



MTConnect[®] Standard

Part 3.0 – Observation Information Model

Version 2.5.0

Prepared for: MTConnect Institute
Prepared from: MTConnectSysMLModel.xml
Prepared on: January 5, 2026

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	10
2.3	Protocol Terms	11
2.4	HTTP Terms	13
2.5	XML Terms	15
2.6	MTConnect Terms	16
2.7	Acronyms	17
2.8	MTConnect References	29
3	Observation Information Model	30
3.1	DeviceStream	31
3.1.1	Value Properties of DeviceStream	31
3.1.2	Part Properties of DeviceStream	32
3.2	ComponentStream	32
3.2.1	Value Properties of ComponentStream	33
3.2.2	Reference Properties of ComponentStream	33
3.3	Observation	34
3.3.1	Value Properties of Observation	37
3.4	Sample	39
3.4.1	Value Properties of Sample	39
3.5	Event	42
3.5.1	Value Properties of Event	42
3.6	Condition	44
3.6.1	Value Properties of Condition	44
4	Representations	48
4.1	Representation	48
4.2	Value	48
4.3	TimeSeries	49
4.3.1	Value Properties of TimeSeries	49
4.4	<<deprecated>>Discrete	50
4.5	DataSet	50
4.5.1	Value Properties of DataSet	53
4.5.2	Part Properties of DataSet	53
4.6	Table	53
4.6.1	Value Properties of Table	55
4.6.2	Part Properties of Table	55
4.7	Entry	56

4.7.1	Value Properties of Entry	56
4.8	TableEntry	57
4.8.1	Value Properties of TableEntry	57
4.8.2	Part Properties of TableEntry	58
4.9	Cell	58
4.9.1	Value Properties of Cell	58
5	Observation Types	60
5.1	Condition Types	60
5.1.1	Actuator	60
5.1.2	Communications	60
5.1.3	DataRange	60
5.1.4	LogicProgram	61
5.1.5	MotionProgram	61
5.1.6	System	61
5.2	Event Types	61
5.2.1	ActivationCount	61
5.2.2	ActiveAxes	62
5.2.3	ActivePowerSource	62
5.2.4	ActuatorState	63
5.2.5	AdapterSoftwareVersion	63
5.2.6	AdapterURI	63
5.2.7	<<deprecated>>Alarm	63
5.2.8	<<deprecated>>AlarmLimit	64
5.2.9	AlarmLimits	65
5.2.10	Application	66
5.2.11	AssetChanged	67
5.2.12	AssetCount	67
5.2.13	AssetRemoved	67
5.2.14	Availability	68
5.2.15	AxisCoupling	68
5.2.16	AxisFeedrateOverride	69
5.2.17	AxisInterlock	70
5.2.18	AxisState	70
5.2.19	BatteryState	71
5.2.20	Block	71
5.2.21	BlockCount	71
5.2.22	CharacteristicPersistentId	71
5.2.23	CharacteristicStatus	72
5.2.24	ChuckInterlock	72
5.2.25	ChuckState	73
5.2.26	ClockTime	74

5.2.27	<<deprecated>>Code	74
5.2.28	ComponentData	74
5.2.29	CompositionState	74
5.2.30	ConnectionStatus	76
5.2.31	<<deprecated>>ControlLimit	77
5.2.32	ControlLimits	78
5.2.33	ControllerMode	78
5.2.34	ControllerModeOverride	79
5.2.35	CoupledAxes	81
5.2.36	CycleCount	81
5.2.37	DateCode	82
5.2.38	DeactivationCount	82
5.2.39	DeviceAdded	83
5.2.40	DeviceChanged	84
5.2.41	DeviceRemoved	84
5.2.42	DeviceUuid	85
5.2.43	Direction	85
5.2.44	DoorState	86
5.2.45	EmergencyStop	87
5.2.46	EndOfBar	87
5.2.47	EquipmentMode	88
5.2.48	Execution	89
5.2.49	FeatureMeasurement	90
5.2.50	FeaturePersistantId	92
5.2.51	Firmware	92
5.2.52	FixtureId	93
5.2.53	FunctionalMode	93
5.2.54	Hardness	94
5.2.55	Hardware	95
5.2.56	HostName	96
5.2.57	LeakDetect	96
5.2.58	Library	96
5.2.59	<<deprecated>>Line	97
5.2.60	LineLabel	97
5.2.61	LineNumber	97
5.2.62	LoadCount	98
5.2.63	LocationAddress	99
5.2.64	LocationNarrative	101
5.2.65	LocationSpatialGeographic	101
5.2.66	LockState	102
5.2.67	MTConnectVersion	102
5.2.68	MaintenanceList	103

5.2.69	Material	104
5.2.70	MaterialLayer	104
5.2.71	MeasurementType	105
5.2.72	MeasurementUnits	105
5.2.73	MeasurementValue	105
5.2.74	Message	105
5.2.75	Network	106
5.2.76	NetworkPort	107
5.2.77	OperatingMode	107
5.2.78	OperatingSystem	108
5.2.79	OperatorId	108
5.2.80	PalletId	108
5.2.81	PartCount	109
5.2.82	PartCountType	110
5.2.83	PartDetect	110
5.2.84	PartGroupId	110
5.2.85	PartId	111
5.2.86	PartIndex	111
5.2.87	PartKindId	111
5.2.88	<<deprecated>>PartNumber	112
5.2.89	PartProcessingState	112
5.2.90	PartStatus	113
5.2.91	PartUniqueId	114
5.2.92	PathFeedrateOverride	114
5.2.93	PathMode	115
5.2.94	PowerState	115
5.2.95	<<deprecated>>PowerStatus	116
5.2.96	ProcessAggregateId	116
5.2.97	ProcessKindId	117
5.2.98	ProcessOccurrenceId	117
5.2.99	ProcessState	118
5.2.100	ProcessTime	118
5.2.101	Program	119
5.2.102	ProgramComment	120
5.2.103	ProgramEdit	120
5.2.104	ProgramEditName	121
5.2.105	ProgramHeader	121
5.2.106	ProgramLocation	121
5.2.107	ProgramLocationType	122
5.2.108	ProgramNestLevel	123
5.2.109	RotaryMode	123
5.2.110	RotaryVelocityOverride	123

5.2.111	Rotation	123
5.2.112	SensorAttachment	124
5.2.113	SensorState	124
5.2.114	SerialNumber	125
5.2.115	<<deprecated>>SpecificationLimit	125
5.2.116	SpecificationLimits	126
5.2.117	SpindleInterlock	127
5.2.118	Thickness	127
5.2.119	ToolAssetId	128
5.2.120	ToolCuttingItem	128
5.2.121	ToolGroup	128
5.2.122	<<deprecated>>ToolId	128
5.2.123	ToolNumber	129
5.2.124	ToolOffset	129
5.2.125	ToolOffsets	129
5.2.126	TransferCount	129
5.2.127	Translation	130
5.2.128	Uncertainty	131
5.2.129	UncertaintyType	131
5.2.130	UnloadCount	131
5.2.131	User	132
5.2.132	ValveState	133
5.2.133	Variable	134
5.2.134	WaitState	134
5.2.135	Wire	135
5.2.136	WorkOffset	135
5.2.137	WorkOffsets	135
5.2.138	WorkholdingId	135
5.3	Sample Types	135
5.3.1	Acceleration	136
5.3.2	AccumulatedTime	136
5.3.3	<<deprecated>>Amperage	136
5.3.4	AmperageAC	137
5.3.5	AmperageDC	138
5.3.6	Angle	138
5.3.7	AngularAcceleration	139
5.3.8	AngularDeceleration	139
5.3.9	AngularVelocity	140
5.3.10	AssetUpdateRate	140
5.3.11	AxisFeedrate	140
5.3.12	BatteryCapacity	141
5.3.13	BatteryCharge	141

5.3.14	CapacityFluid	142
5.3.15	CapacitySpatial	142
5.3.16	ChargeRate	142
5.3.17	Concentration	142
5.3.18	Conductivity	143
5.3.19	CuttingSpeed	143
5.3.20	Deceleration	143
5.3.21	Density	144
5.3.22	DepositionAccelerationVolumetric	144
5.3.23	DepositionDensity	144
5.3.24	DepositionMass	145
5.3.25	DepositionRateVolumetric	145
5.3.26	DepositionVolume	146
5.3.27	DewPoint	146
5.3.28	Diameter	146
5.3.29	DischargeRate	146
5.3.30	Displacement	147
5.3.31	DisplacementAngular	147
5.3.32	DisplacementLinear	147
5.3.33	ElectricalEnergy	148
5.3.34	EquipmentTimer	148
5.3.35	FillHeight	149
5.3.36	FillLevel	149
5.3.37	Flow	149
5.3.38	FollowingError	150
5.3.39	FollowingErrorAngular	150
5.3.40	FollowingErrorLinear	150
5.3.41	Frequency	151
5.3.42	<<deprecated>>GlobalPosition	151
5.3.43	GravitationalAcceleration	151
5.3.44	GravitationalForce	151
5.3.45	HumidityAbsolute	152
5.3.46	HumidityRelative	152
5.3.47	HumiditySpecific	152
5.3.48	Length	153
5.3.49	<<deprecated>>Level	153
5.3.50	LinearForce	154
5.3.51	Load	154
5.3.52	Mass	154
5.3.53	ObservationUpdateRate	154
5.3.54	Openness	154
5.3.55	Orientation	154

5.3.56	PH	155
5.3.57	ParticleCount	155
5.3.58	ParticleSize	156
5.3.59	PathFeedrate	156
5.3.60	PathFeedratePerRevolution	157
5.3.61	<<hasFormatSpecificRepresentation>>PathPosition	157
5.3.62	Position	158
5.3.63	PositionCartesian	159
5.3.64	PowerFactor	159
5.3.65	Pressure	159
5.3.66	PressureAbsolute	159
5.3.67	PressurizationRate	160
5.3.68	ProcessTimer	160
5.3.69	Resistance	161
5.3.70	Resistivity	161
5.3.71	RotaryVelocity	161
5.3.72	SettlingError	161
5.3.73	SettlingErrorAngular	162
5.3.74	SettlingErrorLinear	162
5.3.75	SoundLevel	162
5.3.76	<<deprecated>>SpindleSpeed	163
5.3.77	Strain	164
5.3.78	Temperature	164
5.3.79	Tension	164
5.3.80	Tilt	164
5.3.81	Torque	165
5.3.82	Velocity	165
5.3.83	Viscosity	165
5.3.84	VoltAmpere	165
5.3.85	VoltAmpereReactive	165
5.3.86	<<deprecated>>Voltage	165
5.3.87	VoltageAC	166
5.3.88	VoltageDC	167
5.3.89	VolumeFluid	167
5.3.90	VolumeSpatial	168
5.3.91	Wattage	169
5.3.92	XDimension	169
5.3.93	YDimension	169
5.3.94	ZDimension	169
6	Profile	170
6.1	DataTypes	170

6.1.1	boolean	170
6.1.2	ID	170
6.1.3	string	170
6.1.4	float	170
6.1.5	datetime	171
6.1.6	integer	171
6.1.7	xlinktype	171
6.1.8	xslang	171
6.1.9	SECOND	171
6.1.10	xlinkhref	171
6.1.11	x509	171
6.1.12	int32	172
6.1.13	int64	172
6.1.14	version	172
6.1.15	uint32	172
6.1.16	uint64	172
6.1.17	binary	172
6.1.18	double	172
6.1.19	Array	173
6.1.20	<<hasFormatSpecificRepresentation>>float3d	173
6.1.21	UUID	173
6.1.22	METER	173
6.2	Stereotypes	173
6.2.1	organizer	173
6.2.2	deprecated	173
6.2.3	extensible	173
6.2.4	informative	174
6.2.5	normative	174
6.2.6	observes	174
6.2.7	satisfiedBy	175
6.2.8	hasFormatSpecificRepresentation	175
6.2.9	valueType	175
6.2.10	isArray	175
6.2.11	MTConnectRequirementSpecification	175
Appendices		176
A	Bibliography	176
B	XML Schema Diagrams	178
B.1	Observations Schema Diagrams	178
B.2	Representation Schema Diagrams	178
C	XML Examples	178
C.1	DeviceStream Example	178

C.2	Observations made for DataItem Example	180
C.3	Sample Example	180
C.4	Event Example	181
C.5	Condition Example	181
C.6	DataSet Example	182
C.7	Table Example	182

Table of Figures

Figure 1: Observations	35
Figure 2: DeviceStream Example	35
Figure 3: Observations made for DataItem Example	36
Figure 4: Sample Example	40
Figure 5: Event Example	43
Figure 6: Condition Example	45
Figure 7: Representation	48
Figure 8: TemperatureTimeSeries	49
Figure 9: VariableDataSet	51
Figure 10:DataSet Example	52
Figure 11:WorkOffsetTable	54
Figure 12:Table Example	55
Figure 13:DataTypes	170
Figure 14:Stereotypes	174

List of Tables

Table 1: Value Properties of DeviceStream	32
Table 2: Part Properties of DeviceStream	32
Table 3: Value Properties of ComponentStream	33
Table 4: Reference Properties of ComponentStream	34
Table 5: Value Properties of Observation	37
Table 6: Value Properties of Sample	40
Table 7: Value Properties of Event	42
Table 8: Value Properties of Condition	44
Table 9: Value Properties of TimeSeries	50
Table 10: Value Properties of DataSet	53
Table 11: Part Properties of DataSet	53
Table 12: Value Properties of Table	55
Table 13: Part Properties of Table	56
Table 14: Value Properties of Entry	57
Table 15: Value Properties of TableEntry	57
Table 16: Part Properties of TableEntry	58
Table 17: Value Properties of Cell	59
Table 18: Value Properties of Alarm	64
Table 19: Value Properties of AssetChanged	67
Table 20: Value Properties of AssetRemoved	68
Table 21: Value Properties of DeviceAdded	84
Table 22: Value Properties of DeviceChanged	84
Table 23: Value Properties of DeviceRemoved	85
Table 24: Value Properties of Message	106

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	string	1
uuid	UUID	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 **MUST** be the same as the value defined for the name
97 in the *MTConnectDevices Response Document*.

- 98 • uuid

99 uuid of the Device. The value reported for uuid **MUST** be the same as the value
100 defined for the uuid of the same Device in the *MTConnectDevices Response*
101 *Document*.

102 3.1.2 Part Properties of DeviceStream

103 Table 2 lists the Part Properties of DeviceStream.

Part Property name	Multiplicity
ComponentStream	1..*

Table 2: Part Properties of DeviceStream

104 Descriptions for Part Properties of DeviceStream:

- 105 • ComponentStream

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

108 See Section 3.2 - *ComponentStream*.

109 3.2 ComponentStream

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

At least one of Sample, Event, or Condition **MUST** be organized by a ComponentStream entity.

3.2.1 Value Properties of ComponentStream

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	UUID	0..1

Table 3: Value Properties of ComponentStream

Descriptions for Value Properties of ComponentStream:

- component
identifies the Component type associated with the ComponentStream.
Examples of component are Device, Controller, Linear and Loader.
- componentId
identifier of the Component as defined by the id in the *MTConnectDevices Response Document*.
- name
name of the Component associated with the ComponentStream.
- nativeName
common name of the Component associated with the ComponentStream.
- uuid
uuid of the Component associated with the ComponentStream.

3.2.2 Reference Properties of ComponentStream

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

131 Descriptions for Reference Properties of ComponentStream:

- 132 • Event
- 133 Observation that is a discrete piece of information from a piece of equipment.
- 134 Events groups one or more Event entities. See *Section 3.5 - Event*.
- 135 • Sample
- 136 Observation that is continuously changing or analog data value.
- 137 Samples groups one or more Sample entities. See *Section 3.4 - Sample*.
- 138 • Condition
- 139 Observation that provides the condition of a piece of equipment or a *Compo-*
- 140 *nent*.
- 141 Conditions groups one or more Condition entities. See *Section 3.6 - Condi-*
- 142 *tion*.
- 143 Note: In the Extensible Markup Language (XML) representation, Con-
- 144 ditions **MUST** appear as Condition element in the *MTConnect-*
- 145 *Streams Response Document*.

146 3.3 Observation

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

148 Note: See *Section B.1 - Observations Schema Diagrams* for XML schema.

149 The XML schema also shows differences in XML representation of certain

150 MTConnect entities.

151 Figure 2 shows a complete example of DeviceStream for the Device shown in *MT-*

152 *Connect Standard: Part 2.0 - Device Information Model*.

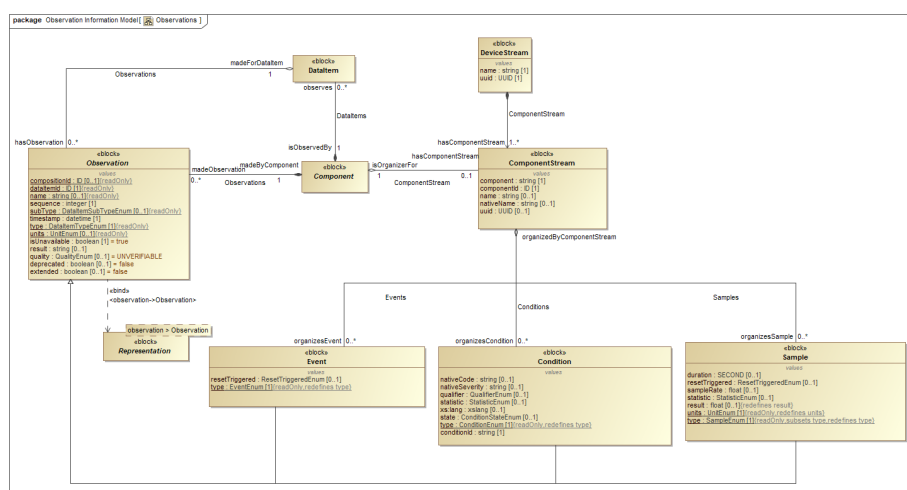


Figure 1: Observations

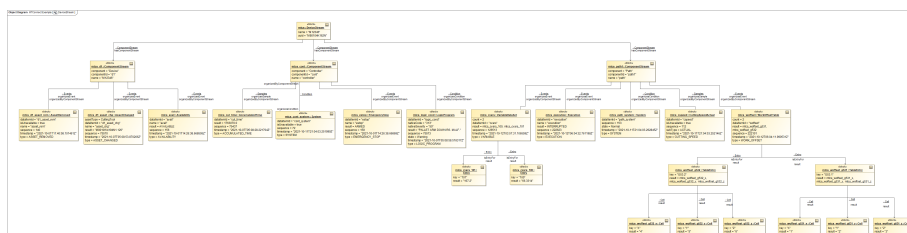


Figure 2: DeviceStream Example

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

154 This section provides semantic information for the Observation model.

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

157 3.3.0.1 Observations made for DataItem

158 Component **observes** DataItem entities and creates Observation entities for the
159 DataItem entities. See Figure 1.

```

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

```

162 Note: See *Section C.2 - Observations made for DataItem Example* for how

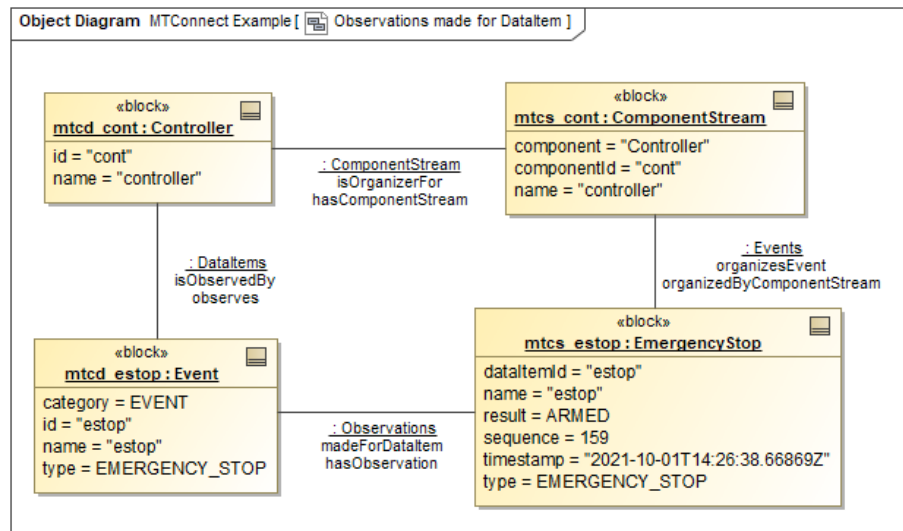


Figure 3: Observations made for DataItem Example

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 associated 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 is used to further describe that DataItem in the *MTConnectDevices Response Document*. See Section 4 - Representations for more details.

179 3.3.1 Value Properties of Observation

180 *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
isUnavailable	boolean	1

Table 5: Value Properties of Observation

181 Descriptions for Value Properties of Observation:

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

- 198 • `subType`
199 subtype of the `DataItem` associated with this `Observation`.
200 `subType` **MUST** match the associated `subType` defined in the *MTConnectDe-*
201 *VICES Response Document*.
202 The value of `subType` **MUST** be one of the `DataItemSubTypeEnum` enumer-
203 ation.
 - 204 • `timestamp`
205 most accurate time available to a piece of equipment that represents the point in time
206 that the data reported was measured.
 - 207 • `type`
208 type of the `DataItem` associated with this `Observation`.
209 `type` **MUST** match the associated `type` defined in the *MTConnectDevices Re-*
210 *sponse Document*.
211 The value of `type` **MUST** be one of the `DataItemTypeEnum` enumeration.
 - 212 • `units`
213 units of the `DataItem` associated with this `Observation`.
214 `units` **MUST** match the associated `units` defined in the *MTConnectDevices Re-*
215 *sponse Document*.
216 The value of `units` **MUST** be one of the `UnitEnum` enumeration.
 - 217 • `isUnavailable`
218 when `true`, result is indeterminate.
- 219 Note 1 to entry: In XML, when `isUnavailable` is `true`, the XML
220 CDATA of the `Observation` **MUST** be UNAVAILABLE. ““xml
221 <Execution dataItemId=”...” ...>UNAVAILABLE</Execution> ““
- 222 Note 2 to entry: In JavaScript Object Notation (JSON), when `isUnavail-`
223 `able` is `true`, the JSON value of the `Observation` **MUST** be UN-
224 AVAILABLE.json `"Execution" : ["dataItemId": "..."`
225 `..., "value": "UNAVAILABLE"]`
- 226 • `result`
227 observation of the `Observation` entity.
228 The default value type for `result` is `string`.

229 Note 1 to entry: in XML the result is the CDATA of the Observation
230 element.

231 ~~~~xml <Execution dataItemId="..."...>READY</Execution> ~~~~

232 Note 2 to entry: in JSON the result is the member value of the
233 Observation object.

234 ~~~~json "Execution": ["dataItemId": "...", "value": "READY"] ~~~~

235 3.4 Sample

236 Observation that is continuously changing or analog data value.

237 It provides the information and data reported from a piece of equipment for those DataItem
238 entities defined with category as SAMPLE in the *MTConnectDevices Response Docu-*
239 *ment*.

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

241 Figure 4 shows Sample type examples. It also shows an example for when the result
242 is not available (dataItemId=cspend).

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

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

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

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

248 3.4.1 Value Properties of Sample

249 Table 6 lists the Value Properties of Sample.

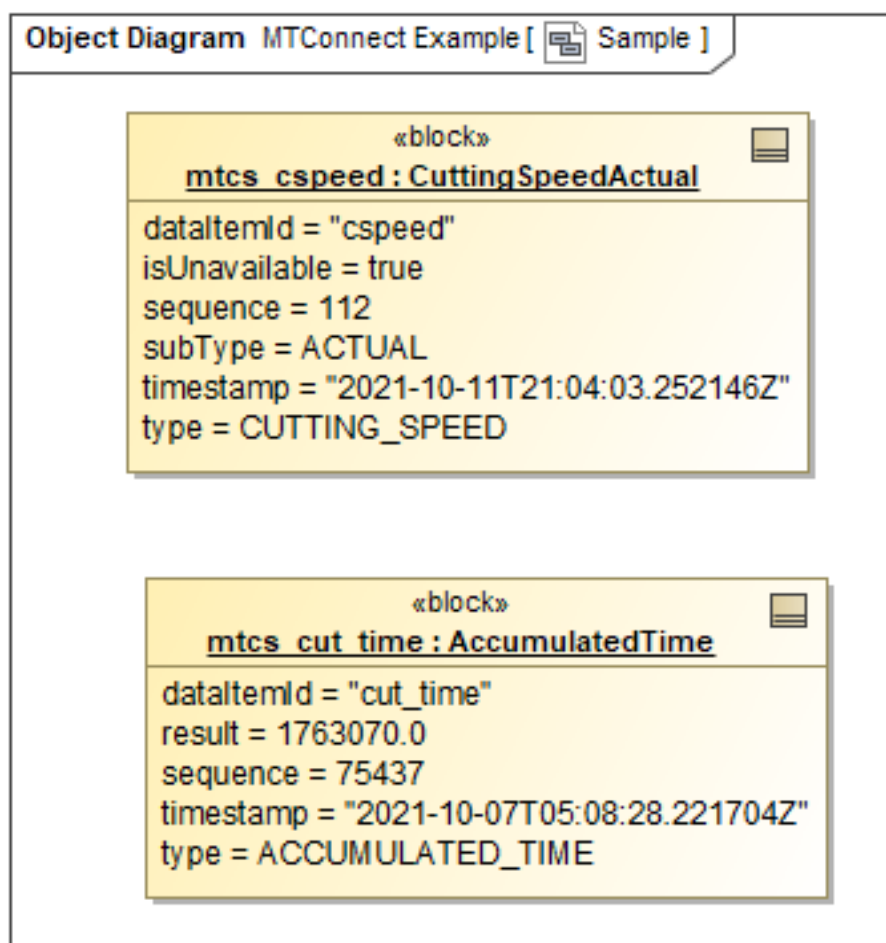


Figure 4: Sample Example

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

250 Descriptions for Value Properties of Sample:

- 251 • `duration`
 252 time-period over which the data was collected.
 253 `duration` **MUST** be provided when the `statistic` is defined in the *MTConnectDevices Response Document*.
 254
- 255 • `resetTriggered`
 256 identifies when a reported value has been reset and what has caused that reset to
 257 occur for those `DataItem` entities that may be periodically reset to an initial value.
 258 `resetTriggered` **MUST** only be provided for the specific occurrence of a `DataItem`
 259 reported in the *MTConnectStreams Response Document* when the reset occurred.
 260 ResetTriggeredEnum Enumeration:
- 261 – ACTION_COMPLETE
 262 `result` is measuring an action or operation was reset upon completion of that
 263 action or operation.
- 264 – ANNUAL
 265 `result` was reset at the end of a 12-month period.
- 266 – DAY
 267 `result` was reset at the end of a 24-hour period.
- 268 – MAINTENANCE
 269 `result` was reset upon completion of a maintenance event.
- 270 – MANUAL
 271 `result` was reset based on a physical reset action.
- 272 – MONTH
 273 `result` was reset at the end of a monthly period.
- 274 – POWER_ON
 275 `result` was reset when power was applied to the piece of equipment after a
 276 planned or unplanned interruption of power has occurred.
- 277 – SHIFT
 278 `result` was reset at the end of a work shift.
- 279 – WEEK
 280 `result` was reset at the end of a 7-day period.
- 281 • `sampleRate`
 282 rate at which successive samples of the value are recorded.

- 283 • `statistic`
- 284 type of statistical calculation defined by the `statistic` defined in the *MTConnectDevices Response Document*.
- 285
- 286 The value of `statistic` **MUST** be one of the `StatisticEnum` enumeration.

287 3.5 Event

- 288 Observation that is a discrete piece of information from a piece of equipment.
- 289 It provides the information and data reported from a piece of equipment for those `DataItem`
 290 entities defined with `category` as `EVENT` in the *MTConnectDevices Response Document*.
 291
- 292 Figure 5 shows `Event` type examples. It also shows an example for when the `result` is
 293 not available (`dataItemId=dl_asset_rem`).

294 Note: See *Example 9* for the XML representation of the same example.

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

297 3.5.1 Value Properties of Event

298 *Table 7* lists the Value Properties of `Event`.

Value Property name	Value Property type	Multiplicity
<code>resetTriggered</code>	<code>ResetTriggeredEnum</code>	0..1

Table 7: Value Properties of Event

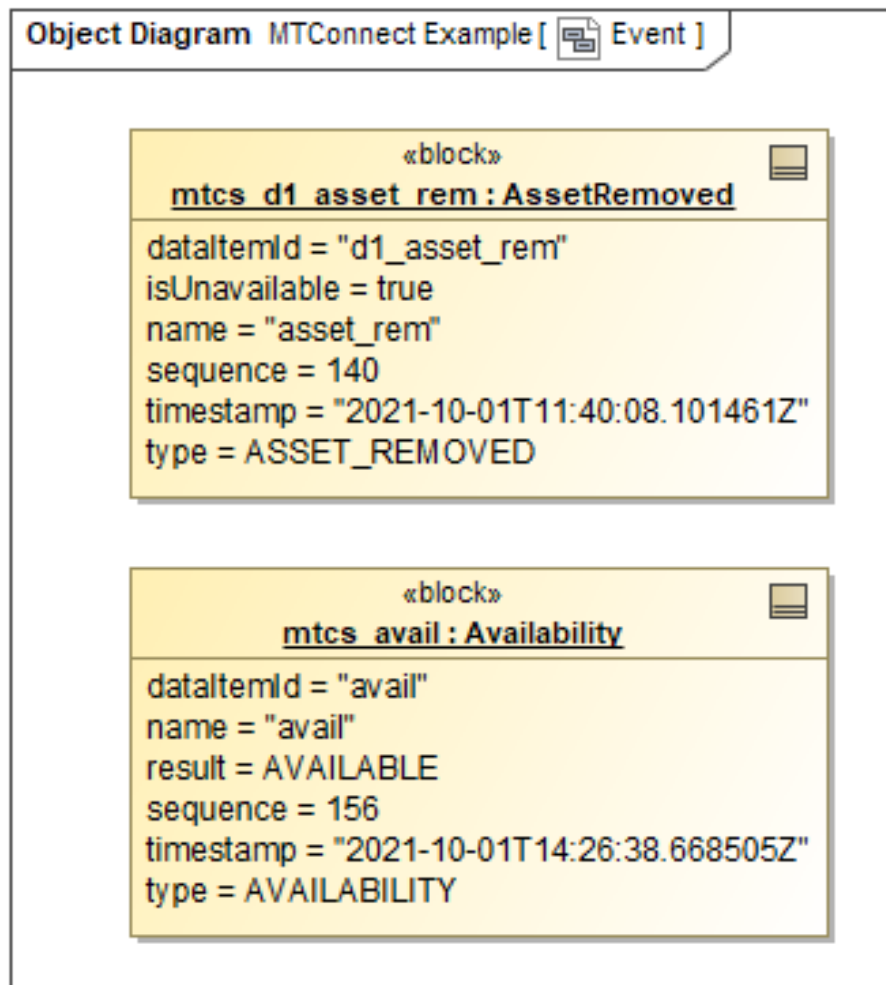


Figure 5: Event Example

299 Descriptions for Value Properties of Event:

300 • resetTriggered

301 identifies when a reported value has been reset and what has caused that reset to
302 occur for those DataItem entities that may be periodically reset to an initial value.

303 resetTriggered **MUST** only be provided for the specific occurrence of a DataItem
304 reported in the *MTConnectStreams Response Document* when the reset occurred.

305 The value of resetTriggered **MUST** be one of the ResetTriggeredEnum
306 enumeration.

307 3.6 Condition

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

309 It provides the information and data reported from a piece of equipment with category
310 as CONDITION in the *MTConnectDevices Response Document*.

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

314 Note: See *Example 10* for the XML representation of the same example.

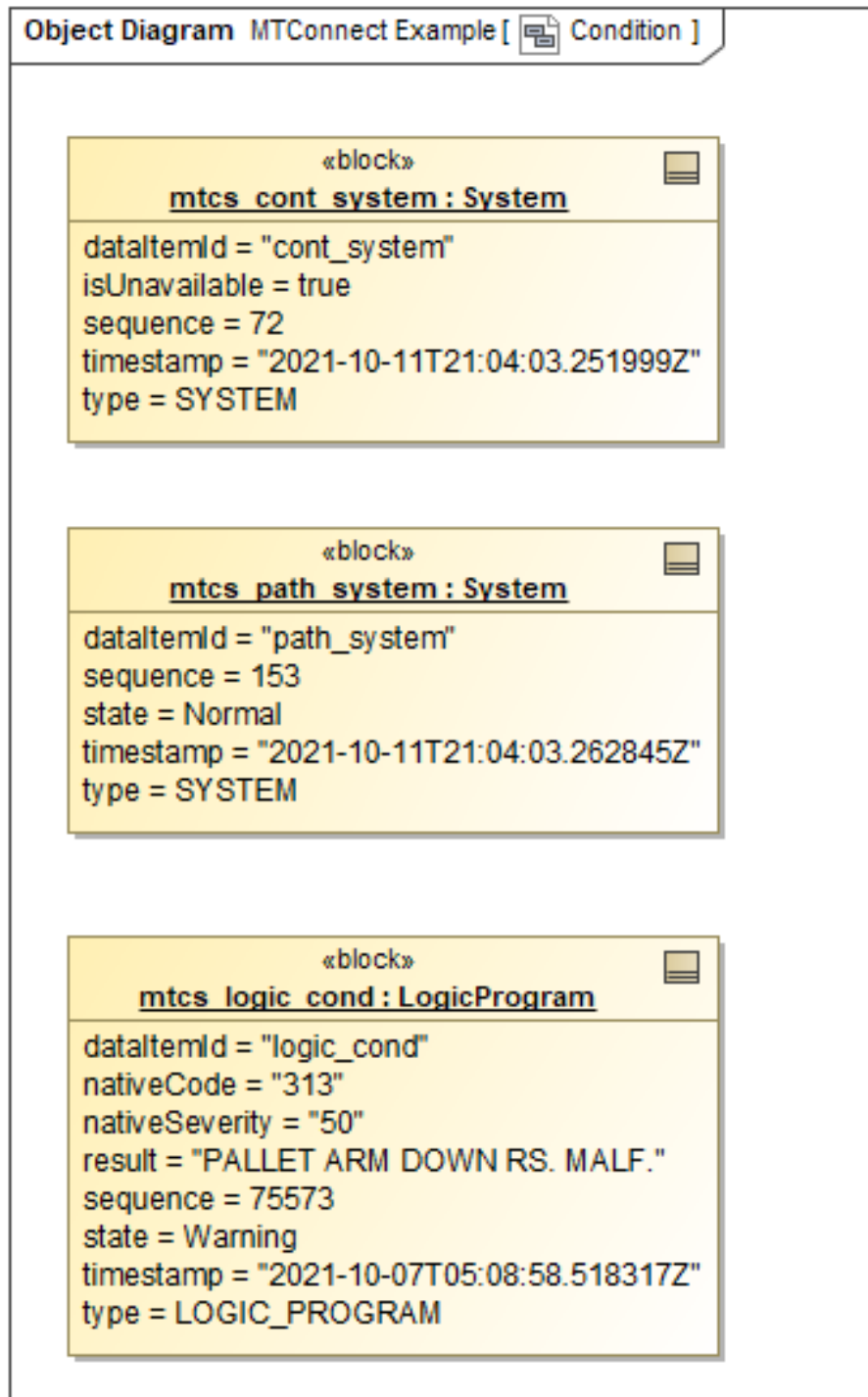
315 The following *Section 3.6.1 - Value Properties of Condition* lists the additional and/or
316 updated attributes for Condition.

317 3.6.1 Value Properties of Condition

318 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
conditionId	string	1

Table 8: Value Properties of Condition

**Figure 6:** Condition Example

319 Descriptions for Value Properties of Condition:

- 320 • `nativeCode`
- 321 `nativeCode` is the proprietary identifier designating a specific alarm, fault or warning
- 322 code provided by the piece of equipment.
- 323 • `nativeSeverity`
- 324 severity information to a client software application if the piece of equipment desig-
- 325 nates a severity level to a fault.
- 326 • `qualifier`
- 327 additional information regarding a condition state associated with the measured
- 328 value of a process variable.
- 329 `qualifier` defines whether the condition state represented indicates a measured
- 330 value that is above or below an expected value of a process variable.
- 331 QualifierEnum Enumeration:
- 332 – HIGH
- 333 measured value is greater than the expected value for a process variable.
- 334 – LOW
- 335 measured value is less than the expected value for a process variable.
- 336 • `statistic`
- 337 `statistic` provides additional information describing the meaning of the Con-
- 338 dition entity.
- 339 `statistic` **MUST** match the `statistic` defined in the *MTConnectDevices Re-*
- 340 *sponse Document*.
- 341 The value of `statistic` **MUST** be one of the `StatisticEnum` enumeration.
- 342 • `xs:lang`
- 343 specifies the language of the `result` returned for the `Condition`.
- 344 See *Ref IETF RFC 4646* (<http://www.ietf.org/rfc/rfc4646.txt>).
- 345 • `state`
- 346 condition state of the piece of equipment or `Component`.
- 347 ConditionStateEnum Enumeration:

- 348 – Fault
- 349 condition state that requires intervention to continue operation to function prop-
- 350 erly.
- 351 – Normal
- 352 condition state that indicates operation within specified limits.
- 353 – Warning
- 354 condition state that requires concern and supervision and may become haz-
- 355 ardous if no action is taken.

- 356 • conditionId
- 357 identifier of an individual *condition activation* provided by a piece of equipment.
- 358 conditionId **MUST** be unique for all concurrent *condition activations*.
- 359 conditionId **MUST** be maintained for all state transitions related to the same
- 360 *condition activation*.
- 361 Multiple conditionIds **MAY** exist for the same nativeCode.
- 362 If conditionId is not given, the value is the nativeCode. If nativeCode
- 363 and conditionId are not given, conditionId **MUST** be generated.

364 4 Representations

365 This section provides semantic information for the Representation model.

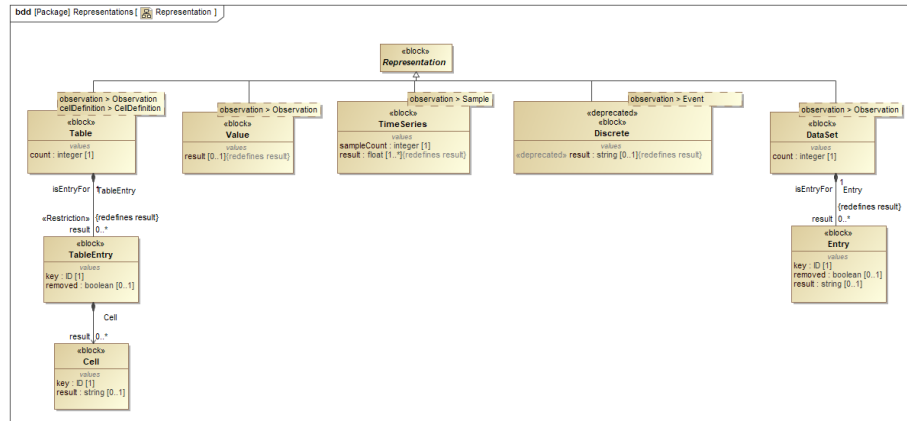


Figure 7: Representation

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

367 4.1 Representation

368 specifies the format and structure of result.

369 The Representation type for an Observation is defined by the associated rep-
370 resentation in the *MTConnectDevices Response Document*.

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

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

374 Example: The name for Sample Observation type Temperaturewith Repre-
375 sentation type of TimeSeries becomes TemperatureTimeSeries.

376 4.2 Value

377 default Representation type for all Observation types where result type is an
378 MTConnect data type. See Section 6.1 - DataTypes.

379 4.3 TimeSeries

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

381 TimeSeries for an Observation is defined by the associated representation
382 as TIME_SERIES.

383 representation as TIME_SERIES **MUST** have category of SAMPLE.

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

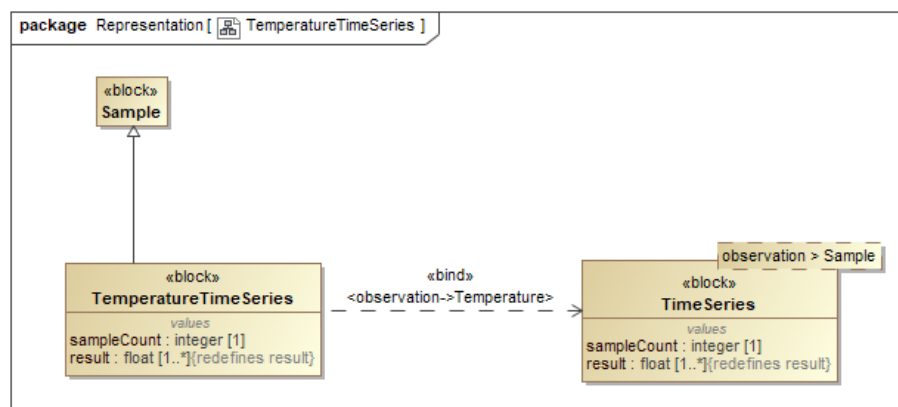


Figure 8: TemperatureTimeSeries

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

387 TimeSeries **MUST** report multiple values at fixed intervals in a single Observation.
388 At minimum, one of `sampleRate` or `sampleRate` **MUST** be specified. When both
389 are specified, the `sampleRate` supersedes the `sampleRate`.

390 `timestamp` **MUST** be set to the time the last value was observed. The duration **MAY**
391 indicate the time interval from the first to the last value in the series.

392 *Section 4.3.1 - Value Properties of TimeSeries* defines additional attributes for an Obser-
393 vation with TimeSeries Representation type.

394 4.3.1 Value Properties of TimeSeries

395 Table 9 lists the Value Properties of TimeSeries.

Value Property name	Value Property type	Multiplicity
sampleCount	integer	1

Table 9: Value Properties of TimeSeries

396 Descriptions for Value Properties of TimeSeries:

- 397 • sampleCount
398 number of values given for the Observation.

399 4.4 <<deprecated>>Discrete

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

402 Discrete for an Observation is defined by the associated representation as
403 DISCRETE.

404 representation as DISCRETE **MUST** have category as EVENT.

405 *MTConnect Version 1.5* replaced representation as DISCRETE with discrete.

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

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

410 4.5 DataSet

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

413 DataSet for an Observation is defined by the associated representation as
414 DATA_SET.

415 representation as DATA_SET **MUST** have category as SAMPLE or EVENT.

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

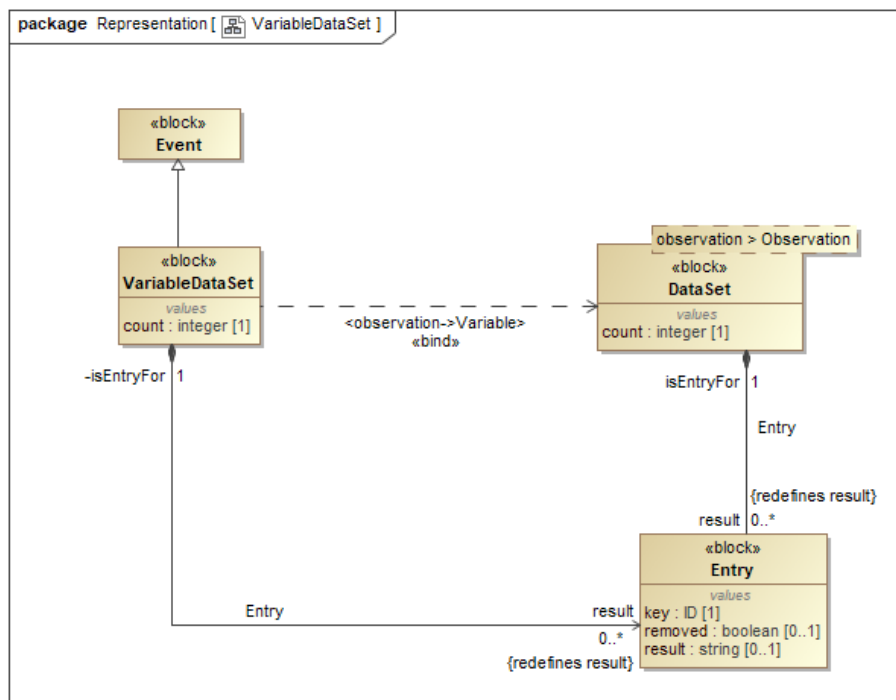


Figure 9: VariableDataSet

418 DataSet reports multiple values as a set of *key-value pair* where each *key* **MUST** be
 419 unique. The representation of the *key-value pair* is an Entry. The value of each Entry
 420 **MUST** have the same constraints and format as the Observation defined for repre-
 421 sentation as VALUE for the DataItem type. (See Value).

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

423 Figure 10 shows Event Observation type Variable with a Representation
 424 type of DataSet.

425 Note: See *Example 11* for the XML representation of the same example.

426 4.5.0.1 Management of Data Set Observations

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

429 One or more *key-value pairs* **MAY** be added, removed, or changed in an Observation.
 430 An agent **MUST** publish the changes to one or more *key-value pairs* as a single Obser-
 431 vation. An agent **MUST** indicate the removal of a *key-value pair* from a DataSet

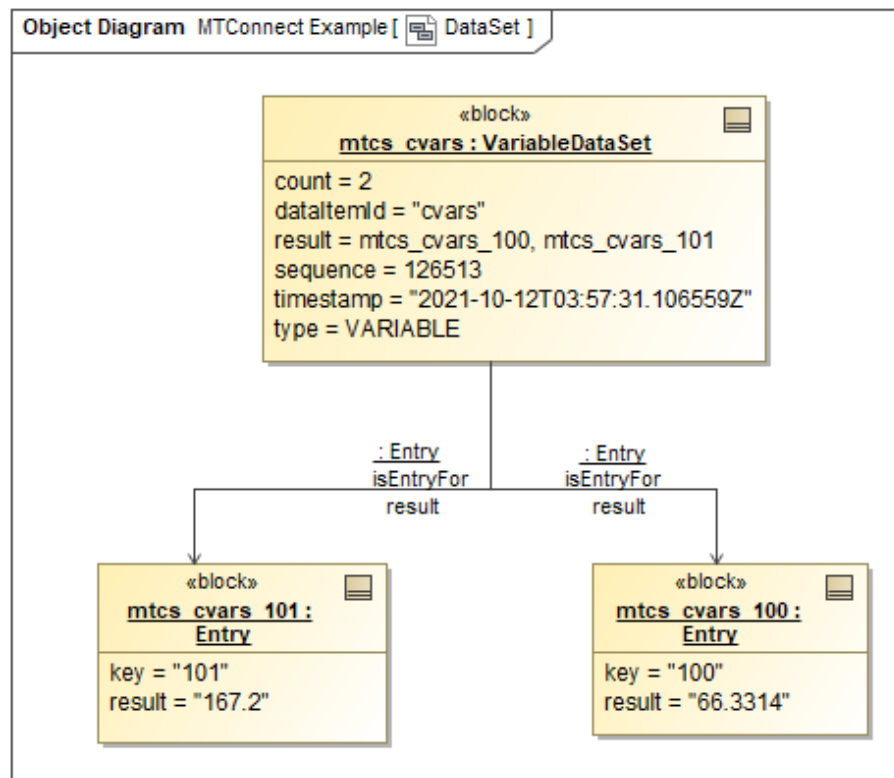


Figure 10: DataSet Example

- 432 using the removed as true.
- 433 When the discrete is false or is not present, an agent in response to a *sample request*
- 434 **MUST** only publish the changed *key-value pair* since the previous state of the DataSet.
- 435 When the discrete attribute is true, an agent, in response to a *sample request*, **MUST**
- 436 report all *key-value pairs* ignoring the state of the DataSet.
- 437 When an agent responds to a *current request*, the *response document* **MUST** include the
- 438 full set of *key-value pairs*. If the *current request* includes an *at query* parameter, the agent
- 439 **MUST** provide the set of *key-value pairs* at the *sequence number*.
- 440 When an Observation *reset* occurs, the DataSet **MUST** remove all *key-value pairs*
- 441 making the set empty. The Observation **MAY** simultaneously populate the DataSet
- 442 with new *key-value pairs*. The previous entries **MUST NOT** be included and **MUST NOT**
- 443 have removed as true.
- 444 When the Observation is UNAVAILABLE the DataSet **MUST** remove all *key-value*
- 445 *pairs* making the set empty.

446 4.5.1 Value Properties of DataSet

447 *Table 10* lists the Value Properties of DataSet.

Value Property name	Value Property type	Multiplicity
count	integer	1

Table 10: Value Properties of DataSet

448 Descriptions for Value Properties of DataSet:

- 449 • count
- 450 number of Entry elements for the Observation.

451 4.5.2 Part Properties of DataSet

452 *Table 11* lists the Part Properties of DataSet.

Part Property name	Multiplicity
Entry	0..*

Table 11: Part Properties of DataSet

453 Descriptions for Part Properties of DataSet:

- 454 • Entry
- 455 *key-value pair* published as part of a DataSet.
- 456 See *Section 4.7 - Entry*.

457 4.6 Table

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

461 Table for an Observation is defined by the associated representation as TA-
 462 BLE.

463 representation as TABLE **MUST** have category as SAMPLE or EVENT.

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

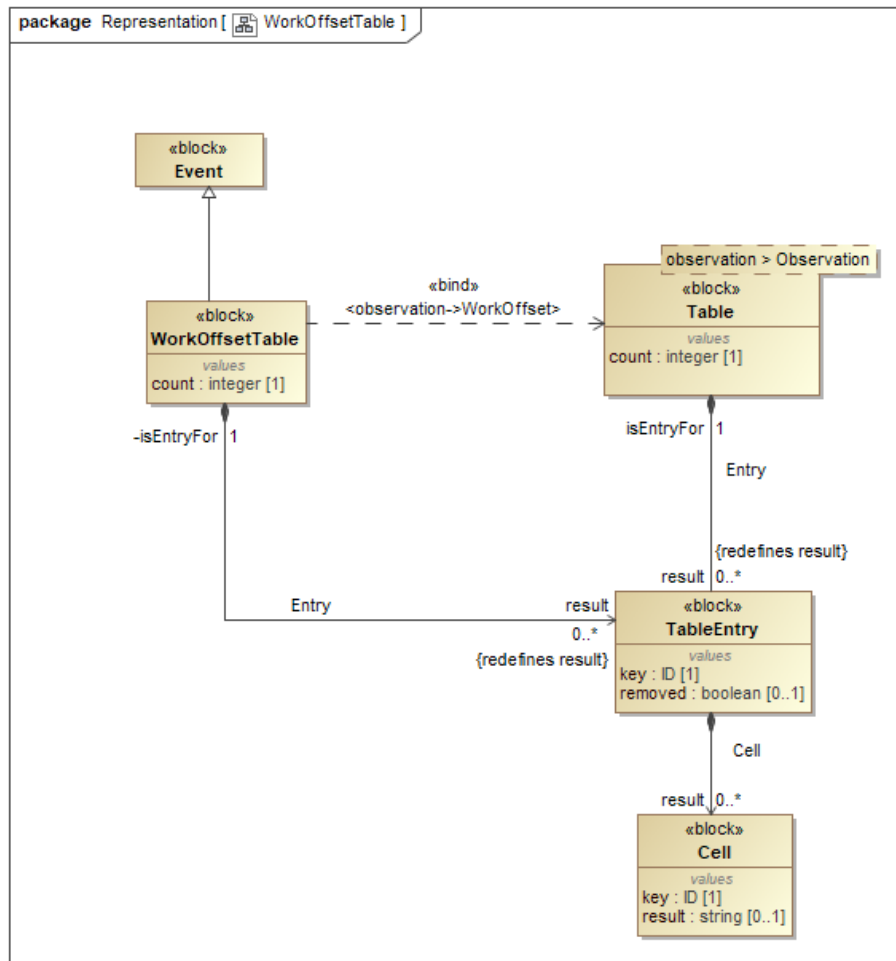


Figure 11: WorkOffsetTable

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

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

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

key **MUST** be the unique identity of the Entry within an Observation. key **MUST** be the unique identity of the Cell within an Entry.

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

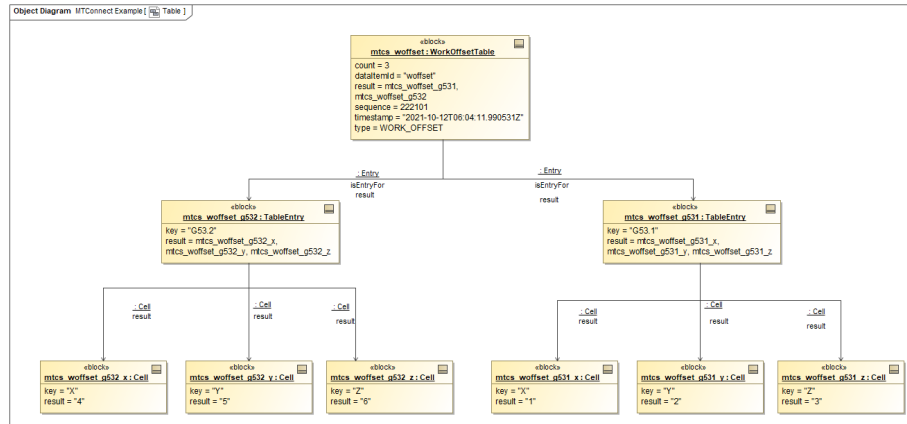


Figure 12: Table Example

Note: See Example 12 for the XML representation of the same example.

4.6.1 Value Properties of Table

Table 12 lists the Value Properties of Table.

Value Property name	Value Property type	Multiplicity
count	integer	1

Table 12: Value Properties of Table

Descriptions for Value Properties of Table:

- count
number of *key-value* pairs represented as Entry entities.

4.6.2 Part Properties of Table

Table 13 lists the Part Properties of Table.

Part Property name	Multiplicity
TableEntry	0..*

Table 13: Part Properties of Table

484 Descriptions for Part Properties of Table:

- 485 • TableEntry
486 *key-value pair* published as part of a Table.

487 Note: In the XML representation, TableEntry **MUST** appear as En-
488 try.

489 See *Section 4.8 - TableEntry*.

490 4.7 Entry

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

492 The value of each Entry **MUST** have the same restrictions as the value of an observation
493 with representation as VALUE.

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

499 EntryDefinition **MAY** provide the type and units of an key.

500 4.7.1 Value Properties of Entry

501 *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

502 Descriptions for Value Properties of Entry:

- 503 • key
- 504 unique identifier for each *key-value pair*.
- 505 • removed
- 506 removal indicator of a *key-value pair*.

507 4.8 TableEntry

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

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

510 4.8.1 Value Properties of TableEntry

511 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

512 Descriptions for Value Properties of TableEntry:

- 513 • key
- 514 unique identifier for each *key-value pair*.
- 515 • removed
- 516 removal indicator of a *key-value pair*.

517 4.8.2 Part Properties of TableEntry

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

Part Property name	Multiplicity
Cell	0..*

Table 16: Part Properties of TableEntry

519 Descriptions for Part Properties of TableEntry:

- 520 • Cell
- 521 *key-value pair* published as part of a TableEntry.
- 522 See *Section 4.9 - Cell*.

523 4.9 Cell

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

525 4.9.0.1 Constraints for Cell Values

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

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

533 CellDefinition **MAY** provide type and units of a key.

534 4.9.1 Value Properties of Cell

535 *Table 17* lists the Value Properties of Cell.

Value Property name	Value Property type	Multiplicity
key	ID	1

Table 17: Value Properties of Cell

536 Descriptions for Value Properties of Cell:

- 537 • key
- 538 unique identifier for each *key-value pair*.

539 **5 Observation Types**

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

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

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

545 **5.1 Condition Types**

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

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

549 The type and subType (where applicable) for a `Condition` **MAY** be any of the type
550 and subType defined for `DataItems` with category as `SAMPLE` or category as
551 `EVENT` listed in the *Device Information Model*.

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

554 **5.1.1 Actuator**

555 indication of a fault associated with an actuator.

556 **5.1.2 Communications**

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

558 **5.1.3 DataRange**

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

561 **5.1.4 LogicProgram**

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

564 **5.1.5 MotionProgram**

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

567 **5.1.6 System**

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

571 **5.2 Event Types**

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

573 **5.2.1 ActivationCount**

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

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

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

578 **5.2.1.1 Subtypes of ActivationCount**

579 • `ABORTED`

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

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

601 **5.2.2 ActiveAxes**

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

604 **5.2.3 ActivePowerSource**

- 605 active energy source for the `Component`.
- 606 The value of `ActivePowerSource` **MUST** be string.

607 **5.2.4 ActuatorState**

608 operational state of an apparatus for moving or controlling a mechanism or system.

609 `ActuatorStateEnum` Enumeration:

610 • `ACTIVE`

611 Actuator is operating.

612 • `INACTIVE`

613 Actuator is not operating.

614 **5.2.5 AdapterSoftwareVersion**

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

616 **5.2.6 AdapterURI**

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

618 **5.2.7 <<deprecated>>Alarm**

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

620 **5.2.7.1 Value Properties of Alarm**

621 *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

622 Descriptions for Value Properties of Alarm:

- 623 • <<deprecated>> code
- 624 type of alarm.
- 625 • <<deprecated>> severity
- 626 severity of the alarm.
- 627 • <<deprecated>> nativeCode
- 628 native code for the piece of equipment.
- 629 • <<deprecated>> state
- 630 state of the alarm.
- 631 • <<deprecated>> lang
- 632 specifies the language of the alarm text.
- 633 See *Ref IETF RFC 4646* (<http://www.ietf.org/rfc/rfc4646.txt>).

634 5.2.8 <<deprecated>>AlarmLimit

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

636 **DEPRECATED** in *Version 2.5*. Replaced by `ALARM_LIMITS`.

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

638 `AlarmLimitResult` keys:

- 639 • `UpperLimit`
- 640 upper conformance boundary for a variable.

641 Note: immediate concern or action may be required.

642 The value of `UpperLimit` **MUST** be float.

643 • `UpperWarning`
 644 upper boundary indicating increased concern and supervision may be required.
 645 The value of `UpperWarning` **MUST** be float.

646 • `LowerWarning`
 647 lower boundary indicating increased concern and supervision may be required.
 648 The value of `LowerWarning` **MUST** be float.

649 • `LowerLimit`
 650 lower conformance boundary for a variable.

651 Note: immediate concern or action may be required.

652 The value of `LowerLimit` **MUST** be float.

653 **5.2.9 AlarmLimits**

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

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

656 `AlarmLimitResult` keys:

657 • `UpperLimit`
 658 upper conformance boundary for a variable.

659 Note: immediate concern or action may be required.

660 The value of `UpperLimit` **MUST** be float.

661 • `UpperWarning`
 662 upper boundary indicating increased concern and supervision may be required.
 663 The value of `UpperWarning` **MUST** be float.

- 664 • LowerWarning
665 lower boundary indicating increased concern and supervision may be required.
666 The value of LowerWarning **MUST** be float.
- 667 • LowerLimit
668 lower conformance boundary for a variable.

669 Note: immediate concern or action may be required.

670 The value of LowerLimit **MUST** be float.

671 **5.2.10 Application**

- 672 application on a Component.
- 673 A subType **MUST** always be specified.

674 **5.2.10.1 Subtypes of Application**

- 675 • INSTALL_DATE
676 date the hardware or software was installed.
677 The value of Application **MUST** be datetime. See *Section 6.1.5 - datetime*.
678 datetime Enumeration:
- 679 • LICENSE
680 license code to validate or activate the hardware or software.
- 681 • MANUFACTURER
682 corporate identity for the maker of the hardware or software.
- 683 • RELEASE_DATE
684 date the hardware or software was released for general use.
685 The value of Application **MUST** be datetime. See *Section 6.1.5 - datetime*.
686 datetime Enumeration:
- 687 • VERSION
688 version of the hardware or software.

689 5.2.11 AssetChanged

690 `assetId` of the *Asset* that has been added or changed.

691 5.2.11.1 Value Properties of AssetChanged

692 *Table 19* lists the Value Properties of `AssetChanged`.

Value Property name	Value Property type	Multiplicity
<code>assetType</code>	string	0..1
<code>hash</code>	string	0..1

Table 19: Value Properties of `AssetChanged`

693 Descriptions for Value Properties of `AssetChanged`:

- 694 • `assetType`
695 type of *Asset* changed. See *MTConnect Standard: Part 4.0 - Asset Information*
696 *Model* for details on the *Asset* model.
- 697 • `hash`
698 condensed message digest from a secure one-way hash function. *Ref FIPS PUB*
699 *180-4*

700 5.2.12 AssetCount

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

702 The value of `AssetCount` **MUST** be integer.

703 5.2.13 AssetRemoved

704 `assetId` of the *Asset* that has been removed.

705 5.2.13.1 Value Properties of AssetRemoved

706 *Table 20* lists the Value Properties of `AssetRemoved`.

Value Property name	Value Property type	Multiplicity
assetType	string	0..1
hash	string	0..1

Table 20: Value Properties of AssetRemoved

707 Descriptions for Value Properties of AssetRemoved:

- 708 • assetType
- 709 type of Asset removed. See *MTConnect Standard: Part 4.0 - Asset Information*
- 710 *Model* for details on the Asset model.
- 711 • hash
- 712 condensed message digest from a secure one-way hash function. *Ref FIPS PUB*
- 713 *180-4*

714 5.2.14 Availability

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

716 AvailabilityEnum Enumeration:

- 717 • AVAILABLE
- 718 data source is active and capable of providing data.
- 719 • UNAVAILABLE
- 720 data source is either inactive or not capable of providing data.

721 5.2.15 AxisCoupling

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

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

724 AxisCouplingEnum Enumeration:

- 725 • MASTER
- 726 axis is the master of the `CoupledAxes`.
- 727 • SLAVE
- 728 axis is a slave to the `CoupledAxes`.
- 729 • SYNCHRONOUS
- 730 axes are not physically connected to each other but are operating together in lock-
- 731 step.
- 732 • TANDEM
- 733 axes are physically connected to each other and operate as a single unit.

734 **5.2.16 AxisFeedrateOverride**

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

737 The value of `AxisFeedrateOverride` **MUST** be float.

738 **5.2.16.1 Subtypes of AxisFeedrateOverride**

- 739 • JOG
- 740 relating to momentary activation of a function or a movement.
- 741 **DEPRECATION WARNING:** May be deprecated in the future.
- 742 When the JOG subtype of `AxisFeedrateOverride` is applied, the resulting
- 743 commanded feedrate for the axis is limited to the value of the original JOG subtype
- 744 of the `AxisFeedrate` multiplied by the value of the JOG subtype of `AxisFeed-`
- 745 `rateOverride`.
- 746 • PROGRAMMED
- 747 directive value without offsets and adjustments.
- 748 • RAPID
- 749 performing an operation faster or in less time than nominal rate.

750 **5.2.17 AxisInterlock**

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

753 `AxisInterlockEnum` Enumeration:

754 • `ACTIVE`

755 axis lockout function is activated, power has been removed from the axis, and the
756 axis is allowed to move freely.

757 • `INACTIVE`

758 axis lockout function has not been activated, the axis may be powered, and the axis
759 is capable of being controlled by another component.

760 **5.2.18 AxisState**

761 state of a `Linear` or `Rotary` component representing an axis.

762 `AxisStateEnum` Enumeration:

763 • `HOME`

764 axis is in its home position.

765 • `PARKED`

766 axis has been moved to a fixed position and is being maintained in that position
767 either electrically or mechanically.

768 Action is required to release the axis from this position.

769 • `STOPPED`

770 axis is stopped.

771 • `TRAVEL`

772 axis is in motion.

773 5.2.19 BatteryState

774 present status of the battery.

775 BatteryStateEnum Enumeration:

776 • CHARGED

777 Component is at it's maximum rated charge level.

778 • CHARGING

779 Component's charge is increasing.

780 • DISCHARGED

781 Component is at it's minimum charge level.

782 • DISCHARGING

783 Component's charge is decreasing.

784 5.2.20 Block

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

786 5.2.21 BlockCount

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

789 The value of BlockCount **MUST** be integer.

790 5.2.22 CharacteristicPersistentId

791 Universally Unique Identifier (UUID) of the *characteristic*.

792 The value of CharacteristicPersistentId **MUST** be ID. See *Section 6.1.2 - ID*.

793 5.2.23 CharacteristicStatus

794 pass/fail result of the measurement.

795 CharacteristicStatusEnum Enumeration:

- 796 • BASIC_OR_THEORETIC_EXACT_DIMENSION
797 nominal provided without tolerance limits. *Ref QIF 3:2018 5.10.2.6*
- 798 • FAIL
799 measurement is not within acceptable tolerances.
- 800 • INDETERMINATE
801 measurement cannot be determined.
- 802 • NOT_ANALYZED
803 measurement cannot be evaluated.
- 804 • PASS
805 measurement is within acceptable tolerances.
- 806 • REWORK
807 failed, but acceptable constraints achievable by utilizing additional manufacturing
808 processes.
- 809 • SYSTEM_ERROR
810 measurement is indeterminate due to an equipment failure.
- 811 • UNDEFINED
812 status of measurement cannot be determined.

813 5.2.24 ChuckInterlock

814 state of an interlock function or control logic state intended to prevent the associated
815 Chuck component from being operated.

816 ChuckInterlockEnum Enumeration:

- 817 • ACTIVE
- 818 chuck cannot be unclamped.
- 819 • INACTIVE
- 820 chuck can be unclamped.

821 **5.2.24.1 Subtypes of ChuckInterlock**

- 822 • MANUAL_UNCLAMP
- 823 indication of the state of an operator controlled interlock that can inhibit the ability
- 824 to initiate an unclamp action of an electronically controlled chuck.
- 825 When `ChuckInterlockManualUnclamp` is ACTIVE, it is expected that a chuck
- 826 cannot be unclamped until `ChuckInterlockManualUnclamp` is set to INAC-
- 827 TIVE.

828 **5.2.25 ChuckState**

829 operating state of a mechanism that holds a part or stock material during a manufacturing
830 process.

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

833 `ChuckStateEnum` Enumeration:

- 834 • CLOSED
- 835 Chuck is closed to the point of a positive confirmation.
- 836 • OPEN
- 837 Chuck is open to the point of a positive confirmation.
- 838 • UNLATCHED
- 839 Chuck is not closed to the point of a positive confirmation and not open to the point
- 840 of a positive confirmation.
- 841 It is in an intermediate position.

842 5.2.26 ClockTime

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

844 The value of ClockTime **MUST** be datetime. See Section 6.1.5 - datetime.

845 5.2.27 <<deprecated>>Code

846 programmatic code being executed.

847 **DEPRECATED** in Version 1.1.

848 5.2.28 ComponentData

849 tabular Event that represents a Component where the EntryDefinition identi-
850 fies the Component and the CellDefinitions define the Component's observed
851 DataItems.

852 If the Component multiplicity can be determined, the device model **MUST** use a fixed
853 set of Components.

854 ComponentData **MUST** provide a DataItem Definition.

855 5.2.29 CompositionState

856 operating state of a mechanism represented by a Composition entity.

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

858 5.2.29.1 Subtypes of CompositionState

859 • ACTION

860 indication of the operating state of a mechanism.

861 CompositionStateActionEnum Enumeration:

862 – ACTIVE
 863 Composition is operating.

864 – INACTIVE
 865 Composition is not operating.

866 • LATERAL
 867 indication of the position of a mechanism that may move in a lateral direction.
 868 CompositionStateLateralEnum Enumeration:

869 – LEFT
 870 position of the Composition is oriented to the left to the point of a positive
 871 confirmation.

872 – RIGHT
 873 position of the Composition is oriented to the right to the point of a positive
 874 confirmation.

875 – TRANSITIONING
 876 position of the Composition is not oriented to the right to the point of a
 877 positive confirmation and is not oriented to the left to the point of a positive
 878 confirmation.
 879 It is in an intermediate position.

880 • MOTION
 881 indication of the open or closed state of a mechanism.
 882 CompositionStateMotionEnum Enumeration:

883 – CLOSED
 884 position of the Composition is closed to the point of a positive confirmation.

885 – OPEN
 886 position of the Composition is open to the point of a positive confirmation.

887 – UNLATCHED
 888 position of the Composition is not open to the point of a positive confirma-
 889 tion and is not closed to the point of a positive confirmation.
 890 It is in an intermediate position.

891 • SWITCHED
 892 indication of the activation state of a mechanism.
 893 CompositionStateSwitchedEnum Enumeration:

- 894 – OFF
- 895 activation state of the `Composition` is in an OFF condition, it is not operat-
- 896 ing, or it is not powered.
- 897 – ON
- 898 activation state of the `Composition` is in an ON condition, it is operating, or
- 899 it is powered.
- 900 • VERTICAL
- 901 indication of the position of a mechanism that may move in a vertical direction.
- 902 `CompositionStateVerticalEnum` Enumeration:
- 903 – DOWN
- 904 position of the `Composition` element is oriented in a downward direction to
- 905 the point of a positive confirmation.
- 906 – TRANSITIONING
- 907 position of the `Composition` element is not oriented in an upward direc-
- 908 tion to the point of a positive confirmation and is not oriented in a downward
- 909 direction to the point of a positive confirmation.
- 910 It is in an intermediate position.
- 911 – UP
- 912 position of the `Composition` element is oriented in an upward direction to
- 913 the point of a positive confirmation.

914 5.2.30 `ConnectionStatus`

- 915 status of the connection between an *adapter* and an *agent*.
- 916 `ConnectionStatusEnum` Enumeration:
- 917 • CLOSED
 - 918 no connection at all.
 - 919 • ESTABLISHED
 - 920 open connection.
 - 921 The normal state for the data transfer phase of the connection.
 - 922 • LISTEN
 - 923 *agent* is waiting for a connection request from an *adapter*.

924 5.2.31 <<deprecated>>ControlLimit

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

926 **DEPRECATED** in *Version 2.5*. Replaced by `CONTROL_LIMITS`.

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

928 `ControlLimitResult` keys:

929 • `UpperLimit`

930 upper conformance boundary for a variable.

931 Note: immediate concern or action may be required.

932 The value of `UpperLimit` **MUST** be float.

933 • `UpperWarning`

934 upper boundary indicating increased concern and supervision may be required.

935 The value of `UpperWarning` **MUST** be float.

936 • `Nominal`

937 ideal or desired value for a variable.

938 The value of `Nominal` **MUST** be float.

939 • `LowerWarning`

940 lower boundary indicating increased concern and supervision may be required.

941 The value of `LowerWarning` **MUST** be float.

942 • `LowerLimit`

943 lower conformance boundary for a variable.

944 Note: immediate concern or action may be required.

945 The value of `LowerLimit` **MUST** be float.

946 5.2.32 ControlLimits

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

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

949 ControlLimitResult keys:

950 • UpperLimit

951 upper conformance boundary for a variable.

952 Note: immediate concern or action may be required.

953 The value of UpperLimit **MUST** be float.

954 • UpperWarning

955 upper boundary indicating increased concern and supervision may be required.

956 The value of UpperWarning **MUST** be float.

957 • Nominal

958 ideal or desired value for a variable.

959 The value of Nominal **MUST** be float.

960 • LowerWarning

961 lower boundary indicating increased concern and supervision may be required.

962 The value of LowerWarning **MUST** be float.

963 • LowerLimit

964 lower conformance boundary for a variable.

965 Note: immediate concern or action may be required.

966 The value of LowerLimit **MUST** be float.

967 5.2.33 ControllerMode

968 current mode of the Controller component.

969 ControllerModeEnum Enumeration:

- 970 • AUTOMATIC
- 971 Controller is configured to automatically execute a program.
- 972 • EDIT
- 973 Controller is currently functioning as a programming device and is not capable
- 974 of executing an active program.
- 975 • <<deprecated>> FEED_HOLD
- 976 axes of the device are commanded to stop, but the spindle continues to function.
- 977 • MANUAL
- 978 Controller is not executing an active program.
- 979 It is capable of receiving instructions from an external source – typically an operator.
- 980 The Controller executes operations based on the instructions received from the
- 981 external source.
- 982 • MANUAL_DATA_INPUT
- 983 operator can enter a series of operations for the Controller to perform.
- 984 The Controller will execute this specific series of operations and then stop.
- 985 • SEMI_AUTOMATIC
- 986 Controller is operating in a mode that restricts the active program from process-
- 987 ing its next process step without operator intervention.

988 5.2.34 ControllerModeOverride

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

990 ControllerModeOverrideEnum Enumeration:

- 991 • OFF
- 992 ControllerModeOverride is in the OFF state and the mode override is inac-
- 993 tive.
- 994 • ON
- 995 ControllerModeOverride is in the ON state and the mode override is active.

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

997 5.2.34.1 Subtypes of ControllerModeOverride

- 998 • DRY_RUN
- 999 setting or operator selection used to execute a test mode to confirm the execution of
1000 machine functions.
- 1001 When DRY_RUN is ON, the equipment performs all of its normal functions, except
1002 no part or product is produced. If the equipment has a spindle, spindle operation is
1003 suspended.
- 1004 • MACHINE_AXIS_LOCK
- 1005 setting or operator selection that changes the behavior of the controller on a piece of
1006 equipment.
- 1007 When MACHINE_AXIS_LOCK is ON, program execution continues normally, but
1008 no equipment motion occurs.
- 1009 • OPTIONAL_STOP
- 1010 setting or operator selection that changes the behavior of the controller on a piece of
1011 equipment.
- 1012 The program execution is stopped after a specific program block is executed when
1013 OPTIONAL_STOP is ON.
- 1014 In the case of a G-Code program, a program block containing a M01 code designates
1015 the command for an OPTIONAL_STOP.
- 1016 Execution **MUST** change to OPTIONAL_STOP after a program block speci-
1017 fying an optional stop is executed and the ControllerModeOverride OP-
1018 TIONAL_STOP selection is ON.
- 1019 • SINGLE_BLOCK
- 1020 setting or operator selection that changes the behavior of the controller on a piece of
1021 equipment.
- 1022 Program execution is paused after each block of code is executed when SINGLE_BLOCK
1023 is ON.
- 1024 When SINGLE_BLOCK is ON, Execution **MUST** change to INTERRUPTED
1025 after completion of each block of code.
- 1026 • TOOL_CHANGE_STOP
- 1027 setting or operator selection that changes the behavior of the controller on a piece of
1028 equipment.
- 1029 Program execution is paused when a command is executed requesting a cutting tool
1030 to be changed.

1031 Execution **MUST** change to INTERRUPTED after completion of the command
1032 requesting a cutting tool to be changed and TOOL_CHANGE_STOP is ON.

1033 5.2.35 CoupledAxes

1034 set of associated axes.

1035 The value of CoupledAxes **MUST** be a list of string of size 0 . . *.

1036 5.2.36 CycleCount

1037 accumulation of the number of times a cyclic function has attempted to, or is planned to
1038 attempt to execute.

1039 The default subType of CycleCount is ALL.

1040 The value of CycleCount **MUST** be integer.

1041 5.2.36.1 Subtypes of CycleCount

1042 • ABORTED

1043 accumulation of actions or activities that were attempted, but terminated before they
1044 could be completed.

1045 • ALL

1046 accumulation of all actions, items, or activities being counted independent of the
1047 outcome.

1048 • BAD

1049 accumulation of actions, items, or activities being counted that do not conform to
1050 specification or expectation.

1051 • COMPLETE

1052 accumulation of actions, items, or activities that have been completed, independent
1053 of the outcome.

- 1054 • FAILED
- 1055 accumulation of actions or activities that were attempted, but failed to complete or
- 1056 resulted in an unexpected or unacceptable outcome.
- 1057 • GOOD
- 1058 accumulation of actions, items, or activities being counted that conform to specifi-
- 1059 cation or expectation.
- 1060 • REMAINING
- 1061 accumulation of actions, items, or activities yet to be counted.
- 1062 • TARGET
- 1063 goal of the operation or process.

1064 **5.2.37 DateCode**

- 1065 time and date code associated with a material or other physical item.
- 1066 The value of `DateCode` **MUST** be `datetime`. See *Section 6.1.5 - datetime*.

1067 **5.2.37.1 Subtypes of DateCode**

- 1068 • EXPIRATION
- 1069 time and date code relating to the expiration or end of useful life for a material or
- 1070 other physical item.
- 1071 • FIRST_USE
- 1072 time and date code relating the first use of a material or other physical item.
- 1073 • MANUFACTURE
- 1074 time and date code relating to the production of a material or other physical item.

1075 **5.2.38 DeactivationCount**

- 1076 accumulation of the number of times a function has attempted to, or is planned to attempt
- 1077 to, deactivate or cease.

1078 The default subType of DeactivationCount is ALL.

1079 The value of DeactivationCount **MUST** be integer.

1080 **5.2.38.1 Subtypes of DeactivationCount**

1081 • ABORTED

1082 accumulation of actions or activities that were attempted, but terminated before they
1083 could be completed.

1084 • ALL

1085 accumulation of all actions, items, or activities being counted independent of the
1086 outcome.

1087 • BAD

1088 accumulation of actions, items, or activities being counted that do not conform to
1089 specification or expectation.

1090 • COMPLETE

1091 accumulation of actions, items, or activities that have been completed, independent
1092 of the outcome.

1093 • FAILED

1094 accumulation of actions or activities that were attempted, but failed to complete or
1095 resulted in an unexpected or unacceptable outcome.

1096 • GOOD

1097 accumulation of actions, items, or activities being counted that conform to specifi-
1098 cation or expectation.

1099 • REMAINING

1100 accumulation of actions, items, or activities yet to be counted.

1101 • TARGET

1102 goal of the operation or process.

1103 **5.2.39 DeviceAdded**

1104 UUID of new device added to an *MTConnect Agent*.

1105 **5.2.39.1 Value Properties of DeviceAdded**

1106 *Table 21* lists the Value Properties of `DeviceAdded`.

Value Property name	Value Property type	Multiplicity
hash	string	0..1

Table 21: Value Properties of `DeviceAdded`

1107 Descriptions for Value Properties of `DeviceAdded`:

- 1108 • hash
- 1109 condensed message digest from a secure one-way hash function. *Ref FIPS PUB*
- 1110 *180-4*

1111 **5.2.40 DeviceChanged**

1112 UUID of the device whose *metadata* has changed.

1113 **5.2.40.1 Value Properties of DeviceChanged**

1114 *Table 22* lists the Value Properties of `DeviceChanged`.

Value Property name	Value Property type	Multiplicity
hash	string	0..1

Table 22: Value Properties of `DeviceChanged`

1115 Descriptions for Value Properties of `DeviceChanged`:

- 1116 • hash
- 1117 condensed message digest from a secure one-way hash function. *Ref FIPS PUB*
- 1118 *180-4*

1119 **5.2.41 DeviceRemoved**

1120 UUID of a device removed from an *MTConnect Agent*.

1121 5.2.41.1 Value Properties of DeviceRemoved

1122 *Table 23* lists the Value Properties of DeviceRemoved.

Value Property name	Value Property type	Multiplicity
hash	string	0..1

Table 23: Value Properties of DeviceRemoved

1123 Descriptions for Value Properties of DeviceRemoved:

- 1124 • hash
- 1125 condensed message digest from a secure one-way hash function. *Ref FIPS PUB*
- 1126 *180-4*

1127 5.2.42 DeviceUuid

1128 identifier of another piece of equipment that is temporarily associated with a component
1129 of this piece of equipment to perform a particular function.

1130 5.2.43 Direction

1131 direction of motion.

1132 <<deprecated>> DirectionEnum Enumeration:

- 1133 • <<deprecated>> CLOCKWISE
- 1134 clockwise rotation using the right-hand rule.
- 1135 • <<deprecated>> COUNTER_CLOCKWISE
- 1136 counter-clockwise rotation using the right-hand rule.
- 1137 • <<deprecated>> NEGATIVE
- 1138 • <<deprecated>> POSITIVE

1139 A subType **MUST** always be specified.

1140 **5.2.43.1 Subtypes of Direction**

- 1141 • LINEAR
- 1142 direction of motion of a linear motion.
- 1143 DirectionLinearEnum Enumeration:
- 1144 – NEGATIVE
- 1145 linear position is decreasing.
- 1146 – NONE
- 1147 no direction.
- 1148 – POSITIVE
- 1149 linear position is increasing.
- 1150 • ROTARY
- 1151 rotational direction of a rotary motion using the right hand rule convention.
- 1152 DirectionRotaryEnum Enumeration:
- 1153 – CLOCKWISE
- 1154 clockwise rotation using the right-hand rule.
- 1155 – COUNTER_CLOCKWISE
- 1156 counter-clockwise rotation using the right-hand rule.
- 1157 – NONE
- 1158 no direction.

1159 **5.2.44 DoorState**

- 1160 operational state of a Door component or composition element.
- 1161 DoorStateEnum Enumeration:
- 1162 • CLOSED
- 1163 Door is closed to the point of a positive confirmation.
- 1164 • OPEN
- 1165 Door is open to the point of a positive confirmation.

- 1166 • UNLATCHED
- 1167 Door is not closed to the point of a positive confirmation and not open to the point
- 1168 of a positive confirmation.
- 1169 It is in an intermediate position.

1170 5.2.45 EmergencyStop

- 1171 state of the emergency stop signal for a piece of equipment, controller path, or any other
- 1172 component or subsystem of a piece of equipment.

1173 EmergencyStopEnum Enumeration:

- 1174 • ARMED
- 1175 emergency stop circuit is complete and the piece of equipment, component, or com-
- 1176 position is allowed to operate.
- 1177 • TRIGGERED
- 1178 operation of the piece of equipment, component, or composition is inhibited.

1179 5.2.46 EndOfBar

- 1180 indication of whether the end of a piece of bar stock being feed by a bar feeder has been
- 1181 reached.

1182 EndOfBarEnum Enumeration:

- 1183 • NO
- 1184 EndOfBar has not been reached.
- 1185 • YES
- 1186 EndOfBar has been reached.

1187 The default subType of EndOfBar is PRIMARY.

1188 **5.2.46.1 Subtypes of EndOfBar**

- 1189 • AUXILIARY
- 1190 when multiple locations on a piece of bar stock are referenced as the indication for
- 1191 the EndOfBar, the additional location(s) **MUST** be designated as AUXILIARY
- 1192 indication(s) for the EndOfBar.
- 1193 • PRIMARY
- 1194 specific applications **MAY** reference one or more locations on a piece of bar stock
- 1195 as the indication for the EndOfBar.
- 1196 The main or most important location **MUST** be designated as the PRIMARY indica-
- 1197 tion for the EndOfBar.
- 1198 If no subType is specified, PRIMARY **MUST** be the default EndOfBar indica-
- 1199 tion.

1200 **5.2.47 EquipmentMode**

1201 indication that a piece of equipment, or a sub-part of a piece of equipment, is performing
1202 specific types of activities.

1203 EquipmentModeEnum Enumeration:

- 1204 • OFF
- 1205 equipment is not functioning in the mode designated by the subType.
- 1206 • ON
- 1207 equipment is functioning in the mode designated by the subType.

1208 A subType **MUST** always be specified.

1209 **5.2.47.1 Subtypes of EquipmentMode**

- 1210 • DELAY
- 1211 elapsed time of a temporary halt of action.

- 1212 • LOADED
- 1213 indication that the sub-parts of a piece of equipment are under load.
- 1214 Example: For traditional machine tools, this is an indication that the cutting tool is
- 1215 assumed to be engaged with the part.
- 1216 • OPERATING
- 1217 indication that the major sub-parts of a piece of equipment are powered or perform-
- 1218 ing any activity whether producing a part or product or not.
- 1219 Example: For traditional machine tools, this includes when the piece of equipment
- 1220 is WORKING or it is idle.
- 1221 • POWERED
- 1222 indication that primary power is applied to the piece of equipment and, as a min-
- 1223 imum, the controller or logic portion of the piece of equipment is powered and
- 1224 functioning or components that are required to remain on are powered.
- 1225 Example: Heaters for an extrusion machine that required to be powered even when
- 1226 the equipment is turned off.
- 1227 • WORKING
- 1228 indication that a piece of equipment is performing any activity, the equipment is
- 1229 active and performing a function under load or not.
- 1230 Example: For traditional machine tools, this includes when the piece of equipment
- 1231 is LOADED, making rapid moves, executing a tool change, etc.

1232 5.2.48 Execution

1233 operating state of a Component.

1234 ExecutionEnum Enumeration:

- 1235 • ACTIVE
- 1236 Component is actively executing an instruction.
- 1237 • FEED_HOLD
- 1238 motion of the active axes are commanded to stop at their current position.
- 1239 • INTERRUPTED
- 1240 Component suspends the execution of the program due to an external signal.
- 1241 Action is required to resume execution.

- 1242 • OPTIONAL_STOP
- 1243 command from the program has intentionally interrupted execution.
- 1244 The Component **MAY** have another state that indicates if the execution is inter-
- 1245 rupted or the execution ignores the interrupt instruction.
- 1246 • PROGRAM_COMPLETED
- 1247 program completed execution.
- 1248 • <<deprecated>> PROGRAM_OPTIONAL_STOP
- 1249 program has been intentionally optionally stopped using an M01 or similar code.
- 1250 **DEPRECATED** in *version 1.4* and replaced with OPTIONAL_STOP.
- 1251 • PROGRAM_STOPPED
- 1252 command from the program has intentionally interrupted execution.
- 1253 Action is required to resume execution.
- 1254 • READY
- 1255 Component is ready to execute instructions.
- 1256 It is currently idle.
- 1257 • STOPPED
- 1258 Component program is not READY to execute.
- 1259 • WAIT
- 1260 Component suspends execution while a secondary operation executes.
- 1261 Execution resumes automatically once the secondary operation completes.

1262 5.2.49 FeatureMeasurement

- 1263 tabular representation of assessing elements of a *feature*.
- 1264 FeatureMeasurement **MAY** include a *characteristic* in which case it **MAY** include a
- 1265 CHARACTERISTIC_STATUS.
- 1266 The Entry key **MUST** be one or more from the FeatureMeasurementResult
- 1267 keys.
- 1268 FeatureMeasurementResult keys:

- 1269 • MeasurementId
- 1270 identifier of this measurement.
- 1271 The value of MeasurementId **MUST** be ID. See *Section 6.1.2 - ID*.
- 1272 • FeaturePersistentId
- 1273 UUID of the feature.
- 1274 The value of FeaturePersistentId **MUST** be ID. See *Section 6.1.2 - ID*.
- 1275 • CharacteristicPersistentId
- 1276 UUID of the characteristic.
- 1277 The value of CharacteristicPersistentId **MUST** be ID. See *Section 6.1.2*
- 1278 *- ID*.
- 1279 • MeasurementType
- 1280 class of measurement being performed. *Ref QIF 3:2018 Section 6.3*
- 1281 Examples: POINT, RADIUS, ANGLE, LENGTH, etc.
- 1282 The value of MeasurementType **MUST** be string.
- 1283 • MeasurementValue
- 1284 measurement based on the measurement type.
- 1285 The value of MeasurementValue **MUST** be float.
- 1286 • MeasurementUnits
- 1287 engineering units of the measurement.
- 1288 The value of MeasurementUnits **MUST** be string.
- 1289 • CharacteristicStatus
- 1290 pass/fail result of the measurement.
- 1291 The value of CharacteristicStatus **MUST** be one of the Characteris-
- 1292 ticStatusEnum enumeration.
- 1293 CharacteristicStatusEnum Enumeration:
- 1294 – BASIC_OR_THEORETIC_EXACT_DIMENSION
- 1295 nominal provided without tolerance limits. *Ref QIF 3:2018 5.10.2.6*
- 1296 – FAIL
- 1297 measurement is not within acceptable tolerances.
- 1298 – INDETERMINATE
- 1299 measurement cannot be determined.

- 1300 – NOT_ANALYZED
- 1301 measurement cannot be evaluated.
- 1302 – PASS
- 1303 measurement is within acceptable tolerances.
- 1304 – REWORK
- 1305 failed, but acceptable constraints achievable by utilizing additional manufacturing processes.
- 1306
- 1307 – SYSTEM_ERROR
- 1308 measurement is indeterminate due to an equipment failure.
- 1309 – UNDEFINED
- 1310 status of measurement cannot be determined.
- 1311 • UncertaintyType
- 1312 method used to compute *standard uncertainty*.
- 1313 UncertaintyTypeEnum Enumeration:
- 1314 – COMBINED
- 1315 *combined standard uncertainty*.
- 1316 – MEAN
- 1317 *standard uncertainty* using arithmetic mean or average the observations. *Ref JCGM*
- 1318 *100:2008 4.2*
- 1319 • Uncertainty
- 1320 *uncertainty* specified by UNCERTAINTY_TYPE.
- 1321 The value of Uncertainty **MUST** be float.

1322 5.2.50 FeaturePersisitentId

- 1323 UUID of a *feature*. *Ref ISO 10303 AP 242/239*.
- 1324 The value of FeaturePersisitentId **MUST** be ID. See *Section 6.1.2 - ID*.

1325 5.2.51 Firmware

- 1326 embedded software of a Component .
- 1327 A subType **MUST** always be specified.

1328 **5.2.51.1 Subtypes of Firmware**

- 1329 • `INSTALL_DATE`
1330 date the hardware or software was installed.
1331 The value of `Firmware` **MUST** be `datetime`. See *Section 6.1.5 - datetime*.
1332 `datetime` Enumeration:
- 1333 • `LICENSE`
1334 license code to validate or activate the hardware or software.
- 1335 • `MANUFACTURER`
1336 corporate identity for the maker of the hardware or software.
- 1337 • `RELEASE_DATE`
1338 date the hardware or software was released for general use.
1339 The value of `Firmware` **MUST** be `datetime`. See *Section 6.1.5 - datetime*.
1340 `datetime` Enumeration:
- 1341 • `VERSION`
1342 version of the hardware or software.

1343 **5.2.52 FixtureId**

1344 identifier for the current workholding or part clamp in use by a piece of equipment.

1345 **5.2.53 FunctionalMode**

- 1346 current intended production status of the `Component`.
- 1347 `FunctionalModeEnum` Enumeration:
- 1348 • `MAINTENANCE`
1349 `Component` is not currently producing product.
1350 It is currently being repaired, waiting to be repaired, or has not yet been returned to
1351 a normal production status after maintenance has been performed.

- 1352 • PROCESS_DEVELOPMENT
- 1353 Component is being used to prove-out a new process, testing of equipment or
- 1354 processes, or any other active use that does not result in the production of product.
- 1355 • PRODUCTION
- 1356 Component is currently producing product, ready to produce product, or its current
- 1357 intended use is to be producing product.
- 1358 • SETUP
- 1359 Component is not currently producing product.
- 1360 It is being prepared or modified to begin production of product.
- 1361 • TEARDOWN
- 1362 Component is not currently producing product.
- 1363 Typically, it has completed the production of a product and is being modified or
- 1364 returned to a neutral state such that it may then be prepared to begin production of a
- 1365 different product.

1366 5.2.54 Hardness

- 1367 hardness of a material.
- 1368 The value of Hardness **MUST** be float.
- 1369 A subType **MUST** always be specified.

1370 5.2.54.1 Subtypes of Hardness

- 1371 • BRINELL
- 1372 scale to measure the resistance to deformation of a surface.
- 1373 • LEEB
- 1374 scale to measure the elasticity of a surface.
- 1375 • MOHS
- 1376 scale to measure the resistance to scratching of a surface.

- 1377 • ROCKWELL
- 1378 scale to measure the resistance to deformation of a surface.
- 1379 • SHORE
- 1380 scale to measure the resistance to deformation of a surface.
- 1381 • VICKERS
- 1382 scale to measure the resistance to deformation of a surface.

1383 5.2.55 Hardware

- 1384 hardware of a Component.
- 1385 A subType **MUST** always be specified.

1386 5.2.55.1 Subtypes of Hardware

- 1387 • INSTALL_DATE
- 1388 date the hardware or software was installed.
- 1389 The value of Hardware **MUST** be datetime. See *Section 6.1.5 - datetime*.
- 1390 datetime Enumeration:
- 1391 • LICENSE
- 1392 license code to validate or activate the hardware or software.
- 1393 • MANUFACTURER
- 1394 corporate identity for the maker of the hardware or software.
- 1395 • MODEL
- 1396 model info of the hardware or software.
- 1397 • RELEASE_DATE
- 1398 date the hardware or software was released for general use.
- 1399 The value of Hardware **MUST** be datetime. See *Section 6.1.5 - datetime*.
- 1400 datetime Enumeration:
- 1401 • VERSION
- 1402 version of the hardware or software.

1403 5.2.56 HostName

1404 name of the host computer supplying data.

1405 5.2.57 LeakDetect

1406 indication designating whether a leak has been detected.

1407 LeakDetectEnum Enumeration:

- 1408 • DETECTED
- 1409 leak is currently being detected.
- 1410 • NOT_DETECTED
- 1411 leak is currently not being detected.

1412 5.2.58 Library

1413 software library on a Component

1414 A subType **MUST** always be specified.

1415 5.2.58.1 Subtypes of Library

- 1416 • INSTALL_DATE
- 1417 date the hardware or software was installed.
- 1418 The value of Library **MUST** be datetime. See *Section 6.1.5 - datetime*.
- 1419 datetime Enumeration:
- 1420 • LICENSE
- 1421 license code to validate or activate the hardware or software.
- 1422 • MANUFACTURER
- 1423 corporate identity for the maker of the hardware or software.

- 1424 • RELEASE_DATE
- 1425 date the hardware or software was released for general use.
- 1426 The value of `Library` **MUST** be `datetime`. See *Section 6.1.5 - datetime*.
- 1427 `datetime` Enumeration:
- 1428 • VERSION
- 1429 version of the hardware or software.

1430 **5.2.59 <<deprecated>>Line**

- 1431 current line of code being executed.
- 1432 **DEPRECATED** in *Version 1.4.0*.

1433 **5.2.59.1 Subtypes of Line**

- 1434 • MAXIMUM
- 1435 maximum line number of the code being executed.
- 1436 • MINIMUM
- 1437 minimum line number of the code being executed.

1438 **5.2.60 LineLabel**

- 1439 identifier for a `Block` of code in a `Program`.

1440 **5.2.61 LineNumber**

- 1441 position of a block of program code within a control program.
- 1442 The value of `LineNumber` **MUST** be `integer`.

1443 **5.2.61.1 Subtypes of LineNumber**

- 1444 • ABSOLUTE
- 1445 position of a block of program code relative to the beginning of the control program.
- 1446 • INCREMENTAL
- 1447 position of a block of program code relative to the occurrence of the last LineLa-
- 1448 bel encountered in the control program.

1449 **5.2.62 LoadCount**

- 1450 accumulation of the number of times an operation has attempted to, or is planned to attempt
- 1451 to, load materials, parts, or other items.
- 1452 The default subType of LoadCount is ALL.
- 1453 The value of LoadCount **MUST** be integer.

1454 **5.2.62.1 Subtypes of LoadCount**

- 1455 • ABORTED
- 1456 accumulation of actions or activities that were attempted, but terminated before they
- 1457 could be completed.
- 1458 • ALL
- 1459 accumulation of all actions, items, or activities being counted independent of the
- 1460 outcome.
- 1461 • BAD
- 1462 accumulation of actions, items, or activities being counted that do not conform to
- 1463 specification or expectation.
- 1464 • COMPLETE
- 1465 accumulation of actions, items, or activities that have been completed, independent
- 1466 of the outcome.
- 1467 • FAILED
- 1468 accumulation of actions or activities that were attempted, but failed to complete or
- 1469 resulted in an unexpected or unacceptable outcome.

- 1470 • GOOD
- 1471 accumulation of actions, items, or activities being counted that conform to specifi-
- 1472 cation or expectation.
- 1473 • REMAINING
- 1474 accumulation of actions, items, or activities yet to be counted.
- 1475 • TARGET
- 1476 goal of the operation or process.

1477 5.2.63 LocationAddress

1478 structured information that allows the unambiguous determination of an object for pur-
 1479 poses of identification and location. *Ref ISO 19160-4:2017*

1480 The `Entry` key **MUST** be one or more from the `LocationAddressResult` keys.

1481 `LocationAddressResult` keys:

- 1482 • `GivenName`
- 1483 element specifying the name used to distinguish between persons having the same
- 1484 surname(s) and who may have access to a particular delivery point.
- 1485 The value of `GivenName` **MUST** be `string`.
- 1486 • `Surname`
- 1487 element which identifies the family or parentage of an individual.
- 1488 The value of `Surname` **MUST** be `string`.
- 1489 • `Qualification`
- 1490 element indicating an individual's professional or academic qualification or rank in a
- 1491 professional group or society (e.g. PhD, Fellow of the Royal Society, FRS, Barrister
- 1492 at Law).
- 1493 The value of `Qualification` **MUST** be `string`.
- 1494 • `NameQualifier`
- 1495 element used in some countries to distinguish between persons with the same sur-
- 1496 name(s) who have similar given names or initials (e.g.III, Senior, the Third.).
- 1497 The value of `NameQualifier` **MUST** be `string`.

- 1498 • `OrganizationName`
1499 element giving the official name, the registered business name or other official des-
1500 ignation of an organization.
1501 The value of `OrganizationName` **MUST** be string.
- 1502 • `LegalStatus`
1503 element indicating the formal registration of an organization (e.g. GmbH, Inc., Ltd.).
1504 The value of `LegalStatus` **MUST** be string.
- 1505 • `OrganizationUnit`
1506 element identifying a subdivision of an organization.
1507 The value of `OrganizationUnit` **MUST** be string.
- 1508 • `Building`
1509 element identifying the number or name and type of the edifice or construction in or
1510 adjacent to which a delivery point is located.
1511 The value of `Building` **MUST** be string.
- 1512 • `Floor`
1513 element indicating the floor or level on which a delivery point is located in a multi-
1514 story building.
1515 The value of `Floor` **MUST** be string.
- 1516 • `Door`
1517 element indicating the apartment, room or office in, at or adjacent to which a delivery
1518 point, situated within a building, is located.
1519 The value of `Door` **MUST** be string.
- 1520 • `Thoroughfare`
1521 element which identifies the road or part of a road or other access route along which
1522 a delivery point can be accessed, either directly or via a secondary or tertiary road
1523 or access route.
1524 The value of `Thoroughfare` **MUST** be string.
- 1525 • `PremiseIdentifier`
1526 element designating the area or the object on an area, adjacent to thoroughfare, in
1527 which the delivery point or delivery point access is located.
1528 The value of `PremiseIdentifier` **MUST** be string.

- 1529 • **Town**
- 1530 element indicating the name of the populated place in which a delivery point is
- 1531 located, or the populated.
- 1532 The value of **Town** **MUST** be `string`.
- 1533 • **Region**
- 1534 element specifying the geographic or administrative area of the country in which the
- 1535 town is situated.
- 1536 The value of **Region** **MUST** be `string`.
- 1537 • **District**
- 1538 element indicating the name of the area within or adjacent to the town in which a
- 1539 delivery point is located, or via which it is accessed.
- 1540 The value of **District** **MUST** be `string`.
- 1541 • **CountryName**
- 1542 element designating the country, dependency or area of geopolitical interest, in
- 1543 which a delivery point is located or via which the delivery point is accessed.
- 1544 The value of **CountryName** **MUST** be `string`.
- 1545 • **CountryCode**
- 1546 interest, in which a delivery point is located or via which the delivery point is ac-
- 1547 cessed.
- 1548 The value of **CountryCode** **MUST** be `string`.
- 1549 • **PostCode**
- 1550 element designating the code used for the sorting of mail.
- 1551 The value of **PostCode** **MUST** be `string`.

1552 **5.2.64 LocationNarrative**

1553 textual description of the location of an object or activity.

1554 **5.2.65 LocationSpatialGeographic**

1555 absolute geographic location defined by two coordinates, longitude and latitude and an
1556 elevation.

1557 The Entry key **MUST** be one or more from the LocationSpatialGeographi-
1558 cResult keys.

1559 LocationSpatialGeographicResult keys:

- 1560 • Longitude
1561 geographic longitude.
1562 The value of Longitude **MUST** be DEGREE.
- 1563 • Latitude
1564 geographic latitude.
1565 The value of Latitude **MUST** be DEGREE.
- 1566 • Altitude
1567 height relative to a reference.
1568 The value of Altitude **MUST** be METER.

1569 5.2.66 LockState

1570 state or operating mode of a Lock.

1571 LockStateEnum Enumeration:

- 1572 • LOCKED
1573 mechanism is engaged and preventing the associated Component from being opened
1574 or operated.
- 1575 • UNLOCKED
1576 mechanism is disengaged and the associated Component is able to be opened or
1577 operated.

1578 5.2.67 MTConnectVersion

1579 reference version of the MTConnect Standard supported by the *adapter*.

1580 5.2.68 MaintenanceList

1581 actions or activities to be performed in support of a piece of equipment.

1582 If Interval key is not provided, it is assumed ABSOLUTE.

1583 If Direction key is not provided, it is assumed UP.

1584 If Units key is not provided, it is assumed to be COUNT.

1585 The Entry key **MUST** be one or more from the MaintenanceListResult keys.

1586 MaintenanceListResult keys:

1587 • Value

1588 current interval value of the activity.

1589 The value of Value **MUST** be float.

1590 • Interval

1591 interval of the value observed.

1592 MaintenanceListIntervalEnum Enumeration:

1593 – ABSOLUTE

1594 – INCREMENTAL

1595 • NextServiceDate

1596 next date/time stamp that maintenance should be performed.

1597 The value of NextServiceDate **MUST** be datetime. See Section 6.1.5 -
1598 *datetime*.

1599 • Reset

1600 last date/time stamp of the observation was reset.

1601 The value of Reset **MUST** be datetime. See Section 6.1.5 - *datetime*.

1602 • Severity

1603 level of severity on a scale of 1-10.

1604 The value of Severity **MUST** be integer.

- 1605 • `Direction`
- 1606 direction of the value observed.
- 1607 `MaintenanceListDirectionEnum` Enumeration:
- 1608 – `DOWN`
- 1609 – `UP`
- 1610 • `Name`
- 1611 identifier of the maintenance activity.
- 1612 The value of `Name` **MUST** be `string`.
- 1613 • `LastServiceDate`
- 1614 last date/time stamp that maintenance was performed.
- 1615 The value of `LastServiceDate` **MUST** be `datetime`. See *Section 6.1.5 -*
- 1616 *datetime*.
- 1617 • `Units`
- 1618 same as `units`. See *MTConnect Standard: Part 2.0 - Device Information Model*.
- 1619 The value of `Units` **MUST** be one of the `UnitEnum` enumeration.
- 1620 • `Target`
- 1621 target value of the next maintenance.
- 1622 The value of `Target` **MUST** be `float`.

1623 5.2.69 Material

1624 identifier of a material used or consumed in the manufacturing process.

1625 5.2.70 MaterialLayer

1626 identifies the layers of material applied to a part or product as part of an additive manufac-
1627 turing process.

1628 The value of `MaterialLayer` **MUST** be `integer`.

1629 **5.2.70.1 Subtypes of MaterialLayer**

- 1630 • ACTUAL
- 1631 measured or reported value of an observation.
- 1632 • TARGET
- 1633 goal of the operation or process.

1634 **5.2.71 MeasurementType**

1635 class of measurement being performed. *Ref QIF 3:2018 Section 6.3*

1636 Examples: POINT, RADIUS, ANGLE, LENGTH, etc.

1637 **5.2.72 MeasurementUnits**

1638 engineering units of the measurement.

1639 **5.2.73 MeasurementValue**

1640 measurement based on the measurement type.

1641 The value of MeasurementValue **MUST** be double.

1642 **5.2.74 Message**

1643 information to be transferred from a piece of equipment to a client software application.

1644 **5.2.74.1 Value Properties of Message**

1645 *Table 24* lists the Value Properties of Message.

Value Property name	Value Property type	Multiplicity
nativeCode	string	0..1

Table 24: Value Properties of Message

1646 Descriptions for Value Properties of Message:

- 1647 • nativeCode
- 1648 control system local identification of the information being transferred.

1649 5.2.75 Network

1650 network details of a Component.

1651 A subType **MUST** always be specified.

1652 5.2.75.1 Subtypes of Network

- 1653 • GATEWAY
- 1654 Gateway for the component network.
- 1655 • IPV4_ADDRESS
- 1656 IPV4 network address of the component.
- 1657 • IPV6_ADDRESS
- 1658 IPV6 network address of the component.
- 1659 • MAC_ADDRESS
- 1660 Media Access Control Address.
- 1661 The unique physical address of the network hardware.
- 1662 • SUBNET_MASK
- 1663 SubNet mask for the component network.
- 1664 • VLAN_ID
- 1665 layer2 Virtual Local Network (VLAN) ID for the component network.

- 1666 • WIRELESS
- 1667 identifies whether the connection type is wireless.
- 1668 NetworkWirelessEnum Enumeration:
- 1669 – NO
- 1670 – YES

1671 5.2.76 NetworkPort

- 1672 number of the TCP/IP or UDP/IP port for the connection endpoint.
- 1673 The value of NetworkPort **MUST** be integer.

1674 5.2.77 OperatingMode

- 1675 state of Component or Composition that describes the automatic or manual operation
- 1676 of the entity.
- 1677 OperatingModeEnum Enumeration:
- 1678 • AUTOMATIC
- 1679 automatically execute instructions from a recipe or program.
- 1680 Note: Setpoint comes from a recipe.
- 1681 • MANUAL
- 1682 execute instructions from an external agent or person.
- 1683 Note 1 to entry: Valve or switch is manipulated by an agent/person.
- 1684 Note 2 to entry: Direct control of the PID output. % of the range: A user
- 1685 manually sets the % output, not the setpoint.
- 1686 • SEMI_AUTOMATIC
- 1687 executes a single instruction from a recipe or program.
- 1688 Note 1 to entry: Setpoint is entered and fixed, but the PID is controlling.
- 1689 Note 2 to entry: Still goes through the PID control system.
- 1690 Note 3 to entry: Manual fixed entry from a recipe.

1691 **5.2.78 OperatingSystem**

1692 Operating System (OS) of a Component.

1693 A subType **MUST** always be specified.

1694 **5.2.78.1 Subtypes of OperatingSystem**

1695 • INSTALL_DATE

1696 date the hardware or software was installed.

1697 The value of OperatingSystem **MUST** be datetime. See *Section 6.1.5 -*
1698 *datetime*.

1699 datetime Enumeration:

1700 • LICENSE

1701 license code to validate or activate the hardware or software.

1702 • MANUFACTURER

1703 corporate identity for the maker of the hardware or software.

1704 • RELEASE_DATE

1705 date the hardware or software was released for general use.

1706 The value of OperatingSystem **MUST** be datetime. See *Section 6.1.5 -*
1707 *datetime*.

1708 datetime Enumeration:

1709 • VERSION

1710 version of the hardware or software.

1711 **5.2.79 OperatorId**

1712 identifier of the person currently responsible for operating the piece of equipment.

1713 **5.2.80 PalletId**

1714 identifier for a pallet.

1715 **5.2.81 PartCount**

1716 aggregate count of parts.

1717 The value of PartCount **MUST** be integer.

1718 The default subType of PartCount is ALL.

1719 **5.2.81.1 Subtypes of PartCount**

1720 • ABORTED

1721 accumulation of actions or activities that were attempted, but terminated before they
1722 could be completed.

1723 • ALL

1724 accumulation of all actions, items, or activities being counted independent of the
1725 outcome.

1726 • BAD

1727 accumulation of actions, items, or activities being counted that do not conform to
1728 specification or expectation.

1729 • COMPLETE

1730 accumulation of actions, items, or activities that have been completed, independent
1731 of the outcome.

1732 • FAILED

1733 accumulation of actions or activities that were attempted, but failed to complete or
1734 resulted in an unexpected or unacceptable outcome.

1735 • GOOD

1736 accumulation of actions, items, or activities being counted that conform to specifi-
1737 cation or expectation.

1738 • REMAINING

1739 accumulation of actions, items, or activities yet to be counted.

1740 • TARGET

1741 goal of the operation or process.

1742 **5.2.82 PartCountType**

1743 interpretation of PART_COUNT.

1744 PartCountTypeEnum Enumeration:

- 1745 • BATCH
- 1746 pre-specified group of items.
- 1747 • EACH
- 1748 count is of individual items.

1749 **5.2.83 PartDetect**

1750 indication designating whether a part or work piece has been detected or is present.

1751 PartDetectEnum Enumeration:

- 1752 • NOT_PRESENT
- 1753 part or work piece is not detected or is not present.
- 1754 • PRESENT
- 1755 part or work piece is detected or is present.

1756 **5.2.84 PartGroupId**

1757 identifier given to a collection of individual parts.

1758 If no subType is specified, UUID is default.

1759 The default subType of PartGroupId is UUID.

1760 **5.2.84.1 Subtypes of PartGroupId**

- 1761 • BATCH
- 1762 identifier that references a group of parts produced in a batch.

- 1763 • HEAT_TREAT
- 1764 identifier used to reference a material heat number.
- 1765 • LOT
- 1766 identifier that references a group of parts tracked as a lot.
- 1767 • RAW_MATERIAL
- 1768 material that is used to produce parts.
- 1769 • UUID
- 1770 universally unique identifier as specified in ISO 11578 or RFC 4122.

1771 **5.2.85 PartId**

1772 identifier of a part in a manufacturing operation.

1773 **5.2.86 PartIndex**

1774 sequence of a part in a group of parts.

1775 The value of `PartIndex` **MUST** be integer.

1776 **5.2.87 PartKindId**

1777 identifier given to link the individual occurrence to a class of parts, typically distinguished
1778 by a particular part design.

1779 If no `subType` is specified, `UUID` is default.

1780 The default `subType` of `PartKindId` is `UUID`.

1781 **5.2.87.1 Subtypes of PartKindId**

- 1782 • PART_FAMILY
- 1783 identifier given to a group of parts having similarities in geometry, manufacturing
1784 process, and/or functions.

- 1785 • PART_NAME
- 1786 word or set of words by which a part is known, addressed, or referred to.
- 1787 • PART_NUMBER
- 1788 identifier of a particular part design or model.
- 1789 • UUID
- 1790 universally unique identifier as specified in ISO 11578 or RFC 4122.

1791 5.2.88 <<deprecated>>PartNumber

- 1792 identifier of a part or product moving through the manufacturing process.
- 1793 **DEPRECATED** in *Version 1.7*. PART_NUMBER is now a subType of PART_KIND_ID.

1794 5.2.89 PartProcessingState

- 1795 particular condition of the part occurrence at a specific time.
- 1796 PartProcessingStateEnum Enumeration:
- 1797 • IN_PROCESS
 - 1798 part occurrence is actively being processed.
 - 1799 • IN_TRANSIT
 - 1800 part occurrence is being transported to its destination.
 - 1801 • NEEDS_PROCESSING
 - 1802 part occurrence is not actively being processed, but the processing has not ended.
 - 1803 Processing requirements exist that have not yet been fulfilled. This is the default
 - 1804 entry state when the part occurrence is originally received. In some cases, the part
 - 1805 occurrence may return to this state while it waits for additional processing to be
 - 1806 performed.
 - 1807 • PROCESSING_ENDED
 - 1808 part occurrence is no longer being processed.
 - 1809 A general state when the reason for termination is unknown.

- 1810 • PROCESSING_ENDED_ABORTED
- 1811 processing of the part occurrence has come to a premature end.
- 1812 • PROCESSING_ENDED_COMPLETE
- 1813 part occurrence has completed processing successfully.
- 1814 • PROCESSING_ENDED_LOST
- 1815 terminal state when the part occurrence has been removed from the equipment by
- 1816 an external entity and it no longer exists at the equipment.
- 1817 • PROCESSING_ENDED_REJECTED
- 1818 part occurrence has been processed completely. However, the processing may have
- 1819 a problem.
- 1820 • PROCESSING_ENDED_SKIPPED
- 1821 part occurrence has been skipped for processing on the piece of equipment.
- 1822 • PROCESSING_ENDED_STOPPED
- 1823 process has been stopped during the processing.
- 1824 The part occurrence will require special treatment.
- 1825 • TRANSIT_COMPLETE
- 1826 part occurrence has been placed at its designated destination.
- 1827 • WAITING_FOR_TRANSIT
- 1828 part occurrence is waiting for transit.

1829 5.2.90 PartStatus

1830 state or condition of a part.

1831 If unique identifier is given, part status is for that individual. If group identifier is given
1832 without a unique identifier, then the status is assumed to be for the whole group.

1833 PartStatusEnum Enumeration:

- 1834 • FAIL
- 1835 part does not conform to some given requirements.
- 1836 • PASS
- 1837 part conforms to given requirements.

1838 5.2.91 PartUniqueId

1839 identifier given to a distinguishable, individual part.

1840 If no subType is specified, UUID is default.

1841 The default subType of PartUniqueId is UUID.

1842 5.2.91.1 Subtypes of PartUniqueId

1843 • RAW_MATERIAL

1844 material that is used to produce parts.

1845 • SERIAL_NUMBER

1846 serial number that uniquely identifies a specific part.

1847 • UUID

1848 universally unique identifier as specified in ISO 11578 or RFC 4122.

1849 5.2.92 PathFeedrateOverride

1850 value of a signal or calculation issued to adjust the feedrate for the axes associated with a
1851 Path component that may represent a single axis or the coordinated movement of multiple
1852 axes.

1853 The value of PathFeedrateOverride **MUST** be float.

1854 5.2.92.1 Subtypes of PathFeedrateOverride

1855 • JOG

1856 relating to momentary activation of a function or a movement.

1857 **DEPRECATION WARNING:** May be deprecated in the future.

1858 • PROGRAMMED

1859 directive value without offsets and adjustments.

1860 • RAPID

1861 performing an operation faster or in less time than nominal rate.

1862 5.2.93 PathMode

1863 describes the operational relationship between a `Path` entity and another `Path` entity for
 1864 pieces of equipment comprised of multiple logical groupings of controlled axes or other
 1865 logical operations.

1866 `PathModeEnum` Enumeration:

- 1867 • `INDEPENDENT`
 1868 path is operating independently and without the influence of another path.
- 1869 • `MASTER`
 1870 path provides information or state values that influences the operation of other `DataItem`
 1871 of similar type.
- 1872 • `MIRROR`
 1873 axes associated with the path are mirroring the motion of the `MASTER` path.
- 1874 • `SYNCHRONOUS`
 1875 physical or logical parts which are not physically connected to each other but are
 1876 operating together.

1877 5.2.94 PowerState

1878 indication of the status of the source of energy for an entity to allow it to perform its
 1879 intended function or the state of an enabling signal providing permission for the entity to
 1880 perform its functions.

1881 `PowerStateEnum` Enumeration:

- 1882 • `OFF`
 1883 source of energy for an entity or the enabling signal providing permission for the
 1884 entity to perform its function(s) is not present or is disconnected.
- 1885 • `ON`
 1886 source of energy for an entity or the enabling signal providing permission for the
 1887 entity to perform its function(s) is present and active.

1888 **5.2.94.1 Subtypes of PowerState**

- 1889 • CONTROL
- 1890 state of the enabling signal or control logic that enables or disables the function or
- 1891 operation of the entity.
- 1892 • LINE
- 1893 state of the power source for the entity.

1894 **5.2.95 <<deprecated>>PowerStatus**

1895 status of the Component.

1896 **DEPRECATED** in *Version 1.1.0*.

1897 <<deprecated>> PowerStatusEnum Enumeration:

- 1898 • <<deprecated>> OFF
- 1899 • <<deprecated>> ON

1900 **5.2.96 ProcessAggregateId**

1901 identifier given to link the individual occurrence to a group of related occurrences, such as

1902 a process step in a process plan.

1903 **5.2.96.1 Subtypes of ProcessAggregateId**

- 1904 • ORDER_NUMBER
- 1905 identifier of the authorization of the process occurrence. Synonyms include “job id”,
- 1906 “work order”.
- 1907 • PROCESS_PLAN
- 1908 identifier of the process plan that this occurrence belongs to. Synonyms include
- 1909 “routing id”, “job id”.

- 1910 • PROCESS_STEP
- 1911 identifier of the step in the process plan that this occurrence corresponds to. Syn-
- 1912 onyms include “operation id”.

1913 **5.2.97 ProcessKindId**

1914 identifier given to link the individual occurrence to a class of processes or process defini-
 1915 tion.

1916 **5.2.97.1 Subtypes of ProcessKindId**

- 1917 • ISO_STEP_EXECUTABLE
- 1918 reference to a ISO 10303 Executable.
- 1919 • PROCESS_NAME
- 1920 word or set of words by which a process being executed (process occurrence) by the
- 1921 device is known, addressed, or referred to.
- 1922 • UUID
- 1923 universally unique identifier as specified in ISO 11578 or RFC 4122.

1924 **5.2.98 ProcessOccurrenceId**

1925 identifier of a process being executed by the device.

1926 **5.2.98.1 Subtypes of ProcessOccurrenceId**

- 1927 • ACTIVITY
- 1928 phase or segment of a recipe or program.
- 1929 • OPERATION
- 1930 step of a discrete manufacturing process.
- 1931 • RECIPE
- 1932 process as part of product production; can be a subprocess of a larger process.

- 1933 • SEGMENT
- 1934 phase of a recipe process.

1935 **5.2.99 ProcessState**

1936 particular condition of the process occurrence at a specific time.

1937 ProcessStateEnum Enumeration:

- 1938 • ABORTED
- 1939 process occurrence has come to a premature end and cannot be resumed.
- 1940 • ACTIVE
- 1941 process occurrence is actively executing.
- 1942 • COMPLETE
- 1943 process occurrence is now finished.
- 1944 • INITIALIZING
- 1945 device is preparing to execute the process occurrence.
- 1946 • INTERRUPTED
- 1947 process occurrence has been stopped and may be resumed.
- 1948 • READY
- 1949 process occurrence is ready to be executed.

1950 **5.2.100 ProcessTime**

1951 time and date associated with an activity or event.

1952 A subType **MUST** always be specified.

1953 **5.2.100.1 Subtypes of ProcessTime**

- 1954 • COMPLETE
- 1955 time and date associated with the completion of an activity or event.

- 1956 • START
- 1957 boundary when an activity or an event commences.
- 1958 • TARGET_COMPLETION
- 1959 projected time and date associated with the end or completion of an activity or event.

1960 **5.2.101 Program**

1961 name of the logic or motion program being executed by the Controller component.

1962 **5.2.101.1 Subtypes of Program**

- 1963 • ACTIVE
- 1964 identity of the logic or motion program currently executing.
- 1965 • ACTIVITY
- 1966 phase or segment of a recipe or program.
- 1967 • MAIN
- 1968 identity of the primary logic or motion program currently being executed.
- 1969 It is the starting nest level in a call structure and may contain calls to sub programs.
- 1970 • OPERATION
- 1971 step of a discrete manufacturing process.
- 1972 • RECIPE
- 1973 process as part of product production; can be a subprocess of a larger process.
- 1974 • SCHEDULE
- 1975 identity of a control program that is used to specify the order of execution of other
- 1976 programs.
- 1977 • SEGMENT
- 1978 phase of a recipe process.

1979 **5.2.102 ProgramComment**

1980 comment or non-executable statement in the control program.

1981 **5.2.102.1 Subtypes of ProgramComment**

1982 • ACTIVE

1983 identity of the logic or motion program currently executing.

1984 • MAIN

1985 identity of the primary logic or motion program currently being executed.

1986 It is the starting nest level in a call structure and may contain calls to sub programs.

1987 • SCHEDULE

1988 identity of a control program that is used to specify the order of execution of other
1989 programs.

1990 **5.2.103 ProgramEdit**

1991 indication of the status of the Controller components program editing mode.

1992 A program may be edited while another is executed.

1993 ProgramEditEnum Enumeration:

1994 • ACTIVE

1995 Controller is in the program edit mode.

1996 • NOT_READY

1997 Controller is being inhibited by a function from entering the program edit mode.

1998 • READY

1999 Controller is capable of entering the program edit mode and no function is
2000 inhibiting a change to that mode.

2001 5.2.104 ProgramEditName

2002 name of the program being edited.

2003 This is used in conjunction with `ProgramEdit` when in `ACTIVE` state.

2004 5.2.105 ProgramHeader

2005 non-executable header section of the control program.

2006 The default `subType` of `ProgramHeader` is `MAIN`.

2007 5.2.105.1 Subtypes of ProgramHeader

2008 • `ACTIVE`

2009 identity of the logic or motion program currently executing.

2010 • `MAIN`

2011 identity of the primary logic or motion program currently being executed.

2012 It is the starting nest level in a call structure and may contain calls to sub programs.

2013 • `SCHEDULE`

2014 identity of a control program that is used to specify the order of execution of other
2015 programs.

2016 5.2.106 ProgramLocation

2017 URI for the source file associated with `Program`.

2018 5.2.106.1 Subtypes of ProgramLocation

2019 • `ACTIVE`

2020 identity of the logic or motion program currently executing.

- 2021 • MAIN
- 2022 identity of the primary logic or motion program currently being executed.
- 2023 It is the starting nest level in a call structure and may contain calls to sub programs.
- 2024 • SCHEDULE
- 2025 identity of a control program that is used to specify the order of execution of other
- 2026 programs.

2027 **5.2.107 ProgramLocationType**

- 2028 defines whether the logic or motion program defined by `Program` is being executed from
- 2029 the local memory of the controller or from an outside source.

2030 `ProgramLocationTypeEnum` Enumeration:

- 2031 • EXTERNAL
- 2032 not managed by the controller.
- 2033 • LOCAL
- 2034 managed by the controller.

2035 **5.2.107.1 Subtypes of ProgramLocationType**

- 2036 • ACTIVE
- 2037 identity of the logic or motion program currently executing.
- 2038 • MAIN
- 2039 identity of the primary logic or motion program currently being executed.
- 2040 It is the starting nest level in a call structure and may contain calls to sub programs.
- 2041 • SCHEDULE
- 2042 identity of a control program that is used to specify the order of execution of other
- 2043 programs.

2044 **5.2.108 ProgramNestLevel**

2045 indication of the nesting level within a control program that is associated with the code or
2046 instructions that is currently being executed.

2047 If an initial value is not defined, the nesting level associated with the highest or initial
2048 nesting level of the program **MUST** default to zero (0).

2049 The value of ProgramNestLevel **MUST** be integer.

2050 **5.2.109 RotaryMode**

2051 current operating mode for a Rotary type axis.

2052 RotaryModeEnum Enumeration:

- 2053 • CONTOUR
2054 position of the axis is being interpolated.
- 2055 • INDEX
2056 axis is configured to index.
- 2057 • SPINDLE
2058 axis is functioning as a spindle.

2059 **5.2.110 RotaryVelocityOverride**

2060 percentage change to the velocity of the programmed velocity for a Rotary axis.

2061 This command represents a percentage change to the velocity calculated by a logic or
2062 motion program or set by a switch for a Rotary type axis.

2063 The value of RotaryVelocityOverride **MUST** be float.

2064 **5.2.111 Rotation**

2065 three space angular displacement of an object or coordinate system relative to a *cartesian*
2066 *coordinate system*.

2067 The units of Rotation **MUST** be DEGREE_3D.

2068 The value of Rotation **MUST** be float3d. See *Section 6.1.20* - *<<hasFormat-SpecificRepresentation>>float3d*.

2070 5.2.112 SensorAttachment

2071 *attachment* between a sensor and an entity.

2072 The Entry key **MUST** be one or more from the SensorAttachmentResult keys.

2073 SensorAttachmentResult keys:

- 2074 • SensorId
- 2075 identity of a sensor used to observe some measurement of an item.
- 2076 The value of SensorId **MUST** be ID. See *Section 6.1.2* - *ID*.

2077 5.2.113 SensorState

2078 detection result of a sensor.

2079 The default subType of SensorState is BINARY.

2080 5.2.113.1 Subtypes of SensorState

- 2081 • BINARY
- 2082 detection result of a sensor.
- 2083 where the state is observed as a binary data type.
- 2084 The value of SensorState **MUST** be binary. See *Section 6.1.17* - *binary*.
- 2085 binary Enumeration:
- 2086 • BOOLEAN
- 2087 detection result of a sensor.
- 2088 where the state is observed as a boolean data type.
- 2089 The value of SensorState **MUST** be boolean.
- 2090 boolean Enumeration:

- 2091 • DETECT
- 2092 detection result of a sensor.
- 2093 where the state is indicated by the presence or existence of something.
- 2094 The value of `SensorState` **MUST** be one of the `CompositionStateSwitchedE-`
- 2095 `num` enumeration.
- 2096 • ENUMERATED
- 2097 detection result of a sensor.
- 2098 where the state is observed as a set containing a restricted number of discrete values
- 2099 where each discrete value is named and unique. *Ref ISO 21961:2003, 013*
- 2100 The value of `SensorState` **MUST** be integer.
- 2101 integer Enumeration:

2102 **5.2.114 SerialNumber**

2103 serial number associated with a Component, Asset, or Device.

2104 **5.2.115 <<deprecated>>SpecificationLimit**

2105 set of limits defining a range of values designating acceptable performance for a variable.

2106 **DEPRECATED** in *Version 2.5*. Replaced by `SPECIFICATION_LIMITS`.

2107 The `Entry` key **MUST** be one or more from the `SpecificationLimitResult`

2108 keys.

2109 `SpecificationLimitResult` keys:

- 2110 • UpperLimit
- 2111 upper conformance boundary for a variable.

2112 Note: immediate concern or action may be required.

2113 The value of `UpperLimit` **MUST** be float.

- 2114 • Nominal
- 2115 ideal or desired value for a variable.
- 2116 The value of Nominal **MUST** be float.
- 2117 • LowerLimit
- 2118 lower conformance boundary for a variable.
- 2119 Note: immediate concern or action may be required.
- 2120 The value of LowerLimit **MUST** be float.

2121 **5.2.116 SpecificationLimits**

- 2122 set of limits defining a range of values designating acceptable performance for a variable.
- 2123 The Entry key **MUST** be one or more from the SpecificationLimitResult
- 2124 keys.
- 2125 SpecificationLimitResult keys:
- 2126 • UpperLimit
- 2127 upper conformance boundary for a variable.
- 2128 Note: immediate concern or action may be required.
- 2129 The value of UpperLimit **MUST** be float.
- 2130 • Nominal
- 2131 ideal or desired value for a variable.
- 2132 The value of Nominal **MUST** be float.
- 2133 • LowerLimit
- 2134 lower conformance boundary for a variable.
- 2135 Note: immediate concern or action may be required.
- 2136 The value of LowerLimit **MUST** be float.

2137 **5.2.117 SpindleInterlock**

2138 indication of the status of the spindle for a piece of equipment when power has been
2139 removed and it is free to rotate.

2140 `SpindleInterlockEnum` Enumeration:

- 2141 • `ACTIVE`
2142 power has been removed and the spindle cannot be operated.
- 2143 • `INACTIVE`
2144 spindle has not been deactivated.

2145 **5.2.118 Thickness**

2146 dimension between two surfaces of an object, usually the dimension of smallest measure,
2147 for example an additive layer, or a depth of cut.

2148 The default `subType` of `Thickness` is `ACTUAL`.

2149 The value of `Thickness` **MUST** be `float`.

2150 **5.2.118.1 Subtypes of Thickness**

- 2151 • `ACTUAL`
2152 dimension between two surfaces of an object, usually the dimension of smallest
2153 measure, for example an additive layer, or a depth of cut.
2154 that is reported value of an observation.
2155 .
- 2156 • `COMMANDED`
2157 dimension between two surfaces of an object, usually the dimension of smallest
2158 measure, for example an additive layer, or a depth of cut.
2159 that is directive value including adjustments such as an offset or overrides.
2160 .

- 2161 • PROGRAMMED
- 2162 dimension between two surfaces of an object, usually the dimension of smallest
- 2163 measure, for example an additive layer, or a depth of cut.
- 2164 that is directive value without offsets and adjustments.
- 2165 .
- 2166 • TARGET
- 2167 dimension between two surfaces of an object, usually the dimension of smallest
- 2168 measure, for example an additive layer, or a depth of cut.
- 2169 that is goal of the operation or process.
- 2170 .

2171 **5.2.119 ToolAssetId**

2172 identifier of an individual tool asset.

2173 **5.2.120 ToolCuttingItem**

2174 references the CuttingToolLifeCycle CuttingItem index related to the in-

2175 dices of the currently active cutting tool edge.

2176 **5.2.121 ToolGroup**

2177 identifier for the tool group associated with a specific tool. Commonly used to designate

2178 spare tools.

2179 **5.2.122 <<deprecated>>ToolId**

2180 identifier of the tool currently in use for a given Path.

2181 **DEPRECATED** in *Version 1.2.0*. See TOOL_NUMBER.

2182 **5.2.123 ToolNumber**

2183 identifier assigned by the Controller component to a cutting tool when in use by a
2184 piece of equipment.

2185 **5.2.124 ToolOffset**

2186 reference to the tool offset variables applied to the active cutting tool.

2187 The value of ToolOffset **MUST** be float.

2188 A subType **MUST** always be specified.

2189 **5.2.124.1 Subtypes of ToolOffset**

2190 • LENGTH

2191 reference to a length type tool offset variable.

2192 • RADIAL

2193 reference to a radial type tool offset variable.

2194 **5.2.125 ToolOffsets**

2195 tabular representation of properties of each addressable tool offset.

2196 **5.2.126 TransferCount**

2197 accumulation of the number of times an operation has attempted to, or is planned to attempt
2198 to, transfer materials, parts, or other items from one location to another.

2199 The default subType of TransferCount is ALL.

2200 The value of TransferCount **MUST** be integer.

2201 5.2.126.1 Subtypes of TransferCount

- 2202 • ABORTED
- 2203 accumulation of actions or activities that were attempted, but terminated before they
- 2204 could be completed.
- 2205 • ALL
- 2206 accumulation of all actions, items, or activities being counted independent of the
- 2207 outcome.
- 2208 • BAD
- 2209 accumulation of actions, items, or activities being counted that do not conform to
- 2210 specification or expectation.
- 2211 • COMPLETE
- 2212 accumulation of actions, items, or activities that have been completed, independent
- 2213 of the outcome.
- 2214 • FAILED
- 2215 accumulation of actions, items, or activities being counted that do not conform to
- 2216 specification or expectation.
- 2217 • GOOD
- 2218 accumulation of actions, items, or activities being counted that conform to speci-
- 2219 fication or expectation.
- 2220 • REMAINING
- 2221 accumulation of actions, items, or activities yet to be counted.
- 2222 • TARGET
- 2223 goal of the operation or process.

2224 5.2.127 Translation

- 2225 three space linear displacement of an object or coordinate system relative to a *cartesian*
- 2226 *coordinate system*.
- 2227 The units of Translation **MUST** be MILLIMETER_3D.
- 2228 The value of Translation **MUST** be float3d. See *Section 6.1.20* - *<<hasFor-*
- 2229 *matSpecificRepresentation>>float3d*.

2230 5.2.128 Uncertainty

2231 *uncertainty* specified by `UncertaintyType`.

2232 The value of `Uncertainty` **MUST** be double.

2233 5.2.129 UncertaintyType

2234 method used to compute *standard uncertainty*.

2235 The value of `UncertaintyType` **MUST** be one of the `UncertaintyTypeEnum`
2236 enumeration.

2237 `UncertaintyTypeEnum` Enumeration:

2238 • COMBINED

2239 *combined standard uncertainty*.

2240 • MEAN

2241 *standard uncertainty* using arithmetic mean or average the observations. *Ref JCGM*
2242 *100:2008 4.2*

2243 5.2.130 UnloadCount

2244 accumulation of the number of times an operation has attempted to, or is planned to attempt
2245 to, unload materials, parts, or other items.

2246 The default `subType` of `UnloadCount` is ALL.

2247 The value of `UnloadCount` **MUST** be integer.

2248 5.2.130.1 Subtypes of UnloadCount

2249 • ABORTED

2250 accumulation of actions or activities that were attempted, but terminated before they
2251 could be completed.

- 2252 • ALL
- 2253 accumulation of all actions, items, or activities being counted independent of the
- 2254 outcome.
- 2255 • BAD
- 2256 accumulation of actions, items, or activities being counted that do not conform to
- 2257 specification or expectation.
- 2258 • COMPLETE
- 2259 accumulation of actions, items, or activities that have been completed, independent
- 2260 of the outcome.
- 2261 • FAILED
- 2262 accumulation of actions, items, or activities being counted that do not conform to
- 2263 specification or expectation.
- 2264 • GOOD
- 2265 accumulation of actions, items, or activities being counted that conform to specifi-
- 2266 cation or expectation.
- 2267 • REMAINING
- 2268 accumulation of actions, items, or activities yet to be counted.
- 2269 • TARGET
- 2270 goal of the operation or process.

2271 5.2.131 User

- 2272 identifier of the person currently responsible for operating the piece of equipment.
- 2273 A subType **MUST** always be specified.

2274 5.2.131.1 Subtypes of User

- 2275 • MAINTENANCE
- 2276 identifier of the person currently responsible for performing maintenance on the
- 2277 piece of equipment.

- 2278 • OPERATOR
- 2279 identifier of the person currently responsible for operating the piece of equipment.
- 2280 • SET_UP
- 2281 identifier of the person currently responsible for preparing a piece of equipment for
- 2282 production or restoring the piece of equipment to a neutral state after production.

2283 5.2.132 ValveState

2284 state of a valve is one of open, closed, or transitioning between the states.

2285 ValveStateEnum Enumeration:

- 2286 • CLOSED
- 2287 ValveState where flow is not possible, the aperture is static, and the valve is
- 2288 completely shut.
- 2289 • CLOSING
- 2290 valve is transitioning from an OPEN state to a CLOSED state.
- 2291 • OPEN
- 2292 ValveState where flow is allowed and the aperture is static.
- 2293 Note: For a binary value, OPEN indicates the valve has the maximum
- 2294 possible aperture.
- 2295 • OPENING
- 2296 valve is transitioning from a CLOSED state to an OPEN state.

2297 5.2.132.1 Subtypes of ValveState

- 2298 • ACTUAL
- 2299 measured or reported value of an observation.
- 2300 • PROGRAMMED
- 2301 directive value without offsets and adjustments.

2302 5.2.133 Variable

2303 data whose meaning may change over time due to changes in the operation of a piece of
2304 equipment or the process being executed on that piece of equipment.

2305 5.2.134 WaitState

2306 indication of the reason that `Execution` is reporting a value of `WAIT`.

2307 When `result` is not `WAIT`, `isUnavailable` of `WaitState` **MUST** be `true`.

2308 `WaitStateEnum` Enumeration:

- 2309 • `MATERIAL_LOAD`
2310 execution is waiting while material is being loaded.
- 2311 • `MATERIAL_UNLOAD`
2312 execution is waiting while material is being unloaded.
- 2313 • `PART_LOAD`
2314 execution is waiting while one or more discrete workpieces are being loaded.
- 2315 • `PART_UNLOAD`
2316 execution is waiting while one or more discrete workpieces are being unloaded.
- 2317 • `PAUSING`
2318 execution is waiting while the equipment is pausing but the piece of equipment has
2319 not yet reached a fully paused state.
- 2320 • `POWERING_DOWN`
2321 execution is waiting while the equipment is powering down but has not fully reached
2322 a stopped state.
- 2323 • `POWERING_UP`
2324 execution is waiting while the equipment is powering up and is not currently avail-
2325 able to begin producing parts or products.
- 2326 • `RESUMING`
2327 execution is waiting while the equipment is resuming the production cycle but has
2328 not yet resumed execution.

2329 • SECONDARY_PROCESS
 2330 execution is waiting while another process is completed before the execution can
 2331 resume.

2332 • TOOL_LOAD
 2333 execution is waiting while a tool or tooling is being loaded.

2334 • TOOL_UNLOAD
 2335 execution is waiting while a tool or tooling is being unloaded.

2336 5.2.135 Wire

2337 identifier for the type of wire used as the cutting mechanism in Electrical Discharge Ma-
 2338 chining or similar processes.

2339 5.2.136 WorkOffset

2340 reference to offset variables for a work piece or part.

2341 5.2.137 WorkOffsets

2342 tabular representation of properties of each addressable work offset.

2343 5.2.138 WorkholdingId

2344 identifier for the current workholding or part clamp in use by a piece of equipment.

2345 **DEPRECATION WARNING:** Recommend using `FIXTURE_ID` instead.

2346 5.3 Sample Types

2347 This section provides semantic information for `Sample` types.

2348 5.3.1 Acceleration

2349 positive rate of change of velocity.

2350 The units of Acceleration **MUST** be MILLIMETER/SECOND².

2351 The default subType of Acceleration is ACTUAL.

2352 5.3.1.1 Subtypes of Acceleration

2353 • ACTUAL

2354 measured or reported value of an observation.

2355 • COMMANDED

2356 directive value including adjustments such as an offset or overrides.

2357 • PROGRAMMED

2358 directive value without offsets and adjustments.

2359 5.3.2 AccumulatedTime

2360 accumulated time for an activity or event.

2361 The units of AccumulatedTime **MUST** be SECOND.

2362 5.3.3 <<deprecated>>Amperage

2363 strength of electrical current.

2364 **DEPRECATED** in *Version 1.6*. Replaced by AMPERAGE_AC and AMPERAGE_DC.

2365 The units of Amperage **MUST** be AMPERE.

2366 5.3.3.1 Subtypes of Amperage

- 2367 • ACTUAL
- 2368 measured or reported value of an observation.
- 2369 **DEPRECATED** in *Version 1.6*.
- 2370 • ALTERNATING
- 2371 measurement of alternating voltage or current.
- 2372 If not specified further in statistic, defaults to RMS voltage.
- 2373 **DEPRECATED** in *Version 1.6*.
- 2374 • DIRECT
- 2375 measurement of DC current or voltage.
- 2376 **DEPRECATED** in *Version 1.6*.
- 2377 • TARGET
- 2378 goal of the operation or process.
- 2379 **DEPRECATED** in *Version 1.6*.

2380 5.3.4 AmperageAC

- 2381 electrical current that reverses direction at regular short intervals.
- 2382 The units of AmperageAC **MUST** be AMPERE.
- 2383 A subType **MUST** always be specified.

2384 5.3.4.1 Subtypes of AmperageAC

- 2385 • ACTUAL
- 2386 measured or reported value of an observation.
- 2387 • COMMANDED
- 2388 directive value including adjustments such as an offset or overrides.
- 2389 • PROGRAMMED
- 2390 directive value without offsets and adjustments.

2391 **5.3.5 AmperageDC**

2392 electric current flowing in one direction only.

2393 The units of AmperageDC **MUST** be AMPERE.

2394 A subType **MUST** always be specified.

2395 **5.3.5.1 Subtypes of AmperageDC**

2396 • ACTUAL

2397 measured or reported value of an observation.

2398 • COMMANDED

2399 directive value including adjustments such as an offset or overrides.

2400 • PROGRAMMED

2401 directive value without offsets and adjustments.

2402 **5.3.6 Angle**

2403 angular position.

2404 The units of Angle **MUST** be DEGREE.

2405 **5.3.6.1 Subtypes of Angle**

2406 • ACTUAL

2407 measured or reported value of an observation.

2408 • COMMANDED

2409 directive value including adjustments such as an offset or overrides.

2410 5.3.7 AngularAcceleration

2411 positive rate of change of angular velocity.

2412 The units of AngularAcceleration **MUST** be DEGREE/SECOND².

2413 The default subType of AngularAcceleration is ACTUAL.

2414 5.3.7.1 Subtypes of AngularAcceleration

2415 • ACTUAL

2416 measured or reported value of an observation.

2417 • COMMANDED

2418 directive value including adjustments such as an offset or overrides.

2419 • PROGRAMMED

2420 directive value without offsets and adjustments.

2421 5.3.8 AngularDeceleration

2422 negative rate of change of angular velocity.

2423 The units of AngularDeceleration **MUST** be DEGREE/SECOND².

2424 The default subType of AngularDeceleration is ACTUAL.

2425 5.3.8.1 Subtypes of AngularDeceleration

2426 • ACTUAL

2427 measured or reported value of an observation.

2428 • COMMANDED

2429 directive value including adjustments such as an offset or overrides.

2430 • PROGRAMMED

2431 directive value without offsets and adjustments.

2432 **5.3.9 AngularVelocity**

2433 rate of change of angular position.

2434 The units of AngularVelocity **MUST** be DEGREE/SECOND.

2435 **5.3.10 AssetUpdateRate**

2436 average rate of change of values for assets in the MTConnect streams.

2437 The average is computed over a rolling window defined by the implementation.

2438 The units of AssetUpdateRate **MUST** be COUNT/SECOND.

2439 **5.3.11 AxisFeedrate**

2440 feedrate of a linear axis.

2441 The units of AxisFeedrate **MUST** be MILLIMETER/SECOND.

2442 **5.3.11.1 Subtypes of AxisFeedrate**

- 2443 • ACTUAL
- 2444 measured or reported value of an observation.
- 2445 • COMMANDED
- 2446 directive value including adjustments such as an offset or overrides.
- 2447 • JOG
- 2448 relating to momentary activation of a function or a movement.
- 2449 **DEPRECATION WARNING:** May be deprecated in the future.
- 2450 • OVERRIDE
- 2451 operator's overridden value.
- 2452 • PROGRAMMED
- 2453 directive value without offsets and adjustments.

- 2454 • RAPID
- 2455 performing an operation faster or in less time than nominal rate.

2456 **5.3.12 BatteryCapacity**

2457 maximum rated charge a battery is capable of maintaining based on the battery discharging
2458 at a specified current over a specified time period.

2459 The units of BatteryCapacity **MUST** be COULOMB.

2460 The default subType of BatteryCapacity is ACTUAL.

2461 **5.3.12.1 Subtypes of BatteryCapacity**

- 2462 • ACTUAL
- 2463 measured or reported value of an observation.
- 2464 • TARGET
- 2465 goal of the operation or process.

2466 **5.3.13 BatteryCharge**

2467 value of the battery's present capacity expressed as a percentage of the battery's maximum
2468 rated capacity.

2469 The units of BatteryCharge **MUST** be PERCENT.

2470 The default subType of BatteryCharge is ACTUAL.

2471 **5.3.13.1 Subtypes of BatteryCharge**

- 2472 • ACTUAL
- 2473 measured or reported value of an observation.
- 2474 • TARGET
- 2475 goal of the operation or process.

2476 **5.3.14 CapacityFluid**

2477 maximum amount of fluid that can be held by a container.

2478 The units of CapacityFluid **MUST** be MILLILITER.

2479 **5.3.15 CapacitySpatial**

2480 maximum amount of material that can be held by a container.

2481 The units of CapacitySpatial **MUST** be CUBIC_MILLIMETER.

2482 **5.3.16 ChargeRate**

2483 value of the current being supplied to the Component for the purpose of charging.

2484 The units of ChargeRate **MUST** be AMPERE.

2485 The default subType of ChargeRate is ACTUAL.

2486 **5.3.16.1 Subtypes of ChargeRate**

- 2487 • ACTUAL
- 2488 measured or reported value of an observation.
- 2489 • TARGET
- 2490 goal of the operation or process.

2491 **5.3.17 Concentration**

2492 percentage of one component within a mixture of components.

2493 The units of Concentration **MUST** be PERCENT.

2494 **5.3.18 Conductivity**

2495 ability of a material to conduct electricity.

2496 The units of Conductivity **MUST** be SIEMENS/METER.

2497 **5.3.19 CuttingSpeed**

2498 speed difference (relative velocity) between the cutting mechanism and the surface of the
2499 workpiece it is operating on.

2500 The units of CuttingSpeed **MUST** be MILLIMETER/SECOND.

2501 **5.3.19.1 Subtypes of CuttingSpeed**

- 2502 • ACTUAL
- 2503 measured or reported value of an observation.
- 2504 • COMMANDED
- 2505 directive value including adjustments such as an offset or overrides.
- 2506 • PROGRAMMED
- 2507 directive value without offsets and adjustments.

2508 **5.3.20 Deceleration**

2509 negative rate of change of velocity.

2510 The units of Deceleration **MUST** be MILLIMETER/SECOND².

2511 The default subType of Deceleration is ACTUAL.

2512 **5.3.20.1 Subtypes of Deceleration**

- 2513 • ACTUAL
- 2514 measured or reported value of an observation.

- 2515 • COMMANDED
- 2516 directive value including adjustments such as an offset or overrides.
- 2517 • PROGRAMMED
- 2518 directive value without offsets and adjustments.

2519 5.3.21 Density

- 2520 volumetric mass of a material per unit volume of that material.
- 2521 The units of Density **MUST** be MILLIGRAM/CUBIC_MILLIMETER.

2522 5.3.22 DepositionAccelerationVolumetric

- 2523 rate of change in spatial volume of material deposited in an additive manufacturing pro-
- 2524 cess.
- 2525 The units of DepositionAccelerationVolumetric **MUST** be CUBIC_MILLIMETER/SECOND²

2526 5.3.22.1 Subtypes of DepositionAccelerationVolumetric

- 2527 • ACTUAL
- 2528 measured or reported value of an observation.
- 2529 • COMMANDED
- 2530 directive value including adjustments such as an offset or overrides.

2531 5.3.23 DepositionDensity

- 2532 density of the material deposited in an additive manufacturing process per unit of volume.
- 2533 The units of DepositionDensity **MUST** be MILLIGRAM/CUBIC_MILLIMETER.

2534 **5.3.23.1 Subtypes of DepositionDensity**

- 2535 • ACTUAL
- 2536 measured or reported value of an observation.
- 2537 • COMMANDED
- 2538 directive value including adjustments such as an offset or overrides.

2539 **5.3.24 DepositionMass**

2540 mass of the material deposited in an additive manufacturing process.

2541 The units of DepositionMass **MUST** be MILLIGRAM.

2542 **5.3.24.1 Subtypes of DepositionMass**

- 2543 • ACTUAL
- 2544 measured or reported value of an observation.
- 2545 • COMMANDED
- 2546 directive value including adjustments such as an offset or overrides.

2547 **5.3.25 DepositionRateVolumetric**

2548 rate at which a spatial volume of material is deposited in an additive manufacturing pro-
2549 cess.

2550 The units of DepositionRateVolumetric **MUST** be CUBIC_MILLIMETER/SECOND.

2551 **5.3.25.1 Subtypes of DepositionRateVolumetric**

- 2552 • ACTUAL
- 2553 measured or reported value of an observation.
- 2554 • COMMANDED
- 2555 directive value including adjustments such as an offset or overrides.

2556 **5.3.26 DepositionVolume**

2557 spatial volume of material to be deposited in an additive manufacturing process.

2558 The units of DepositionVolume **MUST** be CUBIC_MILLIMETER.

2559 **5.3.26.1 Subtypes of DepositionVolume**

2560 • ACTUAL

2561 measured or reported value of an observation.

2562 • COMMANDED

2563 directive value including adjustments such as an offset or overrides.

2564 **5.3.27 DewPoint**

2565 temperature at which moisture begins to condense, corresponding to saturation for a given
2566 absolute humidity.

2567 The units of DewPoint **MUST** be CELSIUS.

2568 **5.3.28 Diameter**

2569 dimension of a diameter.

2570 The units of Diameter **MUST** be MILLIMETER.

2571 **5.3.29 DischargeRate**

2572 value of current being drawn from the Component.

2573 The units of DischargeRate **MUST** be AMPERE.

2574 The default subType of DischargeRate is ACTUAL.

2575 **5.3.29.1 Subtypes of DischargeRate**

- 2576 • ACTUAL
- 2577 measured or reported value of an observation.
- 2578 • TARGET
- 2579 goal of the operation or process.

2580 **5.3.30 Displacement**

- 2581 change in position of an object.
- 2582 The units of Displacement **MUST** be MILLIMETER.

2583 **5.3.31 DisplacementAngular**

- 2584 absolute value of the change in angular position around a vector
- 2585 Note: The displacement vector **MAY** be defined by the motion of the owning
- 2586 Component.
- 2587 The units of DisplacementAngular **MUST** be DEGREE.

2588 **5.3.32 DisplacementLinear**

- 2589 absolute value of the change in position along a vector.
- 2590 Note: The displacement vector **MAY** be defined by the motion of the owning
- 2591 Component.
- 2592 The units of DisplacementLinear **MUST** be MILLIMETER.

2593 5.3.33 ElectricalEnergy

2594 Wattage used or generated by a component over an interval of time.

2595 The units of ElectricalEnergy **MUST** be WATT_SECOND.

2596 5.3.34 EquipmentTimer

2597 amount of time a piece of equipment or a sub-part of a piece of equipment has performed
2598 specific activities.

2599 The units of EquipmentTimer **MUST** be SECOND.

2600 A subType **MUST** always be specified.

2601 5.3.34.1 Subtypes of EquipmentTimer

2602 • DELAY

2603 elapsed time of a temporary halt of action.

2604 • LOADED

2605 time that the sub-parts of a piece of equipment are under load.

2606 Example: For traditional machine tools, this is a measurement of the time that the
2607 cutting tool is assumed to be engaged with the part.

2608 • OPERATING

2609 time that the major sub-parts of a piece of equipment are powered or performing any
2610 activity whether producing a part or product or not.

2611 Example: For traditional machine tools, this includes WORKING, plus idle time.

2612 • POWERED

2613 time that primary power is applied to the piece of equipment and, as a minimum, the
2614 controller or logic portion of the piece of equipment is powered and functioning or
2615 components that are required to remain on are powered.

2616 Example: Heaters for an extrusion machine that are required to be powered even
2617 when the equipment is turned off.

- 2618 • WORKING
- 2619 time that a piece of equipment is performing any activity the equipment is active and
- 2620 performing a function under load or not.
- 2621 Example: For traditional machine tools, this includes LOADED, plus rapid moves,
- 2622 tool changes, etc.

2623 5.3.35 FillHeight

2624 amount of a substance in a container.

2625 The units of FillHeight **MUST** be MILLIMETER.

2626 5.3.35.1 Subtypes of FillHeight

- 2627 • ACTUAL
- 2628 amount of a substance in a container.
- 2629 that is reported value of an observation.
- 2630 • TARGET
- 2631 amount of a substance in a container.
- 2632 that is goal of the operation or process.

2633 5.3.36 FillLevel

2634 amount of a substance remaining compared to the planned maximum amount of that sub-
2635 stance.

2636 The units of FillLevel **MUST** be PERCENT.

2637 5.3.37 Flow

2638 rate of flow of a fluid.

2639 The units of Flow **MUST** be LITER/SECOND.

2640 **5.3.38 FollowingError**

2641 difference between actual and commanded position at any specific point in time during a
2642 motion.

2643 The units of FollowingError **MUST** be COUNT.

2644 **5.3.38.1 Subtypes of FollowingError**

- 2645 • ACTUAL
- 2646 measured or reported value of an observation.

2647 **5.3.39 FollowingErrorAngular**

2648 angular difference between the commanded encoder/resolver position and the actual en-
2649 coder/resolver position at any specified point in time during a motion.

2650 The units of FollowingErrorAngular **MUST** be DEGREE.

2651 **5.3.39.1 Subtypes of FollowingErrorAngular**

- 2652 • ACTUAL
- 2653 measured or reported value of an observation.

2654 **5.3.40 FollowingErrorLinear**

2655 difference between the commanded encoder/resolver position and the actual encoder/re-
2656 solver position at any specified point in time during a motion.

2657 The units of FollowingErrorLinear **MUST** be MILLIMETER.

2658 **5.3.40.1 Subtypes of FollowingErrorLinear**

- 2659 • ACTUAL
- 2660 measured or reported value of an observation.

2661 5.3.41 Frequency

2662 number of occurrences of a repeating event per unit time.

2663 The units of Frequency **MUST** be HERTZ.

2664 5.3.42 <<deprecated>>GlobalPosition

2665 position in three-dimensional space.

2666 **DEPRECATED** in Version 1.1.

2667 The units of GlobalPosition **MUST** be MILLIMETER.

2668 5.3.42.1 Subtypes of GlobalPosition

2669 • ACTUAL

2670 measured or reported value of an observation.

2671 • COMMANDED

2672 directive value including adjustments such as an offset or overrides.

2673 5.3.43 GravitationalAcceleration

2674 acceleration relative to Earth's gravity of 9.80665 METER/SECOND².

2675 The units of GravitationalAcceleration **MUST** be GRAVITATIONAL_ACCELERATION.

2676 5.3.44 GravitationalForce

2677 force relative to earth's gravity.

2678 Note: *Mass × GravitationalAcceleration*

2679 The units of GravitationalForce **MUST** be GRAVITATIONAL_FORCE.

2680 **5.3.45 HumidityAbsolute**

2681 amount of water vapor expressed in grams per cubic meter.

2682 The units of HumidityAbsolute **MUST** be GRAM/CUBIC_METER.

2683 **5.3.45.1 Subtypes of HumidityAbsolute**

- 2684 • ACTUAL
- 2685 measured or reported value of an observation.
- 2686 • COMMANDED
- 2687 directive value including adjustments such as an offset or overrides.

2688 **5.3.46 HumidityRelative**

2689 amount of water vapor present expressed as a percent to reach saturation at the same tem-
2690 perature.

2691 The units of HumidityRelative **MUST** be PERCENT.

2692 **5.3.46.1 Subtypes of HumidityRelative**

- 2693 • ACTUAL
- 2694 measured or reported value of an observation.
- 2695 • COMMANDED
- 2696 directive value including adjustments such as an offset or overrides.

2697 **5.3.47 HumiditySpecific**

2698 ratio of the water vapor present over the total weight of the water vapor and air present
2699 expressed as a percent.

2700 The units of HumiditySpecific **MUST** be PERCENT.

2701 **5.3.47.1 Subtypes of HumiditySpecific**

- 2702 • ACTUAL
- 2703 measured or reported value of an observation.
- 2704 • COMMANDED
- 2705 directive value including adjustments such as an offset or overrides.

2706 **5.3.48 Length**

- 2707 length of an object.
- 2708 The units of Length **MUST** be MILLIMETER.

2709 **5.3.48.1 Subtypes of Length**

- 2710 • REMAINING
- 2711 remaining total length of an object.
- 2712 • STANDARD
- 2713 standard or original length of an object.
- 2714 • USEABLE
- 2715 remaining usable length of an object.

2716 **5.3.49 <<deprecated>>Level**

- 2717 level of a resource.
- 2718 **DEPRECATED** in *Version 1.2*. See FILL_LEVEL.
- 2719 The units of Level **MUST** be PERCENT.

2720 5.3.50 LinearForce

2721 *force* applied to a mass in one direction only.

2722 The units of LinearForce **MUST** be NEWTON.

2723 5.3.51 Load

2724 actual versus the standard rating of a piece of equipment.

2725 The units of Load **MUST** be PERCENT.

2726 5.3.52 Mass

2727 mass of an object(s) or an amount of material.

2728 The units of Mass **MUST** be KILOGRAM.

2729 5.3.53 ObservationUpdateRate

2730 average rate of change of values for data items in the MTConnect streams. The average is
2731 computed over a rolling window defined by the implementation.

2732 The units of ObservationUpdateRate **MUST** be COUNT/SECOND.

2733 5.3.54 Openness

2734 percentage open where 100% is fully open and 0% is fully closed.

2735 The units of Openness **MUST** be PERCENT.

2736 5.3.55 Orientation

2737 angular position of a plane or vector relative to a *cartesian coordinate system*

2738 The units of Orientation **MUST** be DEGREE_3D.

2739 The value of Orientation **MUST** be float3d. See *Section 6.1.20* - <<hasFor-
2740 matSpecificRepresentation>>float3d.

2741 **5.3.55.1 Subtypes of Orientation**

- 2742 • ACTUAL
- 2743 measured or reported value of an observation.
- 2744 • COMMANDED
- 2745 directive value including adjustments such as an offset or overrides.

2746 **5.3.56 PH**

2747 acidity or alkalinity of a solution.

2748 The units of PH **MUST** be PH.

2749 **5.3.57 ParticleCount**

2750 number of particles counted by their size or other characteristics.

2751 The units of ParticleCount **MUST** be COUNT.

2752 **5.3.57.1 Subtypes of ParticleCount**

- 2753 • GAS
- 2754 number of particles counted by their size or other characteristics.
- 2755 for a
- 2756 • LIQUID
- 2757 number of particles counted by their size or other characteristics.
- 2758 for a

- 2759 • SOLID
- 2760 number of particles counted by their size or other characteristics.
- 2761 for a

2762 **5.3.58 ParticleSize**

- 2763 size of particles counted by their size or other characteristics.
- 2764 The units of ParticleSize **MUST** be MILLIMETER.

2765 **5.3.59 PathFeedrate**

- 2766 feedrate for the axes, or a single axis, associated with a Path component.
- 2767 The units of PathFeedrate **MUST** be MILLIMETER/SECOND.

2768 **5.3.59.1 Subtypes of PathFeedrate**

- 2769 • ACTUAL
- 2770 measured or reported value of an observation.
- 2771 • COMMANDED
- 2772 directive value including adjustments such as an offset or overrides.
- 2773 • JOG
- 2774 relating to momentary activation of a function or a movement.
- 2775 **DEPRECATION WARNING:** May be deprecated in the future.
- 2776 • OVERRIDE
- 2777 operator's overridden value.
- 2778 **DEPRECATED** in *Version 1.3*.
- 2779 • PROGRAMMED
- 2780 directive value without offsets and adjustments.
- 2781 • RAPID
- 2782 performing an operation faster or in less time than nominal rate.

2783 5.3.60 PathFeedratePerRevolution

2784 feedrate for the axes, or a single axis.

2785 The units of PathFeedratePerRevolution **MUST** be MILLIMETER/REVO-
2786 LUTION.

2787 5.3.60.1 Subtypes of PathFeedratePerRevolution

- 2788 • ACTUAL
2789 measured or reported value of an observation.
- 2790 • COMMANDED
2791 directive value including adjustments such as an offset or overrides.
- 2792 • PROGRAMMED
2793 directive value without offsets and adjustments.

2794 5.3.61 <<hasFormatSpecificRepresentation>>PathPosition

2795 position of a control point associated with a Controller or a Path.

Example 1: XML Device Model Example for PathPosition using MILLIMETER_3D unit

```
2796 1 <DataItem type='PATH\textunderscore POSITION' id='pathposit1' units=
2797    "MILLIMETER\textunderscore_3D"/>
```

Example 2: XML Streams Response Example for PathPosition using MILLIMETER_3D unit

```
2798 1 <PathPosition>10.0 0.0 20.0</PathPosition>
```

Example 3: XML Device Model Example for PathPosition to demonstrate multi-dimensional representation using DataSet representation

```
2799 1 <DataItem type='PATH\textunderscore POSITION' id='pathposit1'
2800    representation="DATA\textunderscore_SET">
2801 2 <Definition>
2802 3   <EntryDefinition key='X' units='MILLIMETER'>
2803 4   <EntryDefinition key='Y' units='MILLIMETER'>
2804 5   <EntryDefinition key='Z' units='MILLIMETER'>
2805 6 </Definition>
2806 7 </DataItem>
```

Example 4: XML Streams Response Example for PathPosition to demonstrate multi-dimensional representation using DataSet representation

```
2807 1 <PathPositionDataSet id='pathposit1'>
2808 2   <Entry key='X'>10.0</Entry>
2809 3   <Entry key='Z'>20.0</Entry>
2810 4 </PathPositionDataSet>
```

2811 The units of PathPosition **MUST** be MILLIMETER_3D.

2812 The value of PathPosition **MUST** be float3d. See Section 6.1.20 - <<hasFor-
2813 matSpecificRepresentation>>float3d.

2814 5.3.61.1 Subtypes of PathPosition

- 2815 • ACTUAL
- 2816 measured or reported value of an observation.
- 2817 • COMMANDED
- 2818 directive value including adjustments such as an offset or overrides.
- 2819 • PROBE
- 2820 position provided by a measurement probe.
- 2821 **DEPRECATION WARNING:** May be deprecated in the future.
- 2822 • TARGET
- 2823 goal of the operation or process.

2824 5.3.62 Position

2825 point along an axis in a *cartesian coordinate system*.

2826 The units of Position **MUST** be MILLIMETER.

2827 5.3.62.1 Subtypes of Position

- 2828 • ACTUAL
- 2829 measured or reported value of an observation.

- 2830 • COMMANDED
- 2831 directive value including adjustments such as an offset or overrides.
- 2832 • PROGRAMMED
- 2833 directive value without offsets and adjustments.
- 2834 • TARGET
- 2835 goal of the operation or process.

2836 **5.3.63 PositionCartesian**

- 2837 point in a *cartesian coordinate system*.
- 2838 The units of PositionCartesian **MUST** be MILLIMETER_3D.
- 2839 The value of PositionCartesian **MUST** be float3d. See Section 6.1.20 - <<has-
2840 FormatSpecificRepresentation>>float3d.

2841 **5.3.64 PowerFactor**

- 2842 ratio of real power flowing to a load to the apparent power in that AC circuit.
- 2843 The units of PowerFactor **MUST** be PERCENT.

2844 **5.3.65 Pressure**

- 2845 force per unit area measured relative to atmospheric pressure.
- 2846 Commonly referred to as gauge pressure.
- 2847 The units of Pressure **MUST** be PASCAL.

2848 **5.3.66 PressureAbsolute**

- 2849 The force per unit area measured relative to a vacuum.
- 2850 The units of PressureAbsolute **MUST** be PASCAL.

2851 **5.3.67 PressurizationRate**

2852 change of pressure per unit time.

2853 The units of PressurizationRate **MUST** be PASCAL/SECOND.

2854 **5.3.67.1 Subtypes of PressurizationRate**

- 2855 • ACTUAL
- 2856 measured or reported value of an observation.
- 2857 • COMMANDED
- 2858 directive value including adjustments such as an offset or overrides.
- 2859 • PROGRAMMED
- 2860 directive value without offsets and adjustments.

2861 **5.3.68 ProcessTimer**

2862 amount of time a piece of equipment has performed different types of activities associated
2863 with the process being performed at that piece of equipment.

2864 The units of ProcessTimer **MUST** be SECOND.

2865 A subType **MUST** always be specified.

2866 **5.3.68.1 Subtypes of ProcessTimer**

- 2867 • DELAY
- 2868 elapsed time of a temporary halt of action.
- 2869 • PROCESS
- 2870 time from the beginning of production of a part or product on a piece of equipment
- 2871 until the time that production is complete for that part or product on that piece of
- 2872 equipment.
- 2873 This includes the time that the piece of equipment is running, producing parts or
- 2874 products, or in the process of producing parts.

2875 5.3.69 Resistance

2876 degree to which a substance opposes the passage of an electric current.

2877 The units of Resistance **MUST** be OHM.

2878 5.3.70 Resistivity

2879 inability of a material to conduct electricity.

2880 The units of Resistivity **MUST** be OHM_METER.

2881 5.3.71 RotaryVelocity

2882 rotational speed of a rotary axis.

2883 The units of RotaryVelocity **MUST** be REVOLUTION/MINUTE.

2884 5.3.71.1 Subtypes of RotaryVelocity

- 2885 • ACTUAL
- 2886 measured or reported value of an observation.
- 2887 • COMMANDED
- 2888 directive value including adjustments such as an offset or overrides.
- 2889 • OVERRIDE
- 2890 The operators overridden value.
- 2891 • PROGRAMMED
- 2892 directive value without offsets and adjustments.

2893 5.3.72 SettlingError

2894 difference between actual and commanded position at the end of a motion.

2895 The units of SettlingError **MUST** be COUNT.

2896 **5.3.72.1 Subtypes of SettlingError**

- 2897 • ACTUAL
- 2898 measured or reported value of an observation.

2899 **5.3.73 SettlingErrorAngular**

2900 angular difference between the commanded encoder/resolver position, and the actual en-
2901 coder/resolver position when motion is complete.

2902 The units of SettlingErrorAngular **MUST** be DEGREE.

2903 **5.3.73.1 Subtypes of SettlingErrorAngular**

- 2904 • ACTUAL
- 2905 measured or reported value of an observation.

2906 **5.3.74 SettlingErrorLinear**

2907 difference between the commanded encoder/resolver position, and the actual encoder/re-
2908 solver position when motion is complete.

2909 The units of SettlingErrorLinear **MUST** be MILLIMETER.

2910 **5.3.74.1 Subtypes of SettlingErrorLinear**

- 2911 • ACTUAL
- 2912 measured or reported value of an observation.

2913 **5.3.75 SoundLevel**

2914 sound level or sound pressure level relative to atmospheric pressure.

2915 The units of SoundLevel **MUST** be DECIBEL.

2916 The default subType of SoundLevel is A_SCALE.

2917 **5.3.75.1 Subtypes of SoundLevel**

2918 • A_SCALE

2919 A Scale weighting factor. This is the default weighting factor if no factor is specified

2920 • B_SCALE

2921 B Scale weighting factor

2922 • C_SCALE

2923 C Scale weighting factor

2924 • D_SCALE

2925 D Scale weighting factor

2926 • NO_SCALE

2927 No weighting factor on the frequency scale

2928 **5.3.76 <<deprecated>>SpindleSpeed**

2929 rotational speed of the rotary axis.

2930 **DEPRECATED** in *Version 1.2*. Replaced by ROTARY_VELOCITY.

2931 The units of SpindleSpeed **MUST** be REVOLUTION/MINUTE.

2932 **5.3.76.1 Subtypes of SpindleSpeed**

2933 • ACTUAL

2934 measured or reported value of an observation.

2935 **DEPRECATED** in *Version 1.3*.

- 2936 • COMMANDED
- 2937 directive value including adjustments such as an offset or overrides.
- 2938 **DEPRECATED** in *Version 1.3*.
- 2939 • OVERRIDE
- 2940 operator's overridden value.
- 2941 **DEPRECATED** in *Version 1.3*.

2942 **5.3.77 Strain**

- 2943 amount of deformation per unit length of an object when a load is applied.
- 2944 The units of Strain **MUST** be PERCENT.

2945 **5.3.78 Temperature**

- 2946 degree of hotness or coldness measured on a definite scale.
- 2947 The units of Temperature **MUST** be CELSIUS.

2948 **5.3.79 Tension**

- 2949 force that stretches or elongates an object.
- 2950 The units of Tension **MUST** be NEWTON.

2951 **5.3.80 Tilt**

- 2952 angular displacement.
- 2953 The units of Tilt **MUST** be MICRO_RADIAN.

2954 5.3.81 Torque

2955 turning force exerted on an object or by an object.

2956 The units of Torque **MUST** be NEWTON_METER.

2957 5.3.82 Velocity

2958 rate of change of position of a Component.

2959 The units of Velocity **MUST** be MILLIMETER/SECOND.

2960 5.3.83 Viscosity

2961 fluid's resistance to flow.

2962 The units of Viscosity **MUST** be PASCAL_SECOND.

2963 5.3.84 VoltAmpere

2964 apparent power in an electrical circuit, equal to the product of root-mean-square (RMS)
2965 voltage and RMS current (commonly referred to as VA).

2966 The units of VoltAmpere **MUST** be VOLT_AMPERE.

2967 5.3.85 VoltAmpereReactive

2968 reactive power in an AC electrical circuit (commonly referred to as VAR).

2969 The units of VoltAmpereReactive **MUST** be VOLT_AMPERE_REACTIVE.

2970 5.3.86 <<deprecated>>Voltage

2971 electrical potential between two points.

2972 **DEPRECATED** in *Version 1.6*. Replaced by VOLTAGE_AC and VOLTAGE_DC.

2973 The units of Voltage **MUST** be VOLT.

2974 **5.3.86.1 Subtypes of Voltage**

- 2975 • ACTUAL
- 2976 measured or reported value of an observation.
- 2977 **DEPRECATED** in *Version 1.6*.
- 2978 • ALTERNATING
- 2979 alternating voltage or current.
- 2980 If not specified further in statistic, defaults to RMS voltage.
- 2981 **DEPRECATED** in *Version 1.6*.
- 2982 • DIRECT
- 2983 DC current or voltage.
- 2984 **DEPRECATED** in *Version 1.6*.
- 2985 • TARGET
- 2986 goal of the operation or process.
- 2987 **DEPRECATED** in *Version 1.6*.

2988 **5.3.87 VoltageAC**

2989 electrical potential between two points in an electrical circuit in which the current period-
2990 ically reverses direction.

2991 The units of VoltageAC **MUST** be VOLT.

2992 A subType **MUST** always be specified.

2993 **5.3.87.1 Subtypes of VoltageAC**

- 2994 • ACTUAL
- 2995 measured or reported value of an observation.

- 2996 • COMMANDED
- 2997 directive value including adjustments such as an offset or overrides.
- 2998 • PROGRAMMED
- 2999 directive value without offsets and adjustments.

3000 **5.3.88 VoltageDC**

- 3001 electrical potential between two points in an electrical circuit in which the current is uni-
3002 directional.
- 3003 The units of VoltageDC **MUST** be VOLT.
- 3004 A subType **MUST** always be specified.

3005 **5.3.88.1 Subtypes of VoltageDC**

- 3006 • ACTUAL
- 3007 measured or reported value of an observation.
- 3008 • COMMANDED
- 3009 directive value including adjustments such as an offset or overrides.
- 3010 • PROGRAMMED
- 3011 directive value without offsets and adjustments.

3012 **5.3.89 VolumeFluid**

- 3013 fluid volume of an object or container.
- 3014 The units of VolumeFluid **MUST** be MILLILITER.

3015 **5.3.89.1 Subtypes of VolumeFluid**

- 3016 • ACTUAL
- 3017 measured or reported value of an observation.

- 3018 • CONSUMED
- 3019 reported or measured value of the amount used in the manufacturing process.
- 3020 • ENDED
- 3021 boundary when an activity or an event terminates.
- 3022 • PART
- 3023 reported or measured value of amount included in the *part*.
- 3024 • START
- 3025 boundary when an activity or an event commences.
- 3026 • WASTE
- 3027 reported or measured value of the amount discarded.

3028 **5.3.90 VolumeSpatial**

- 3029 geometric volume of an object or container.
- 3030 The units of VolumeSpatial **MUST** be CUBIC_MILLIMETER.

3031 **5.3.90.1 Subtypes of VolumeSpatial**

- 3032 • ACTUAL
- 3033 measured or reported value of an observation.
- 3034 • CONSUMED
- 3035 reported or measured value of the amount used in the manufacturing process.
- 3036 • ENDED
- 3037 boundary when an activity or an event terminates.
- 3038 • PART
- 3039 reported or measured value of amount included in the *part*.
- 3040 • START
- 3041 boundary when an activity or an event commences.
- 3042 • WASTE
- 3043 reported or measured value of the amount discarded

3044 **5.3.91 Wattage**

3045 power flowing through or dissipated by an electrical circuit or piece of equipment.

3046 The units of Wattage **MUST** be WATT.

3047 **5.3.91.1 Subtypes of Wattage**

- 3048 • ACTUAL
- 3049 measured or reported value of an observation.
- 3050 • TARGET
- 3051 goal of the operation or process.

3052 **5.3.92 XDimension**

3053 dimension of an entity relative to the X direction of the referenced coordinate system.

3054 The units of XDimension **MUST** be MILLIMETER.

3055 **5.3.93 YDimension**

3056 dimension of an entity relative to the Y direction of the referenced coordinate system.

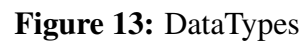
3057 The units of YDimension **MUST** be MILLIMETER.

3058 **5.3.94 ZDimension**

3059 dimension of an entity relative to the Z direction of the referenced coordinate system.

3060 The units of ZDimension **MUST** be MILLIMETER.

3064 6.1 DataTypes



3065 6.1.1 boolean

3067 **6.1.2 ID**

3069 6.1.3 string

3071 **6.1.4 float**

MTConnect Part 3.0: Observation Information Model - Version 2.5.0

3073 6.1.5 datetime

3074 string that represents timestamp in ISO 8601 format.

3075 6.1.6 integer

3076 primitive type.

3077 6.1.7 xlinktype

3078 string that represents the type of an XLink element. See [https://www.w3.org/TR/](https://www.w3.org/TR/xlink11/)
3079 [xlink11/](https://www.w3.org/TR/xlink11/).

3080 6.1.8 xslang

3081 string that represents a language tag. See [http://www.ietf.org/rfc/rfc4646.](http://www.ietf.org/rfc/rfc4646.txt)
3082 [txt](http://www.ietf.org/rfc/rfc4646.txt).

3083 6.1.9 SECOND

3084 float that represents time in seconds.

3085 6.1.10 xlinkhref

3086 string that represents the locator attribute of an XLink element. See [https://www.w3.](https://www.w3.org/TR/xlink11/)
3087 [org/TR/xlink11/](https://www.w3.org/TR/xlink11/).

3088 6.1.11 x509

3089 string that represents an x509 data block. *Ref ISO/IEC 9594-8:2020.*

3090 6.1.12 int32

3091 32-bit integer.

3092 6.1.13 int64

3093 64-bit integer.

3094 6.1.14 version

3095 series of three numeric values, separated by a decimal point, representing a *major*, *minor*,
3096 and *patch* number of the MTConnect Standard.

3097 6.1.15 uint32

3098 32-bit unsigned integer.

3099 6.1.16 uint64

3100 64-bit unsigned integer.

3101 6.1.17 binary

3102 base-2 numeral system or binary numeral system represented by two digits: “0” and “1”.

3103 6.1.18 double

3104 primitive type.

3105 6.1.19 Array

3106 array.

3107 6.1.20 <<hasFormatSpecificRepresentation>>float3d

3108 array of size 3 and datatype float.

3109 6.1.21 UUID

3110 Universally Unique Identifier. *Ref IETF:RFC-4122*

3111 6.1.22 METER

3112 float that represents measurement in meter.

3113 6.2 Stereotypes

3114 6.2.1 organizer

3115 element that *organizes* other elements of a type.

3116 6.2.2 deprecated

3117 element that has been deprecated.

3118 6.2.3 extensible

3119 enumeration that can be extended.

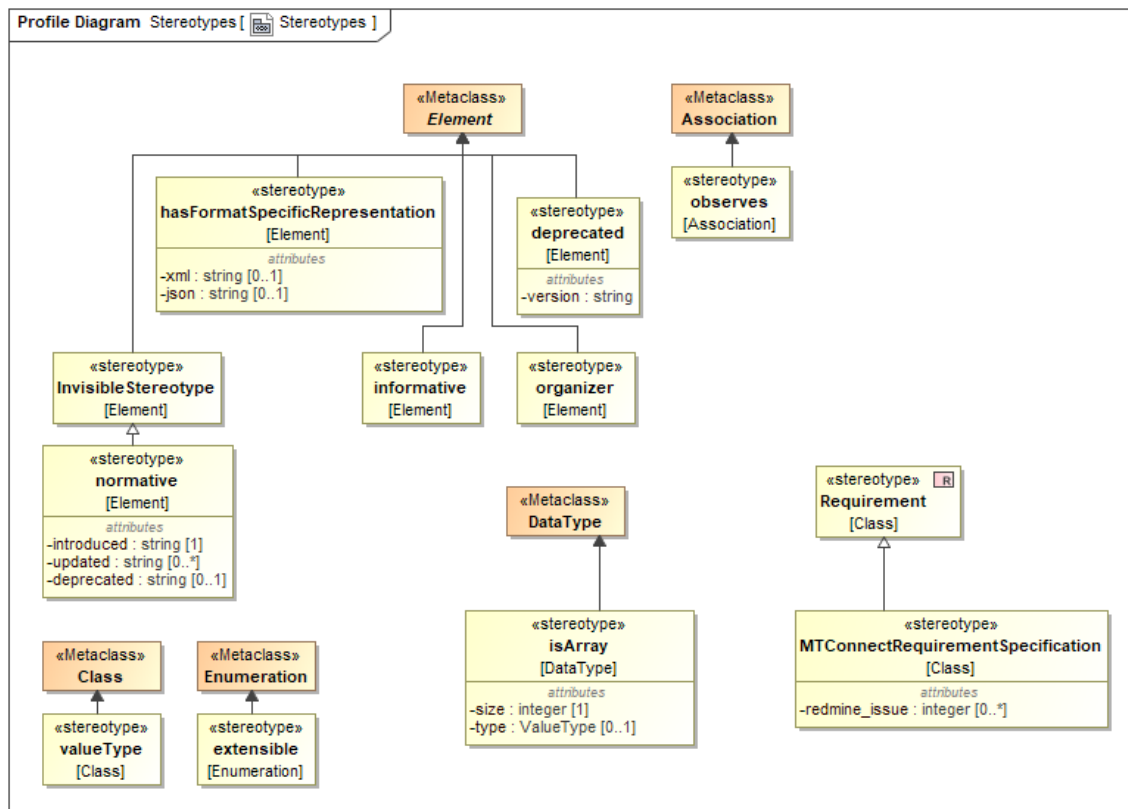


Figure 14: Stereotypes

3120 6.2.4 informative

3121 element that is descriptive and non-normative.

3122 6.2.5 normative

3123 element that has been added to the standard.

3124 6.2.6 observes

3125 association in which a *Component* makes *Observations* about an observable *DataItem*.

3126 6.2.7 satisfiedBy

3127 6.2.8 hasFormatSpecificRepresentation

3128 element that has format specific representation that might be different from the element's
3129 SysML representation.

3130 6.2.9 valueType

3131 extends `Class` to be used as a SysML `<<ValueType>>`.

3132 6.2.10 isArray

3133 datatype that is an array.

3134 6.2.11 MTConnectRequirementSpecification

3135 MTConnect Requirement.

3136 Appendices

3137 A Bibliography

- 3138 Engineering Industries Association. EIA Standard - EIA-274-D, Interchangeable Variable,
3139 Block Data Format for Positioning, Contouring, and Contouring/Positioning Numerically
3140 Controlled Machines. Washington, D.C. 1979.
- 3141 ISO TC 184/SC4/WG3 N1089. ISO/DIS 10303-238: Industrial automation systems and
3142 integration Product data representation and exchange Part 238: Application Protocols: Ap-
3143 plication interpreted model for computerized numerical controllers. Geneva, Switzerland,
3144 2004.
- 3145 International Organization for Standardization. ISO 14649: Industrial automation sys-
3146 tems and integration – Physical device control – Data model for computerized numerical
3147 controllers – Part 10: General process data. Geneva, Switzerland, 2004.
- 3148 International Organization for Standardization. ISO 14649: Industrial automation sys-
3149 tems and integration – Physical device control – Data model for computerized numerical
3150 controllers – Part 11: Process data for milling. Geneva, Switzerland, 2000.
- 3151 International Organization for Standardization. ISO 6983/1 – Numerical Control of ma-
3152 chines – Program format and definition of address words – Part 1: Data format for posi-
3153 tioning, line and contouring control systems. Geneva, Switzerland, 1982.
- 3154 Electronic Industries Association. ANSI/EIA-494-B-1992, 32 Bit Binary CL (BCL) and
3155 7 Bit ASCII CL (ACL) Exchange Input Format for Numerically Controlled Machines.
3156 Washington, D.C. 1992.
- 3157 National Aerospace Standard. Uniform Cutting Tests - NAS Series: Metal Cutting Equip-
3158 ment Specifications. Washington, D.C. 1969.
- 3159 International Organization for Standardization. ISO 10303-11: 1994, Industrial automa-
3160 tion systems and integration Product data representation and exchange Part 11: Descrip-
3161 tion methods: The EXPRESS language reference manual. Geneva, Switzerland, 1994.
- 3162 International Organization for Standardization. ISO 10303-21: 1996, Industrial automa-
3163 tion systems and integration – Product data representation and exchange – Part 21: Imple-
3164 mentation methods: Clear text encoding of the exchange structure. Geneva, Switzerland,
3165 1996.
- 3166 H.L. Horton, F.D. Jones, and E. Oberg. Machinery's Handbook. Industrial Press, Inc.

- 3167 New York, 1984.
- 3168 International Organization for Standardization. ISO 841-2001: Industrial automation sys-
3169 tems and integration - Numerical control of machines - Coordinate systems and motion
3170 nomenclature. Geneva, Switzerland, 2001.
- 3171 ASME B5.57: Methods for Performance Evaluation of Computer Numerically Controlled
3172 Lathes and Turning Centers, 1998.
- 3173 ASME/ANSI B5.54: Methods for Performance Evaluation of Computer Numerically Con-
3174 trolled Machining Centers. 2005.
- 3175 OPC Foundation. OPC Unified Architecture Specification, Part 1: Concepts Version 1.00.
3176 July 28, 2006.
- 3177 IEEE STD 1451.0-2007, Standard for a Smart Transducer Interface for Sensors and Ac-
3178 tuators – Common Functions, Communication Protocols, and Transducer Electronic Data
3179 Sheet (TEDS) Formats, IEEE Instrumentation and Measurement Society, TC-9, The In-
3180 stitute of Electrical and Electronics Engineers, Inc., New York, N.Y. 10016, SH99684,
3181 October 5, 2007.
- 3182 IEEE STD 1451.4-1994, Standard for a Smart Transducer Interface for Sensors and Ac-
3183 tuators – Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet
3184 (TEDS) Formats, IEEE Instrumentation and Measurement Society, TC-9, The Institute of
3185 Electrical and Electronics Engineers, Inc., New York, N.Y. 10016, SH95225, December
3186 15, 2004.

3187 B XML Schema Diagrams

3188 B.1 Observations Schema Diagrams

3189 See `Streams` element in `MTConnectStreams` schema.

3190 See `DeviceStream` element in `MTConnectStreams` schema.

3191 See `ComponentStream` element in `MTConnectStreams` schema.

3192 See `Sample` element in `MTConnectStreams` schema.

3193 See `Event` element in `MTConnectStreams` schema.

3194 See `Condition` element in `MTConnectStreams` schema.

3195 B.2 Representation Schema Diagrams

3196 See `AbstTimeSeries` element in `MTConnectStreams` schema.

3197 See `PartCountDiscrete` element in `MTConnectStreams` schema.

3198 See `VariableDataSet` element in `MTConnectStreams` schema.

3199 See `Entry` element in `MTConnectStreams` schema.

3200 See `WorkOffsetTable` element in `MTConnectStreams` schema.

3201 C XML Examples

3202 C.1 DeviceStream Example

Example 5: DeviceStream Example

```

3203 1 <Streams>
3204 2   <DeviceStream name="M12346" uuid="M8010W4194N">
3205 3     <ComponentStream component="Device" name="M12346" componentId="
3206         dl">
3207 4       <Events>
3208 5         <Availability dataItemId="avail" sequence="156" timestamp="
3209         2021-10-01T14:26:38.668505Z">AVAILABLE</Availability>

```

```

3210 6      <AssetChanged assetType="CuttingTool" dataItemId="d1\
3211          textunderscore_asset\textunderscore_chg" sequence="75570"
3212          timestamp="2021-10-07T05:08:53.870206Z">M8010W4194N1
3213          .120</AssetChanged>
3214 7      <AssetRemoved assetType="CuttingTool" dataItemId="d1\
3215          textunderscore_asset\textunderscore_rem" sequence="140"
3216          timestamp="2021-10-01T11:40:08.101461Z">UNAVAILABLE</
3217          AssetRemoved>
3218 8      </Events>
3219 9      </ComponentStream>
3220 10     <ComponentStream component="Controller" name="controller"
3221          componentId="cont">
3222 11         <Events>
3223 12             <EmergencyStop dataItemId="estop" sequence="159" timestamp="
3224 2021-10-01T14:26:38.66869Z">ARMED</EmergencyStop>
3225 13         </Events>
3226 14         <Samples>
3227 15             <AccumulatedTime dataItemId="cut\textunderscore_time"
3228          sequence="75437" timestamp="2021-10-07T05:08:28.221704Z">
3229 1763070.0</AccumulatedTime>
3230 16         </Samples>
3231 17         <Condition>
3232 18             <Unavailable dataItemId="cont\textunderscore_system"
3233          sequence="72" timestamp="2021-10-11T21:04:03.251999Z" type="
3234 19 SYSTEM"/>
3235 19             <Warning dataItemId="cont\textunderscore_system" nativecode=
3236          "313" nativeSeverity="50" sequence="75573" timestamp="
3237          2021-10-07T05:08:58.518317Z" type="LOGIC\textunderscore_
3238          PROGRAM">PALLET ARM DOWN RS. MALF.</Warning>
3239 20         </Condition>
3240 21     </ComponentStream>
3241 22     <ComponentStream component="Path" name="path" componentId="path1
3242          ">
3243 23         <Events>
3244 24             <Execution dataItemId="execution" name="execution" sequence=
3245          "222623" timestamp="2021-10-12T06:04:32.761198Z">INTERRUPTED</
3246          Execution>
3247 25             <VariableDataSet count="2" dataItemId="cvars" sequence="
3248          126513" timestamp="2021-10-12T03:57:31.106559Z">
3249 26                 <Entry key="100">66.3314</Entry>
3250 27                 <Entry key="101">167.2</Entry>
3251 28             </VariableDataSet>
3252 29             <WorkOffsetTable count="2" dataItemId="woffset" sequence="
3253          222101" timestamp="2021-10-12T06:04:11.990531Z">
3254 30                 <Entry key="G53.1">
3255 31                     <Cell key="X">1</Cell>
3256 32                     <Cell key="Y">2</Cell>
3257 33                     <Cell key="Z">3</Cell>
3258 34                 </Entry>
3259 35                 <Entry key="G53.2">
3260 36                     <Cell key="X">4</Cell>

```

```

3261 37         <Cell key="Y">5</Cell>
3262 38         <Cell key="Z">6</Cell>
3263 39     </Entry>
3264 40 </WorkOffsetTable>
3265 41 </Events>
3266 42 <Samples>
3267 43     <CuttingSpeed dataItemId="cspeed" sequence="112" timestamp="
3268 2021-10-07T05:08:28.221704Z" subType="ACTUAL">UNAVAILABLE</
3269 CuttingSpeed>
3270 44 </Samples>
3271 45 <Condition>
3272 46     <Normal dataItemId="path\textunderscore_system" sequence="
3273 153" timestamp="2021-10-11T21:04:03.262845Z" type="SYSTEM"/>
3274 47 </Condition>
3275 48 </ComponentStream>
3276 49 </DeviceStream>
3277 50 </Streams>

```

3278 C.2 Observations made for DataItem Example

Example 6: MTConnectDevices Response Document

```

3279 1 <Components>
3280 2   <Controller id="cont" name="controller">
3281 3     <DataItems>
3282 4       <DataItem category="EVENT" id="estop" name="estop" type="
3283 EMERGENCY\textunderscore_STOP"/>
3284 5     </DataItems>
3285 6   </Controller>
3286 7 </Components>

```

Example 7: MTConnectStreams Response Document

```

3287 1 <ComponentStream component="Controller" name="controller"
3288   componentId="cont">
3289 2   <Events>
3290 3     <EmergencyStop dataItemId="estop" sequence="159" timestamp="
3291 2021-10-01T14:26:38.66869Z">ARMED</EmergencyStop>
3292 4   </Events>
3293 5 </ComponentStream>

```

3294 C.3 Sample Example

Example 8: Sample Example

```

3295 1 <Samples>

```

```

3296 2    <AccumulatedTime dataItemId="cut\textunderscore_time" sequence="
3297        75437" timestamp="2021-10-07T05:08:28.221704Z">1763070.0</
3298        AccumulatedTime>
3299 3    <CuttingSpeed dataItemId="cspeed" sequence="112" timestamp="
3300        2021-10-07T05:08:28.221704Z" subType="ACTUAL">UNAVAILABLE</
3301        CuttingSpeed>
3302 4    </Samples>

```

3303 C.4 Event Example

Example 9: Event Example

```

3304 1    <Events>
3305 2    <Availability dataItemId="avail" sequence="156" timestamp="
3306        2021-10-01T14:26:38.668505Z">AVAILABLE</Availability>
3307 3    <AssetRemoved assetType="CuttingTool" dataItemId="d1\
3308        textunderscore_asset\textunderscore_rem" sequence="140"
3309        timestamp="2021-10-01T11:40:08.101461Z">UNAVAILABLE</
3310        AssetRemoved>
3311 4    </Events>

```

3312 C.5 Condition Example

3313 Condition types are represented differently in XML when compared to Sample and
 3314 Event types. The element name is the condition state of the Condition type in Pascal
 3315 Case. The name of the Condition type is represented by the attribute type.

3316 If the condition state is unavailable then the element name is represented by Unavail-
 3317 able.

Example 10: Condition Example

```

3318 1    <Condition>
3319 2    <Unavailable dataItemId="cont\textunderscore_system" sequence="72"
3320        timestamp="2021-10-11T21:04:03.251999Z" type="SYSTEM"/>
3321 3    <Normal dataItemId="path\textunderscore_system" sequence="153"
3322        timestamp="2021-10-11T21:04:03.262845Z" type="SYSTEM"/>
3323 4    <Warning dataItemId="cont\textunderscore_system" nativecode="313"
3324        nativeSeverity="50" sequence="75573" timestamp="2021-10-07T05
3325        :08:58.518317Z" type="LOGIC\textunderscore_PROGRAM">PALLET ARM
3326        DOWN RS. MALF.</Warning>
3327 5    </Condition>

```

3328 C.6 DataSet Example

Example 11: DataSet Example

```

3329 1 <Events>
3330 2   <VariableDataSet count="2" dataItemId="cvars" sequence="126513"
3331   timestamp="2021-10-12T03:57:31.106559Z">
3332 3     <Entry key="100">66.3314</Entry>
3333 4     <Entry key="101">167.2</Entry>
3334 5   </VariableDataSet>
3335 6 </Events>

```

3336 C.7 Table Example

Example 12: Table Example

```

3337 1 <Events>
3338 2   <WorkOffsetTable count="2" dataItemId="woffset" sequence="222101"
3339   timestamp="2021-10-12T06:04:11.990531Z">
3340 3     <Entry key="G53.1">
3341 4       <Cell key="X">1</Cell>
3342 5       <Cell key="Y">2</Cell>
3343 6       <Cell key="Z">3</Cell>
3344 7     </Entry>
3345 8     <Entry key="G53.2">
3346 9       <Cell key="X">4</Cell>
3347 10      <Cell key="Y">5</Cell>
3348 11      <Cell key="Z">6</Cell>
3349 12    </Entry>
3350 13   </WorkOffsetTable>
3351 14 </Events>

```