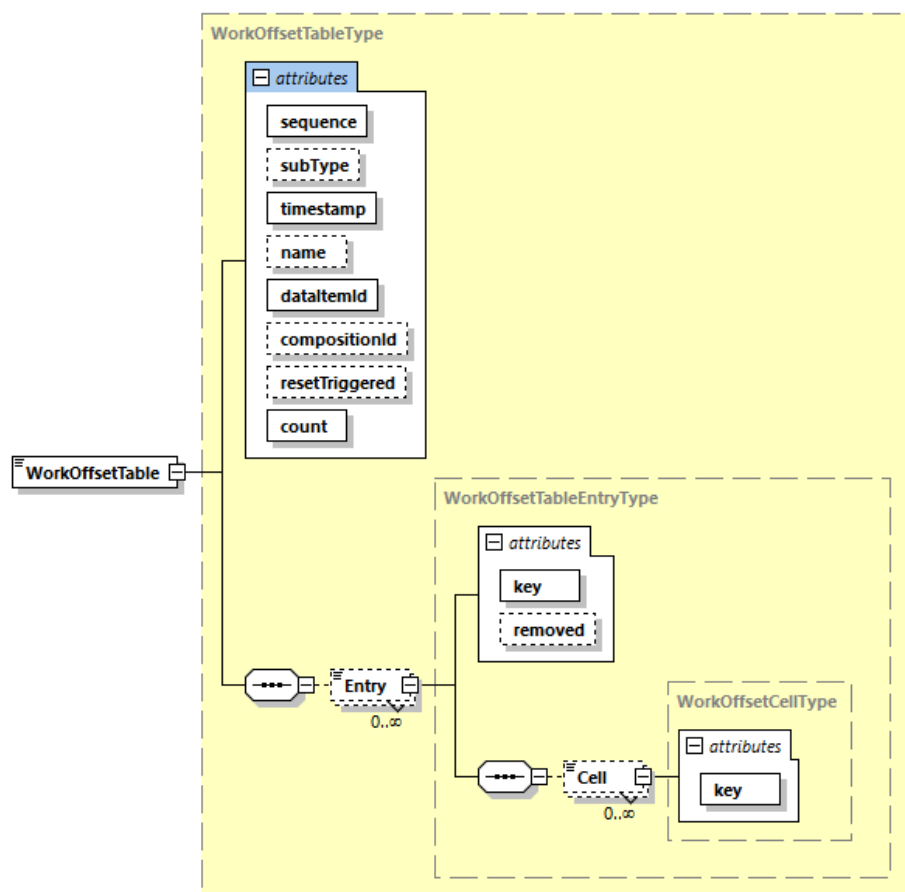


Figure 24: Table Schema