



**MTConnect<sup>®</sup> Standard**  
**Part 3.0 – Observation Information Model**  
**Version 2.2.0**

Prepared for: MTConnect Institute  
Prepared from: MTConnectSysMLModel.xml  
Prepared on: August 8, 2023

MTConnect<sup>®</sup> 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](http://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](http://MTConnectSysMLModel.xml)

# Table of Contents

<b>1</b>	<b>Purpose of This Document</b>	<b>2</b>
<b>2</b>	<b>Terminology and Conventions</b>	<b>3</b>
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
<b>3</b>	<b>Observation Information Model</b>	<b>30</b>
3.1	DeviceStream . . . . .	31
3.1.1	Value Properties of DeviceStream . . . . .	31
3.1.2	Part Properties of DeviceStream . . . . .	32
3.2	ComponentStream . . . . .	33
3.2.1	Value Properties of ComponentStream . . . . .	33
3.2.2	Reference Properties of ComponentStream . . . . .	34
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
<b>4</b>	<b>Representations</b>	<b>48</b>
4.1	Representation . . . . .	48
4.2	Value . . . . .	48
4.3	TimeSeries . . . . .	49
4.3.1	Value Properties of TimeSeries . . . . .	50
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 . . . . .	54
4.6.1	Value Properties of Table . . . . .	55
4.6.2	Part Properties of Table . . . . .	56
4.7	Entry . . . . .	56

4.7.1	Value Properties of Entry . . . . .	57
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 . . . . .	59
<b>5</b>	<b>Observation Types</b>	<b>60</b>
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	ActuatorState . . . . .	62
5.2.4	AdapterSoftwareVersion . . . . .	63
5.2.5	AdapterURI . . . . .	63
5.2.6	<<deprecated>>Alarm . . . . .	63
5.2.7	AlarmLimit . . . . .	64
5.2.8	Application . . . . .	65
5.2.9	AssetChanged . . . . .	65
5.2.10	AssetCount . . . . .	66
5.2.11	AssetRemoved . . . . .	66
5.2.12	Availability . . . . .	67
5.2.13	AxisCoupling . . . . .	67
5.2.14	AxisFeedrateOverride . . . . .	68
5.2.15	AxisInterlock . . . . .	68
5.2.16	AxisState . . . . .	69
5.2.17	BatteryState . . . . .	69
5.2.18	Block . . . . .	70
5.2.19	BlockCount . . . . .	70
5.2.20	CharacteristicPersistentId . . . . .	70
5.2.21	CharacteristicStatus . . . . .	70
5.2.22	ChuckInterlock . . . . .	71
5.2.23	ChuckState . . . . .	72
5.2.24	ClockTime . . . . .	72
5.2.25	<<deprecated>>Code . . . . .	72
5.2.26	ComponentData . . . . .	73

5.2.27	CompositionState . . . . .	73
5.2.28	ConnectionStatus . . . . .	75
5.2.29	ControlLimit . . . . .	75
5.2.30	ControllerMode . . . . .	76
5.2.31	ControllerModeOverride . . . . .	77
5.2.32	CoupledAxes . . . . .	78
5.2.33	CycleCount . . . . .	79
5.2.34	DateCode . . . . .	80
5.2.35	DeactivationCount . . . . .	80
5.2.36	DeviceAdded . . . . .	81
5.2.37	DeviceChanged . . . . .	82
5.2.38	DeviceRemoved . . . . .	82
5.2.39	DeviceUuid . . . . .	83
5.2.40	Direction . . . . .	83
5.2.41	DoorState . . . . .	84
5.2.42	EmergencyStop . . . . .	84
5.2.43	EndOfBar . . . . .	85
5.2.44	EquipmentMode . . . . .	86
5.2.45	Execution . . . . .	87
5.2.46	FeatureMeasurement . . . . .	88
5.2.47	FeaturePersistitentId . . . . .	90
5.2.48	Firmware . . . . .	90
5.2.49	FixtureId . . . . .	91
5.2.50	FunctionalMode . . . . .	91
5.2.51	Hardness . . . . .	92
5.2.52	Hardware . . . . .	93
5.2.53	HostName . . . . .	93
5.2.54	LeakDetect . . . . .	93
5.2.55	Library . . . . .	94
5.2.56	<<deprecated>>Line . . . . .	95
5.2.57	LineLabel . . . . .	95
5.2.58	LineNumber . . . . .	95
5.2.59	LoadCount . . . . .	96
5.2.60	LockState . . . . .	97
5.2.61	MTConnectVersion . . . . .	97
5.2.62	MaintenanceList . . . . .	97
5.2.63	Material . . . . .	99
5.2.64	MaterialLayer . . . . .	99
5.2.65	MeasurementType . . . . .	99
5.2.66	MeasurementUnits . . . . .	99
5.2.67	MeasurementValue . . . . .	100
5.2.68	Message . . . . .	100

5.2.69	Network . . . . .	100
5.2.70	NetworkPort . . . . .	101
5.2.71	OperatingMode . . . . .	101
5.2.72	OperatingSystem . . . . .	102
5.2.73	OperatorId . . . . .	103
5.2.74	PalletId . . . . .	103
5.2.75	PartCount . . . . .	103
5.2.76	PartCountType . . . . .	104
5.2.77	PartDetect . . . . .	104
5.2.78	PartGroupId . . . . .	105
5.2.79	PartId . . . . .	105
5.2.80	PartKindId . . . . .	106
5.2.81	<<deprecated>>PartNumber . . . . .	106
5.2.82	PartProcessingState . . . . .	106
5.2.83	PartStatus . . . . .	108
5.2.84	PartUniqueId . . . . .	108
5.2.85	PathFeedrateOverride . . . . .	109
5.2.86	PathMode . . . . .	109
5.2.87	PowerState . . . . .	110
5.2.88	<<deprecated>>PowerStatus . . . . .	110
5.2.89	ProcessAggregateId . . . . .	111
5.2.90	ProcessKindId . . . . .	111
5.2.91	ProcessOccurrenceId . . . . .	112
5.2.92	ProcessState . . . . .	112
5.2.93	ProcessTime . . . . .	113
5.2.94	Program . . . . .	113
5.2.95	ProgramComment . . . . .	114
5.2.96	ProgramEdit . . . . .	115
5.2.97	ProgramEditName . . . . .	115
5.2.98	ProgramHeader . . . . .	115
5.2.99	ProgramLocation . . . . .	116
5.2.100	ProgramLocationType . . . . .	116
5.2.101	ProgramNestLevel . . . . .	117
5.2.102	RotaryMode . . . . .	117
5.2.103	RotaryVelocityOverride . . . . .	118
5.2.104	Rotation . . . . .	118
5.2.105	SensorAttachment . . . . .	118
5.2.106	SensorState . . . . .	118
5.2.107	SerialNumber . . . . .	119
5.2.108	SpecificationLimit . . . . .	120
5.2.109	SpindleInterlock . . . . .	120
5.2.110	ToolAssetId . . . . .	121

5.2.111	ToolGroup	121
5.2.112	<<deprecated>>ToolId	121
5.2.113	ToolNumber	121
5.2.114	ToolOffset	121
5.2.115	ToolOffsets	122
5.2.116	TransferCount	122
5.2.117	Translation	123
5.2.118	Uncertainty	123
5.2.119	UncertaintyType	123
5.2.120	UnloadCount	124
5.2.121	User	125
5.2.122	ValveState	125
5.2.123	Variable	126
5.2.124	WaitState	126
5.2.125	Wire	127
5.2.126	WorkOffset	127
5.2.127	WorkOffsets	127
5.2.128	WorkholdingId	128
5.3	Sample Types	128
5.3.1	Acceleration	128
5.3.2	AccumulatedTime	128
5.3.3	<<deprecated>>Amperage	129
5.3.4	AmperageAC	129
5.3.5	AmperageDC	130
5.3.6	Angle	130
5.3.7	AngularAcceleration	131
5.3.8	AngularDeceleration	131
5.3.9	AngularVelocity	132
5.3.10	AssetUpdateRate	132
5.3.11	AxisFeedrate	132
5.3.12	BatteryCapacity	133
5.3.13	BatteryCharge	134
5.3.14	CapacityFluid	134
5.3.15	CapacitySpatial	134
5.3.16	ChargeRate	134
5.3.17	Concentration	135
5.3.18	Conductivity	135
5.3.19	CuttingSpeed	135
5.3.20	Deceleration	136
5.3.21	Density	136
5.3.22	DepositionAccelerationVolumetric	136
5.3.23	DepositionDensity	137

5.3.24	DepositionMass	137
5.3.25	DepositionRateVolumetric	138
5.3.26	DepositionVolume	138
5.3.27	DewPoint	138
5.3.28	Diameter	139
5.3.29	DischargeRate	139
5.3.30	Displacement	139
5.3.31	DisplacementAngular	139
5.3.32	DisplacementLinear	140
5.3.33	ElectricalEnergy	140
5.3.34	EquipmentTimer	140
5.3.35	FillLevel	141
5.3.36	Flow	141
5.3.37	FollowingError	141
5.3.38	FollowingErrorAngular	142
5.3.39	FollowingErrorLinear	142
5.3.40	Frequency	142
5.3.41	<<deprecated>>GlobalPosition	143
5.3.42	GravitationalAcceleration	143
5.3.43	GravitationalForce	143
5.3.44	HumidityAbsolute	143
5.3.45	HumidityRelative	144
5.3.46	HumiditySpecific	144
5.3.47	Length	145
5.3.48	<<deprecated>>Level	145
5.3.49	LinearForce	145
5.3.50	Load	146
5.3.51	Mass	146
5.3.52	ObservationUpdateRate	146
5.3.53	Openness	146
5.3.54	Orientation	146
5.3.55	PH	147
5.3.56	PathFeedrate	147
5.3.57	PathFeedratePerRevolution	148
5.3.58	PathPosition	148
5.3.59	Position	149
5.3.60	PositionCartesian	149
5.3.61	PowerFactor	150
5.3.62	Pressure	150
5.3.63	PressureAbsolute	150
5.3.64	PressurizationRate	150
5.3.65	ProcessTimer	151



5.3.66	Resistance	151
5.3.67	RotaryVelocity	151
5.3.68	SettlingError	152
5.3.69	SettlingErrorAngular	152
5.3.70	SettlingErrorLinear	153
5.3.71	SoundLevel	153
5.3.72	<<deprecated>>SpindleSpeed	154
5.3.73	Strain	154
5.3.74	Temperature	154
5.3.75	Tension	155
5.3.76	Tilt	155
5.3.77	Torque	155
5.3.78	Velocity	155
5.3.79	Viscosity	155
5.3.80	VoltAmpere	155
5.3.81	VoltAmpereReactive	156
5.3.82	<<deprecated>>Voltage	156
5.3.83	VoltageAC	157
5.3.84	VoltageDC	157
5.3.85	VolumeFluid	158
5.3.86	VolumeSpatial	158
5.3.87	Wattage	159
5.3.88	XDimension	159
5.3.89	YDimension	160
5.3.90	ZDimension	160
<b>6</b>	<b>Profile</b>	<b>161</b>
6.1	DataTypes	161
6.1.1	boolean	161
6.1.2	ID	161
6.1.3	string	161
6.1.4	float	161
6.1.5	datetime	162
6.1.6	integer	162
6.1.7	xlinktype	162
6.1.8	xslang	162
6.1.9	SECOND	162
6.1.10	IDREF	162
6.1.11	xlinkhref	162
6.1.12	x509	163
6.1.13	int32	163
6.1.14	int64	163

6.1.15	version	163
6.1.16	uint32	163
6.1.17	uint64	163
6.1.18	binary	163
6.1.19	double	164
6.2	Stereotypes	164
6.2.1	organizer	164
6.2.2	deprecated	164
6.2.3	extensible	164
6.2.4	informative	164
6.2.5	valueType	164
6.2.6	normative	164
6.2.7	observes	165
<b>Appendices</b>		<b>167</b>
A	Bibliography	167
B	XML Schema Diagrams	169
B.1	Observations Schema Diagrams	169
B.2	Representation Schema Diagrams	169
C	XML Examples	169
C.1	DeviceStream Example	169
C.2	Observations made for DataItem Example	171
C.3	Sample Example	171
C.4	Event Example	172
C.5	Condition Example	172
C.6	DataSet Example	173
C.7	Table Example	173

# 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** . . . . . 161
- Figure 14:Stereotypes** . . . . . 166

## List of Tables

<b>Table 1: Value Properties of DeviceStream</b>	32
<b>Table 2: Part Properties of DeviceStream</b>	32
<b>Table 3: Value Properties of ComponentStream</b>	33
<b>Table 4: Reference Properties of ComponentStream</b>	34
<b>Table 5: Value Properties of Observation</b>	37
<b>Table 6: Value Properties of Sample</b>	40
<b>Table 7: Value Properties of Event</b>	42
<b>Table 8: Value Properties of Condition</b>	44
<b>Table 9: Value Properties of TimeSeries</b>	50
<b>Table 10: Value Properties of DataSet</b>	53
<b>Table 11: Part Properties of DataSet</b>	53
<b>Table 12: Value Properties of Table</b>	55
<b>Table 13: Part Properties of Table</b>	56
<b>Table 14: Value Properties of Entry</b>	57
<b>Table 15: Value Properties of TableEntry</b>	57
<b>Table 16: Part Properties of TableEntry</b>	58
<b>Table 17: Value Properties of Cell</b>	59
<b>Table 18: Value Properties of Alarm</b>	63
<b>Table 19: Value Properties of AssetChanged</b>	66
<b>Table 20: Value Properties of AssetRemoved</b>	66
<b>Table 21: Value Properties of DeviceAdded</b>	81
<b>Table 22: Value Properties of DeviceChanged</b>	82
<b>Table 23: Value Properties of DeviceRemoved</b>	82
<b>Table 24: Value Properties of Message</b>	100

## 1 **1 Purpose of This Document**

2 This document, *MTCConnect Standard: Part 3.0 - Observation Information Model* of the  
3 MTCConnect Standard, establishes the rules and terminology that describes the informa-  
4 tion returned by an *MTCConnect Agent* from a piece of equipment. The term (Observation  
5 Information Model also defines, in *MTCConnect 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 *MTCConnect Standard: Part 3.0 - Observation Information Model* is not a stand-alone doc-  
9 ument. This document is used in conjunction with *MTCConnect Standard Part 1.0 - Fun-*  
10 *damentals* which defines the fundamentals of the operation of the MTCConnect Standard  
11 and *MTCConnect 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: *MTCConnect 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 MTCConnect 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 General Terms

#### 23 ***adapter***

24 optional piece of hardware or software that transforms information provided by a  
25 piece of equipment into a form that can be received by an *agent*.

#### 26 ***agent***

27 software that collects data published from one or more piece(s) of equipment, or-  
28 ganizes that data in a structured manner, and responds to requests for data from  
29 client software systems by providing a structured response in the form of a *response*  
30 *document* that is constructed using the *semantic data model* of a Standard.

#### 31 ***alarm limit***

32 limit used to trigger warning or alarm indicators.

#### 33 ***application***

34 software or a program that is specific to the solution of an application problem.  
35 *Ref ISO/IEC 20944-1:2013*

#### 36 ***archetype***

37 *archetype* provides the requirements, constraints, and common properties for a type  
38 of *Asset*.

#### 39 ***asset buffer***

40 *buffer* for *Assets*.

#### 41 ***attachment***

42 connection by which one thing is associated with another.

#### 43 ***buffer***

44 section of an *agent* that provides storage for information published from pieces of  
45 equipment.

46 ***cartesian coordinate system***

47 3D orthogonal coordinate system [(ISO/IEC 19794-5:2011)en].

48 ***characteristic***49 control placed on an element of a *feature* such as its size, location, or form, which  
50 may be a specification limit, a nominal with tolerance, or some other numerical or  
51 non-numerical control. *Ref QIF 3.0 3.4.29. Ref AS9102-B.*52 ***client***53 *application* that sends *request* for information to an *agent*.54 Note: Examples include software applications or a function that imple-  
55 ments the *request* portion of an *interface interaction model*.56 ***combined standard uncertainty***57 *standard uncertainty* of the result of a measurement when that result is obtained  
58 from the values of a number of other quantities, equal to the positive square root of a  
59 sum of terms, the terms being the variances or covariances of these other quantities  
60 weighted according to how the measurement result varies with changes in these  
61 quantities. *Ref JCGM 100:2008 2.3.4*62 ***controlled vocabulary***

63 restricted set of values that may be published for an observation.

64 ***data dictionary***65 listing of standardized terms and definitions used in *MTCConnect Information Model*.66 ***data model***67 organizes elements of data and standardizes how they relate to one another and to  
68 the properties of real-world entities.69 ***data set***70 *key-value pairs* where each entry is uniquely identified by the *key*.71 ***data source***72 piece of equipment that can produce data that is published to an *agent*.73 ***deprecated***74 indication that specific content in an *MTCConnect Document* is currently usable but  
75 is regarded as being obsolete or superseded.

76 ***deprecation warning***

77 indication that specific content in an *MTConnect Document* may be changed to *depre-*  
78 *recated* in a future release of the standard.

79 ***document***

80 piece of written, printed, or electronic matter that provides information or evidence  
81 that serves as an official record.

82 ***electric current***

83 rate of flow of electric charge.

84 ***element***

85 constituent part or a basic unit of identifiable and definable data.

86 ***extensible***

87 ability for an implementer to extend *MTConnect Information Model* by adding con-  
88 tent not currently addressed in the MTConnect Standard.

89 ***feature***

90 topological entity(ies) or design requirements related to a geometric model. *Ref QIF*  
91 *3.0-3.4.59*

92 ***force***

93 push or pull on a mass which results in an acceleration.

94 ***heartbeat***

95 function that indicates to a *client* that the communications connection to an *agent* is  
96 still viable during times when there is no new data available to report often referred  
97 to as a “keep alive” message.

98 ***higher level***

99 nested element that is above a lower level element.

100 ***implementation***

101 specific instantiation of the MTConnect Standard.

102 ***information model***

103 rules, relationships, and terminology that are used to define how information is struc-  
104 tured.



105 ***instance***

106 describes a set of *streaming data* in an *agent*. Each time an *agent* is restarted with  
107 an empty *buffer*, data placed in the *buffer* represents a new *instance* of the *agent*.

108 ***interaction model***

109 model that defines how information is exchanged across an *interface* to enable in-  
110 teractions between independent systems.

111 ***interface***

112 means by which communication is achieved between independent systems.

113 ***key***

114 unique identifier in a *key-value pair* association.

115 ***key-value pair***

116 association between an identifier referred to as the *key* and a value which taken  
117 together create a *key-value pair*.

118 ***location***

119 place or named space associated with an object or that can be occupied by an object.

120 ***lower camel case***

121 first word is lowercase and the remaining words are capitalized and all spaces be-  
122 tween words are removed.

123 ***lower level***

124 nested element that is below a higher level element.

125 ***lower limit***

126 lower conformance boundary for a variable.

127 ***lower warning***

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

129 ***major***

130 identifier representing a consistent set of functionalities defined by the MTConnect  
131 Standard.

132 ***maximum***

133 numeric upper constraint.

134 ***message***

135 communication in writing, in speech, or by signals.

136 ***metadata***

137 data that provides information about other data.

138 ***minimum***

139 numeric lower constraint.

140 ***minor***

141 identifier representing a specific set of functionalities defined by the MTConnect  
142 Standard.

143 ***nominal***

144 ideal or desired value for a variable.

145 ***organize***

146 act of containing and owning one or more elements.

147 ***organizer***

148 entity that *organizes* one or more elements.

149 ***parameter***

150 variable that must be given a value during the execution of a program or a commu-  
151 nications command.

152 ***part***

153 discrete item that has both defined and measurable physical characteristics including  
154 mass, material, and features, and is created by applying one or more manufacturing  
155 process steps to a workpiece

156 ***pascal case***

157 first letter of each word is capitalized and the remaining letters are in lowercase. All  
158 space is removed between letters

159 ***persistence***

160 method for retaining or restoring information.

161 ***position***

162 *location* that is represented by a point in space relative to a reference.

163 ***probe***

164 instrument commonly used for measuring the physical geometrical characteristics  
165 of an object.

166 ***profile***

167 extends a reference metamodel (such as Unified Modeling Language (UML)) by  
168 allowing to adapt or customize the metamodel with constructs that are specific to a  
169 particular domain, platform, or a software development method.

170 ***requester***

171 entity that initiates a *request* for information in a communications exchange.

172 ***reset***

173 act of reverting back the accumulated value or statistic to their initial value.

174 Note: An *Observation* with a *data set* representation removes all *key-*  
175 *value pairs*, setting the *data set* to an empty set.

176 ***responder***

177 entity that responds to a *request* for information in a communications exchange.

178 ***response document***

179 electronic *document* published by an *MTCConnect Agent* in response to a *probe re-*  
180 *quest, current request, sample request* or *asset request*.

181 ***revision***

182 supplemental identifier representing only organizational or editorial changes to a  
183 *minor* version document with no changes in the functionality described in that doc-  
184 ument.

185 ***schema***

186 definition of the structure, rules, and vocabularies used to define the information  
187 published in an electronic document.

188 ***semantic data model***

189 methodology for defining the structure and meaning for data in a specific logical  
190 way that can be interpreted by a software system.

191 ***sensing element***

192 mechanism that provides a signal or measured value.

193 ***sequence number***

194 primary key identifier used to manage and locate a specific piece of *streaming data*  
195 in an *agent*.

196 ***specification limit***

197 limit defining a range of values designating acceptable performance for a variable.

198 ***spindle***

199 mechanism that provides rotational capabilities to a piece of equipment.

200 Note: Typically used for either work holding, materials or cutting tools.

201 ***standard***

202 *document* established by consensus that provides rules, guidelines, or characteristics  
203 for activities or their results.. *Ref ISO/IEC Guide 2:2004*

204 ***standard uncertainty***

205 *uncertainty* of the result of a measurement expressed as a standard deviation. *Ref JCGM*  
206 *100:2008 2.3.1*

207 ***stereotype***

208 defines how an existing UML metaclass may be extended as part of a *profile*.

209 ***subtype***

210 secondary or subordinate type of categorization or classification of information.

211 ***table***

212 two dimensional set of values given by a set of *key-value pairs table entries*.

213 ***table cell***

214 subdivision of a *table entry* representing a singular value.

215 ***table entry***

216 subdivision of a *table* containing a set of *key-value pairs* representing *table cells*.

217 ***top level***

218 element that represents the most significant physical or logical functions of a piece  
219 of equipment.

220 ***type***

221 classification or categorization of information.

222 ***uncertainty***

223 uncertainty (of measurement) parameter, associated with the result of a measure-  
 224 ment, that characterizes the dispersion of the values that could reasonably be at-  
 225 tributed to the measurand. *Ref JCGM 100:2008 2.2.3*

226 Note: Use of the term uncertainty refers to uncertainty of measurement.

227 ***upper limit***

228 upper conformance boundary for a variable.

229 ***upper warning***

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

231 ***version***

232 unique identifier of the administered item. *Ref ISO/IEC 11179-:2015*

233 **2.2 Information Model Terms**234 ***Asset Information Model***

235 *information model* that provides semantic models for *Assets*.

236 ***Device Information Model***

237 *information model* that describes the physical and logical configuration for a piece  
 238 of equipment and the data that may be reported by that equipment.

239 ***Error Information Model***

240 *information model* that describes the *response document* returned by an *agent* when  
 241 it encounters an error while interpreting a *request* for information from a *client* or  
 242 when an *agent* experiences an error while publishing the *response* to a *request* for  
 243 information.

244 ***MTCConnect Information Model***

245 *information model* that defines the semantics of the MTCConnect Standard.

246 ***Observation Information Model***

247 *information model* that describes the *streaming data* reported by a piece of equip-  
 248 ment.

## 249 2.3 Protocol Terms

### 250 *asset request*

251 *HTTP Request* to the *agent* regarding *Assets*.

### 252 *current request*

253 *request* to an *agent* to produce an *MTConnectStreams Response Document* contain-  
254 ing the *Observation Information Model* for a snapshot of the latest observations at  
255 the moment of the *request* or at a given *sequence number*.

### 256 *data streaming*

257 method for an *agent* to provide a continuous stream of information in response to a  
258 single *request* from a *client*.

### 259 *MTConnect Request*

260 *request* for information issued from a *client* to an *MTConnect Agent*.

### 261 *MTConnect Response Document*

262 *response document* published by an *MTConnect Agent*.

### 263 *MTConnectAssets Response Document*

264 *response document* published by an *MTConnect Agent* in response to an *asset re-*  
265 *quest*.

### 266 *MTConnectDevices Response Document*

267 *response document* published by an *MTConnect Agent* in response to a *probe re-*  
268 *quest*.

### 269 *MTConnectErrors Response Document*

270 *response document* published by an *MTConnect Agent* whenever it encounters an  
271 error while interpreting an *MTConnect Request*.

### 272 *MTConnectStreams Response Document*

273 *response document* published by an *MTConnect Agent* in response to a *current re-*  
274 *quest* or a *sample request*.

### 275 *probe request*

276 *request* to an *agent* to produce an *MTConnectDevices Response Document* contain-  
277 ing the *Device Information Model*.

278 ***protocol***

279 set of rules that allow two or more entities to transmit information from one to the  
280 other.

281 ***publish***

282 sending of messages in a *publish and subscribe* pattern.

283 ***publish and subscribe***

284 asynchronous communication method in which messages are exchanged between  
285 applications without knowing the identity of the sender or recipient.

286 Note: In the MTConnect Standard, a communications messaging pattern  
287 that may be used to publish *streaming data* from an *agent*.

288 ***request***

289 communications method where a *client* transmits a message to an *agent*. That mes-  
290 sage instructs the *agent* to respond with specific information.

291 ***request and response***

292 communications pattern that supports the transfer of information between an *agent*  
293 and a *client*.

294 ***response***

295 response *interface* which responds to a *request*.

296 ***sample request***

297 *request* to an *agent* to produce an *MTConnectStreams Response Document* contain-  
298 ing the *Observation Information Model* for a set of timestamped observations made  
299 by *Components*.

300 ***streaming data***

301 observations published by a piece of equipment defined by the equipment metadata.

302 ***subscribe***

303 receiving messages in a *publish and subscribe* pattern.

304 ***transport protocol***

305 set of capabilities that provide the rules and procedures used to transport information  
306 between an *agent* and a client software application through a physical connection.

## 307 2.4 HTTP Terms

### 308 **HTTP Body**

309 data bytes transmitted in an HTTP transaction message immediately following the  
310 headers. *Ref IETF:RFC-2616*

### 311 **HTTP Error Message**

312 response provided by an *agent* indicating that an *HTTP Request* is incorrectly for-  
313 matted or identifies that the requested data is not available from the *agent*. *Ref IETF:RFC-*  
314 *2616*

### 315 **HTTP Header**

316 header of either an *HTTP Request* from a *client* or an *HTTP Response* from an *agent*.  
317 *Ref IETF:RFC-2616*

### 318 **HTTP Header Field**

319 components of the header section of request and response messages in an HTTP  
320 transaction. *Ref IETF:RFC-2616*

### 321 **HTTP Message**

322 consist of requests from client to server and responses from server to client. *Ref IETF:RFC-*  
323 *2616*

324 Note: In MTConnect Standard, it describes the information that is ex-  
325 changed between an *agent* and a *client*.

### 326 **HTTP Messaging**

327 *interface* for information exchange functionality. *Ref IETF:RFC-2616*

### 328 **HTTP Method**

329 portion of a command in an *HTTP Request* that indicates the desired action to be  
330 performed on the identified resource; often referred to as verbs. *Ref IETF:RFC-*  
331 *2616*

### 332 **HTTP Query**

333 portion of a request for information that more precisely defines the specific informa-  
334 tion to be published in response to the request. *Ref IETF:RFC-2616*

### 335 **HTTP Request**

336 request message from a client to a server includes, within the first line of that mes-  
337 sage, the method to be applied to the resource, the identifier of the resource, and the  
338 protocol version in use. *Ref IETF:RFC-2616*



339 Note: In MTConnect Standard, a request issued by a *client* to an *agent*  
340 requesting information defined in the *HTTP Request Line*.

### 341 ***HTTP Request Line***

342 begins with a method token, followed by the Request-URI and the protocol version,  
343 and ending with CRLF. A CRLF is allowed in the definition of TEXT only as part  
344 of a header field continuation. *Ref IETF:RFC-2616*

345 Note: the first line of an *HTTP Request* describing a specific *response*  
346 *document* to be published by an *agent*.

### 347 ***HTTP Request Method***

348 indicates the method to be performed on the resource identified by the Request-URI.  
349 *Ref IETF:RFC-2616*

### 350 ***HTTP Request URI***

351 Uniform Resource Identifier that identifies the resource upon which to apply the  
352 request. *Ref IETF:RFC-2616*

### 353 ***HTTP Response***

354 after receiving and interpreting a request message, a server responds with an HTTP  
355 response message. *Ref IETF:RFC-2616*

356 Note: In MTConnect Standard, the information published from an *agent*  
357 in reply to an *HTTP Request*.

### 358 ***HTTP Server***

359 server that accepts *HTTP Request* from *client* and publishes *HTTP Response* as a  
360 reply to those *HTTP Request*. *Ref IETF:RFC-2616*

### 361 ***HTTP Status Code***

362 3-digit integer result code of the attempt to understand and satisfy the request.  
363 *Ref IETF:RFC-2616*

### 364 ***HTTP Version***

365 version of the HTTP protocol. *Ref IETF:RFC-2616*

## 366 2.5 XML Terms

### 367 ***abstract element***

368 element that defines a set of common characteristics that are shared by a group of  
369 elements. An abstract entity cannot appear in a document. In a specific implemen-  
370 tation, an abstract entity is replaced by a derived element that is itself not an abstract  
371 entity. The characteristics for the derived element are inherited from the abstract  
372 entity.

### 373 ***attribute***

374 additional information or property for an *element*.

### 375 ***child element***

376 *element* of a data modeling structure that illustrates the relationship between itself  
377 and the higher-level *parent element* within which it is contained.

### 378 ***document body***

379 portion of the content of an *MTCConnect Response Document* that is defined by the  
380 relative *MTCConnect Information Model*. The *document body* contains the *structural*  
381 *elements* and *Observations* or *DataItems* reported in a *response document*.

### 382 ***document header***

383 portion of the content of an *MTCConnect Response Document* that provides infor-  
384 mation from an *agent* defining version information, storage capacity, protocol, and  
385 other information associated with the management of the data stored in or retrieved  
386 from the *agent*.

### 387 ***element name***

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

### 390 ***namespace***

391 organizes information into logical groups.

### 392 ***parent element***

393 *element* of a data modeling structure that illustrates the relationship between itself  
394 and the lower-level *child element*.

### 395 ***root element***

396 first *structural element* provided in a *response document* encoded using XML.

397 ***structural element***

398 *element* that organizes information that represents the physical and logical parts and  
 399 sub-parts of a piece of equipment.

400 ***XML Document***

401 structured text file encoded using Extensible Markup Language (XML).

402 ***XML Schema***

403 *schema* defining a specific document encoded in XML.

404 **2.6 MTConnect Terms**405 ***Asset***

406 asset that is used by the manufacturing process to perform tasks.

407 Note 1 to entry: An *Asset* relies upon an *Device* to provide observations  
 408 and information about itself and the *Device* revises the information to  
 409 reflect changes to the *Asset* during their interaction. Examples of *Assets*  
 410 are cutting tools, Part Information, Manufacturing Processes, Fixtures,  
 411 and Files.

412 Note 2 to entry: A singular `assetId`, *Asset* uniquely identifies an  
 413 *Asset* throughout its lifecycle and is used to track and relate the *Asset* to  
 414 other *Devices* and entities.

415 Note 3 to entry: *Assets* are temporally associated with a device and can  
 416 be removed from the device without damage or alteration to its primary  
 417 functions.

418 ***Component***

419 engineered system part of a *Device* composed of zero or more *Components*

420 ***Composition***

421 *Component* belonging to a *Component* and not composed of any *Components*.

422 ***Configuration***

423 configuration for a *Component*

424 ***DataItem***

425 observable observed by a *Component* that may make *Observations*

426 ***Device***

427 *Component* not belonging to any *Component* that may have assets

428 ***MTCConnect Agent***

429 *agent* for the *MTCConnect Information Model*.

430 ***MTCConnect Document***

431 *document* that represents a Part(s) of the MTCConnect Standard.

432 ***MTCConnect Event***

433 observation of either a state or discrete value of the *Component*.

434 ***MTCConnect Interface***

435 *interaction model* for interoperability between pieces of equipment.

436 ***Observation***

437 observation that provides telemetry data for a *DataItem*.

438 **2.7 Acronyms**

439 ***2D***

440 two-dimensional

441 ***3D***

442 three-dimensional

443 ***AI***

444 artificial intelligence

445 ***ALM***

446 application lifecycle management

447 ***AMT***

448 The Association for Manufacturing Technology

449 ***ANSI***

450 American National Standards Institute

451	<b><i>AP</i></b>
452	Application Protocol
453	<b><i>API</i></b>
454	application programming interface
455	<b><i>ASME</i></b>
456	American Society of Mechanical Engineers
457	<b><i>ASTM</i></b>
458	American Society for Testing and Materials
459	<b><i>AWS</i></b>
460	American Welding Society
461	<b><i>BDD</i></b>
462	block definition diagram
463	<b><i>BOM</i></b>
464	bill of materials
465	<b><i>BST</i></b>
466	Board on Standardization and Testing
467	<b><i>C&amp;R</i></b>
468	cause and remedy
469	<b><i>CA</i></b>
470	certificate authority
471	<b><i>CAD</i></b>
472	computer-aided design
473	<b><i>CAE</i></b>
474	computer-aided engineering
475	<b><i>CAI</i></b>
476	computer-aided inspection
477	<b><i>CAM</i></b>
478	computer-aided manufacturing

- 479 **CAx**
- 480 computer-aided technologies
- 481 **CDATA**
- 482 Character Data
- 483 **CFD**
- 484 computational fluid dynamics
- 485 **CM**
- 486 configuration management
- 487 **CMS**
- 488 coordinate-measurement system
- 489 **CNC**
- 490 Computer Numerical Controller
- 491 **CNRI**
- 492 Corporation for National Research Initiatives
- 493 **CPM**
- 494 Core Product Model
- 495 **CPM2**
- 496 Revised Core Product Model
- 497 **CPSC**
- 498 Consumer Product Safety Commission
- 499 **cUAV**
- 500 configurable unmanned aerial vehicle
- 501 **DARPA**
- 502 Defense Advanced Research Projects Agency
- 503 **DER**
- 504 designated-engineering representative
- 505 **DFM**
- 506 design for manufacturing

507	<b><i>DLA</i></b>
508	Defense Logistics Agency
509	<b><i>DMC</i></b>
510	digital manufacturing certificate
511	<b><i>DMSC</i></b>
512	Dimensional Metrology Standards Consortium
513	<b><i>DNS</i></b>
514	Domain Name System
515	<b><i>DoD</i></b>
516	U.S. Department of Defense
517	<b><i>DOI</i></b>
518	Distributed Object Identifier
519	<b><i>DRM</i></b>
520	digital rights management
521	<b><i>ECR</i></b>
522	engineering change request
523	<b><i>ERP</i></b>
524	enterprise resource planning
525	<b><i>FAA</i></b>
526	Federal Aviation Administration
527	<b><i>FAIR</i></b>
528	first article inspection reporting
529	<b><i>FDA</i></b>
530	Food and Drug Administration
531	<b><i>FEA</i></b>
532	finite-element analysis
533	<b><i>GD&amp;T</i></b>
534	geometric dimensions and tolerances

535	<b><i>GID</i></b>
536	global identifier
537	<b><i>HMI</i></b>
538	Human Machine Interface
539	<b><i>HTML</i></b>
540	Hypertext Markup Language
541	<b><i>HTTP</i></b>
542	Hypertext Transfer Protocol
543	<b><i>HTTPS</i></b>
544	Hypertext Transfer Protocol over Secure Sockets Layer
545	<b><i>I/O</i></b>
546	in-out
547	<b><i>ID</i></b>
548	identifier
549	<b><i>IEEE</i></b>
550	Institute of Electrical and Electronics Engineers
551	<b><i>IIoT</i></b>
552	industrial internet of things
553	<b><i>INCOSE</i></b>
554	International Council on Systems Engineering
555	<b><i>IP</i></b>
556	intellectual property
557	<b><i>ISO</i></b>
558	International Standards Organization
559	<b><i>ISS</i></b>
560	International Space Station
561	<b><i>ISV</i></b>
562	Independent Software Vendor



563	<b><i>IT</i></b>
564	information technology
565	<b><i>ITU-T</i></b>
566	Telecommunication Standardization Sector of the International Telecommunication
567	Union
568	<b><i>JSON</i></b>
569	JavaScript Object Notation
570	<b><i>JT</i></b>
571	Jupiter Tessellation
572	<b><i>LHS</i></b>
573	Lifecycle Handler System
574	<b><i>LIFT</i></b>
575	Lifecycle Information Framework and Technology
576	<b><i>LOI</i></b>
577	Lifecycle Object Identifier
578	<b><i>MAC</i></b>
579	media access control
580	<b><i>MADE</i></b>
581	Manufacturing Automation and Design Engineering
582	<b><i>MBD</i></b>
583	model-based definition
584	<b><i>MBE</i></b>
585	Model-Based Enterprise
586	<b><i>MBI</i></b>
587	model-based inspection
588	<b><i>MBM</i></b>
589	model-based manufacturing

- 590 ***MBSD***
- 591       model-based standards development
- 592 ***MBSE***
- 593       model-based systems engineering
- 594 ***MEDALS***
- 595       Military Engineering Data Asset Locator System
- 596 ***MES***
- 597       manufacturing execution system
- 598 ***MOI***
- 599       manufacturing object identifier
- 600 ***MOM***
- 601       Message Orienged Middleware
- 602 ***MQTT***
- 603       Message Queuing Telemetry Transport
- 604 ***MTC***
- 605       Manufacturing Technology Centre
- 606 ***NASA***
- 607       National Aeronautics and Space Administration
- 608 ***NC***
- 609       numerical control
- 610 ***NIST***
- 611       National Institute of Standards and Technology
- 612 ***NMTOKEN***
- 613       Name Token
- 614 ***NNMI***
- 615       National Network of Manufacturing Innovation
- 616 ***NSF***
- 617       National Science Foundation

618	<b><i>NTSC</i></b>
619	National Transportation Safety Board
620	<b><i>OASIS</i></b>
621	Organization for the Advancement of Structured Information Standards
622	<b><i>ODI</i></b>
623	Open Data Institute
624	<b><i>OEM</i></b>
625	original equipment manufacturer
626	<b><i>OOI</i></b>
627	Ocean Observatories Initiative
628	<b><i>OPC</i></b>
629	OLE for Process Control
630	<b><i>OSLC</i></b>
631	Open Services for Lifecycle Collaboration
632	<b><i>OSTP</i></b>
633	Office of Science and Technology Policy
634	<b><i>OT</i></b>
635	operational technology
636	<b><i>OWL</i></b>
637	Ontology Web Language
638	<b><i>PDF</i></b>
639	Portable Document Format
640	<b><i>PDM</i></b>
641	product-data management
642	<b><i>PDQ</i></b>
643	product-data quality
644	<b><i>PHM</i></b>
645	prognosis and health monitoring

646	<b><i>PI</i></b>	
647		principal investigator
648	<b><i>PLC</i></b>	
649		Programmable Logic Controller
650	<b><i>PLCS</i></b>	
651		Product Life Cycle Support
652	<b><i>PLM</i></b>	
653		product lifecycle management
654	<b><i>PLOT</i></b>	
655		product lifecycle of trust
656	<b><i>PMI</i></b>	
657		product and manufacturing information
658	<b><i>PMS</i></b>	
659		Production Management System
660	<b><i>PRC</i></b>	
661		Product Representation Compact
662	<b><i>PSI</i></b>	
663		Physical Science Informatics
664	<b><i>PTAB</i></b>	
665		Primary Trustworthy Digital Repository Authorization Body Ltd.
666	<b><i>QIF</i></b>	
667		Quality Information Framework
668	<b><i>QMS</i></b>	
669		quality management system
670	<b><i>QName</i></b>	
671		Qualified Name
672	<b><i>RDF</i></b>	
673		Resource Description Framework

674	<b>REST</b>
675	Representational State Transfer
676	<b>RII</b>
677	receiving and incoming inspection
678	<b>S/MIME</b>
679	Secure/Multipurpose Internet Mail Extensions
680	<b>SaaS</b>
681	software-as-a-service
682	<b>SAML</b>
683	Security Assertion Markup Language
684	<b>SC</b>
685	Standards Committee
686	<b>SCADA</b>
687	Supervisory Control And Data Acquisition
688	<b>SDO</b>
689	Standards Development Organization
690	<b>SFTP</b>
691	Secure File Transfer Protocol
692	<b>SKOS</b>
693	Simple Knowledge Organization System
694	<b>SLH</b>
695	system lifecycle handler
696	<b>SLR</b>
697	systematic literature review
698	<b>SME</b>
699	small-to-medium enterprise
700	<b>SMOPAC</b>
701	Smart Manufacturing Operations Planning and Control

702	<b><i>SMS Test Bed</i></b>
703	Smart Manufacturing Systems Test Bed
704	<b><i>SOA</i></b>
705	service-oriented architecture
706	<b><i>SPMM</i></b>
707	semantic-based product metamodel
708	<b><i>SSL</i></b>
709	Secure Sockets Layer
710	<b><i>STEP</i></b>
711	Standard for the Exchange of Product Model Data
712	<b><i>STEP AP242</i></b>
713	Standard for the Exchange of Product Model Data Application Protocol 242
714	<b><i>STL</i></b>
715	Stereolithography
716	<b><i>SysML</i></b>
717	Systems Modeling Language
718	<b><i>TCP/IP</i></b>
719	Transmission Control Protocol/Internet Protocol
720	<b><i>TDP</i></b>
721	technical data package
722	<b><i>TLS</i></b>
723	Transport Layer Security
724	<b><i>TSM</i></b>
725	Total System Model
726	<b><i>UA</i></b>
727	Unified Architecture
728	<b><i>UAL</i></b>
729	Unified Architecture Language

730	<b><i>UML</i></b>
731	Unified Modeling Language
732	<b><i>URI</i></b>
733	Uniform Resource Identifier
734	<b><i>URL</i></b>
735	Uniform Resource Locator
736	<b><i>URN</i></b>
737	Uniform Resource Name
738	<b><i>UTC</i></b>
739	Coordinated Universal Time
740	<b><i>UUID</i></b>
741	Universally Unique Identifier
742	<b><i>V&amp;V</i></b>
743	verification and validation
744	<b><i>W3C</i></b>
745	World Wide Web Consortium
746	<b><i>WSN</i></b>
747	Wirth Syntax Notation
748	<b><i>WWW</i></b>
749	World Wide Web
750	<b><i>X.509-PKI</i></b>
751	Public Key Infrastructure
752	<b><i>X.509-PMI</i></b>
753	Privilege Management Infrastructure
754	<b><i>XML</i></b>
755	Extensible Markup Language
756	<b><i>XPath</i></b>
757	XML Path Language
758	<b><i>XSD</i></b>
759	XML Schema Definitions

## 760 **2.8 MTConnect References**

- 761 [MTConnect Part 1.0] *MTConnect Standard Part 1.0 - Fundamentals*. Version 2.0.
- 762 [MTConnect Part 2.0] *MTConnect Standard: Part 2.0 - Device Information Model*. Ver-  
763 sion 2.0.
- 764 [MTConnect Part 3.0] *MTConnect Standard: Part 3.0 - Observation Information Model*.  
765 Version 2.0.
- 766 [MTConnect Part 4.0] *MTConnect Standard: Part 4.0 - Asset Information Model*. Ver-  
767 sion 2.0.
- 768 [MTConnect Part 5.0] *MTConnect Standard: Part 5.0 - Interface Interaction Model*. Ver-  
769 sion 2.0.

770



### 771 3 Observation Information Model

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

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

- 782 • `sample`: Returns a designated number of time stamped *Observations* from an *agent*  
 783 associated with a piece of equipment; subject to any HTTP filtering associated with  
 784 the request. See `Agent` in *MTConnect Standard Part 1.0 - Fundamentals* for details  
 785 on the `sample` HTTP request.
- 786 • `current`: Returns a snapshot of either the most recent values or the values at a  
 787 given sequence number for all *Observations* associated with a piece of equipment  
 788 from an *agent*; subject to any HTTP filtering associated with the request. See `Agent`  
 789 in *MTConnect Standard Part 1.0 - Fundamentals* for details on the `current` HTTP  
 790 request.

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

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

804 *Components* are defined for both the `MTConnectDevices` and the `MTConnectStreams`

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

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

811 • For an `MTConnectStreams` entity: *Components* provide the structure to orga-  
812 nize the data returned from a piece of equipment and establishes the proper context  
813 for that data. The *Components* specifically defined for `MTConnectStreams` are  
814 `DeviceStream` (see *Section 3.1 - DeviceStream*) and `ComponentStream` (see  
815 *Section 3.2 - ComponentStream*).

816 `DeviceStream` and `ComponentStream` entities have a direct correlation to each of  
817 the *Component* defined in the `MTConnectDevices` entity.

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

## 822 3.1 DeviceStream

823 *organizes* data reported from a `Device`.

824 `DeviceStream` **MUST** be provided for each `Device` reporting data in an *MTConnect-*  
825 *Streams Response Document*.

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

### 829 3.1.1 Value Properties of DeviceStream

830 *Table 1* lists the Value Properties of `DeviceStream`.

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

**Table 1:** Value Properties of DeviceStream

831 Descriptions for Value Properties of DeviceStream:

- 832     • name

833         name of the Device.

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

- 837     • uuid

838         uuid of the Device.

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

## 842 3.1.2 Part Properties of DeviceStream

843 *Table 2* lists the Part Properties of DeviceStream.

Part Property name	Multiplicity
ComponentStream	1..*

**Table 2:** Part Properties of DeviceStream

844 Descriptions for Part Properties of DeviceStream:

- 845     • ComponentStream

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

848         See *Section 3.2 - ComponentStream*.

## 849 3.2 ComponentStream

850 organizes the data associated with each `Component` entity defined for a `Device` in the  
851 associated *MTConnectDevices Response Document*.

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

### 854 3.2.1 Value Properties of ComponentStream

855 *Table 3* lists the Value Properties of `ComponentStream`.

Value Property name	Value Property type	Multiplicity
<code>component</code>	string	1
<code>componentId</code>	ID	1
<code>name</code>	string	0..1
<code>nativeName</code>	string	0..1
<code>uuid</code>	ID	0..1

**Table 3:** Value Properties of `ComponentStream`

856 Descriptions for Value Properties of `ComponentStream`:

- 857 • `component`  
858 identifies the `Component` type associated with the `ComponentStream`.  
859 Examples of `component` are `Device`, `Controller`, `Linear` and `Loader`.
- 860 • `componentId`  
861 identifier of the `Component` as defined by the `id` attribute in the *MTConnectDe-*  
862 *vices Response Document*.
- 863 • `name`  
864 name of the `Component` associated with the `ComponentStream`.
- 865 • `nativeName`  
866 common name of the `Component` associated with the `ComponentStream`.
- 867 • `uuid`  
868 `uuid` of the `Component` associated with the `ComponentStream`.

## 869 3.2.2 Reference Properties of ComponentStream

870 *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

871 Descriptions for Reference Properties of ComponentStream:

- 872 • Event
  - 873     Observation that is a discrete piece of information from a piece of equipment.
  - 874     Events groups one or more Event entities. See *Section 3.5 - Event*.
  - 875 • Sample
  - 876     Observation that is continuously changing or analog data value.
  - 877     Samples groups one or more Sample entities. See *Section 3.4 - Sample*.
  - 878 • Condition
  - 879     Observation that provides the condition of a piece of equipment or a *Component*.
  - 880     *ment*.
  - 881     Conditions groups one or more Condition entities. See *Section 3.6 - Condi-*
  - 882     *tion*.
- 883             Note: In the XML representation, Conditions **MUST** appear as Con-
- 884             dition element in the *MTConnectStreams Response Document*.

## 885 3.3 Observation

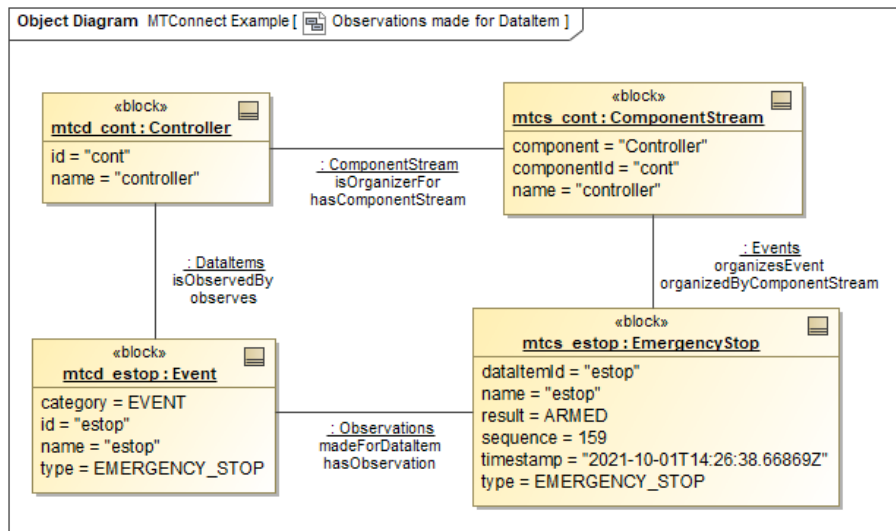
886 abstract entity that provides telemetry data for a DataItem at a point in time.

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

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

889     MTConnect entities.





**Figure 3:** Observations made for DataItem Example

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

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

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

### 909 3.3.0.2 Naming Convention for Observation types

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

914 The name of an Observation type reported in the *MTConnectStreams Response Doc-*  
 915 *ument* is extended when the representation property is used to further describe that  
 916 DataItem in the *MTConnectDevices Response Document*. See *Section 4 - Representa-*  
 917 *tions* for more details.

### 918 3.3.1 Value Properties of Observation

919 *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

920 Descriptions for Value Properties of Observation:

- 921 • compositionId
- 922 identifier of the `Composition` entity defined in the *MTConnectDevices Response Document* associated with the data reported for the Observation.
- 923
- 924 • dataItemId
- 925 unique identifier of the `DataItem` associated with this Observation.
- 926 `dataItemId` **MUST** match the `id` attribute of the `DataItem` defined in the *MT-*
- 927 *ConnectDevices Response Document*.
- 928 • name
- 929 name of the `DataItem` associated with this Observation.
- 930 `name` **MUST** match the `name` attribute of the `DataItem` defined in the *MTCon-*
- 931 *nectDevices Response Document*.
- 932 • sequence
- 933 number representing the sequential position of an occurrence of an observation in
- 934 the data buffer of an *agent*.
- 935 `sequence` **MUST** have a value represented as an unsigned 64-bit value from 1 to
- 936  $2^{64} - 1$ .



- 937 • subType
- 938 subtype of the DataItem associated with this Observation.
- 939 subType **MUST** match the subType attribute of the DataItem defined in the
- 940 *MTConnectDevices Response Document*.
- 941 The value of subType **MUST** be one of the DataItemSubTypeEnum enumer-
- 942 ation.
- 943 • timestamp
- 944 most accurate time available to a piece of equipment that represents the point in time
- 945 that the data reported was measured.
- 946 • type
- 947 type of the DataItem associated with this Observation.
- 948 type **MUST** match the type attribute of the DataItem defined in the *MTCon-*
- 949 *nectDevices Response Document*.
- 950 The value of type **MUST** be one of the DataItemTypeEnum enumeration.
- 951 • units
- 952 units of the DataItem associated with this Observation.
- 953 units **MUST** match the units attribute of the DataItem defined in the *MT-*
- 954 *ConnectDevices Response Document*.
- 955 The value of units **MUST** be one of the UnitEnum enumeration.
- 956 • isUnavailable
- 957 when true, result is indeterminate.
- 958 Note 1 to entry: In XML, when isUnavailable is true, the XML
- 959 CDATA of the Observation **MUST** be UNAVAILABLE. ““xml
- 960 <Execution dataItemId=”...” ...>UNAVAILABLE</Execution> ““
- 961 Note 2 to entry: In JavaScript Object Notation (JSON), when isUnavail-
- 962 able is true, the JSON value of the Observation **MUST** be UN-
- 963 AVAILABLE. json "Execution" : [ "dataItemId": "..."
- 964 ..., "value": "UNAVAILABLE" ]
- 965 • result
- 966 observation of the Observation entity.
- 967 The default value type for result is string.

968 Note 1 to entry: in XML the `Observation:result` is the CDATA of  
 969 the `Observation` *element*.

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

971 Note 2 to entry: in JSON the `Observation:result` is the member  
 972 value of the `Observation` object.

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

## 974 3.4 Sample

975 `Observation` that is continuously changing or analog data value.

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

979 `Sample` **MUST** always be reported in `float`.

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

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

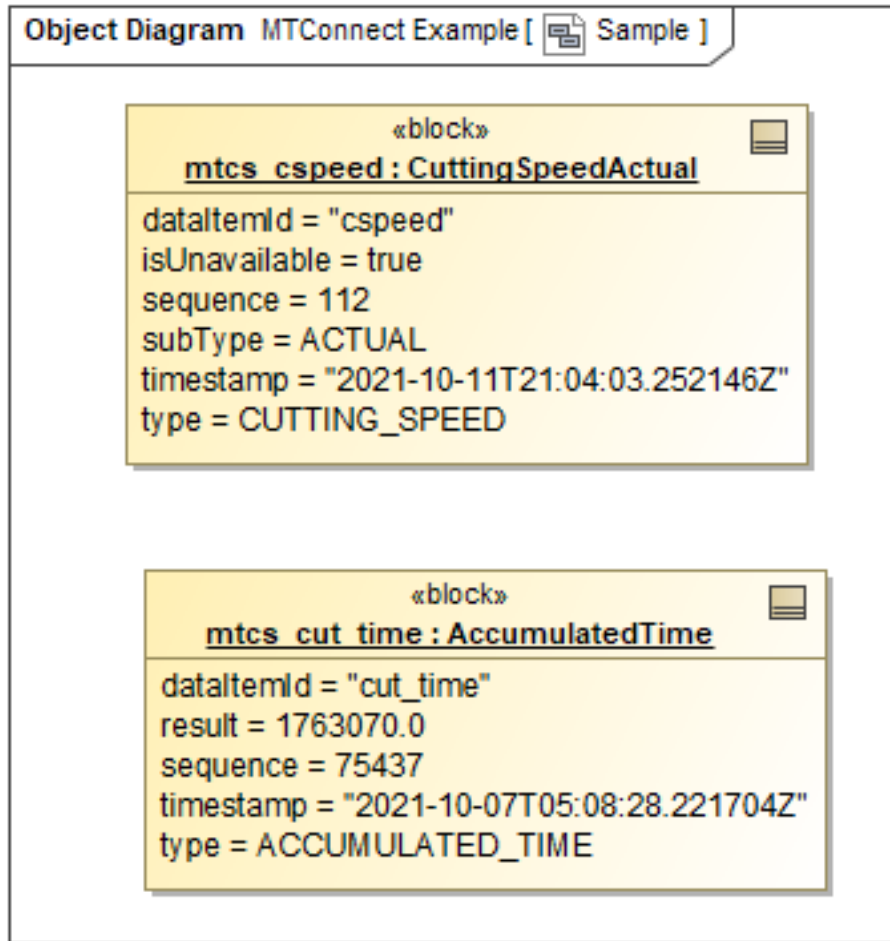
983 The following *Section 3.4.1 - Value Properties of Sample* lists the additional and/or up-  
 984 dated attributes for `Sample`.

985 The value of `Sample` **MUST** be `float`.

986 The `units` for `Sample` **MUST** always be specified.

### 987 3.4.1 Value Properties of Sample

988 *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

## 989 Descriptions for Value Properties of Sample:

990 • duration

991 time-period over which the data was collected.

992 duration **MUST** be provided when the statistic attribute of the DataItem  
993 is defined in the *MTCConnectDevices Response Document*.

994 • resetTriggered

995 identifies when a reported value has been reset and what has caused that reset to  
996 occur for those DataItem entities that may be periodically reset to an initial value.997 resetTriggered **MUST** only be provided for the specific occurrence of a DataItem  
998 reported in the *MTCConnectStreams Response Document* when the reset occurred.

999 ResetTriggeredEnum Enumeration:

1000 – ACTION\_COMPLETE

1001 result of the Observation that is measuring an action or operation was  
1002 reset upon completion of that action or operation.

1003 – ANNUAL

1004 result of the Observation was reset at the end of a 12-month period.

1005 – DAY

1006 result of the Observation was reset at the end of a 24-hour period.

1007 – MAINTENANCE

1008 result of the Observation was reset upon completion of a maintenance  
1009 event.

1010 – MANUAL

1011 result of the Observation was reset based on a physical reset action.

1012 – MONTH

1013 result of the Observation was reset at the end of a monthly period.

1014 – POWER\_ON

1015 result of the Observation was reset when power was applied to the piece  
1016 of equipment after a planned or unplanned interruption of power has occurred.

1017 – SHIFT

1018 result of the Observation was reset at the end of a work shift.

1019 – WEEK

1020 result of the Observation was reset at the end of a 7-day period.

- 1021 • `sampleRate`
- 1022 rate at which successive samples of the value are recorded.
- 1023 • `statistic`
- 1024 type of statistical calculation defined by the `statistic` attribute of the `DataItem`
- 1025 defined in the *MTCConnectDevices Response Document*.
- 1026 The value of `statistic` **MUST** be one of the `StatisticEnum` enumeration.

**1027 3.5 Event**

1028 `Observation` that is a discrete piece of information from a piece of equipment.

1029 It provides the information and data reported from a piece of equipment for those `DataItem`

1030 entities defined with a `category`, `DataItem` property of `EVENT` in the *MTCConnectDe-*

1031 *vices Response Document*.

1032 Figure 5 shows `Event` type examples. It also shows an example for when the result is

1033 not available (`dataItemId=dl_asset_rem`).

1034 Note: See *Example 5* for the XML representation of the same example.

1035 The following *Section 3.5.1 - Value Properties of Event* lists the additional and/or updated

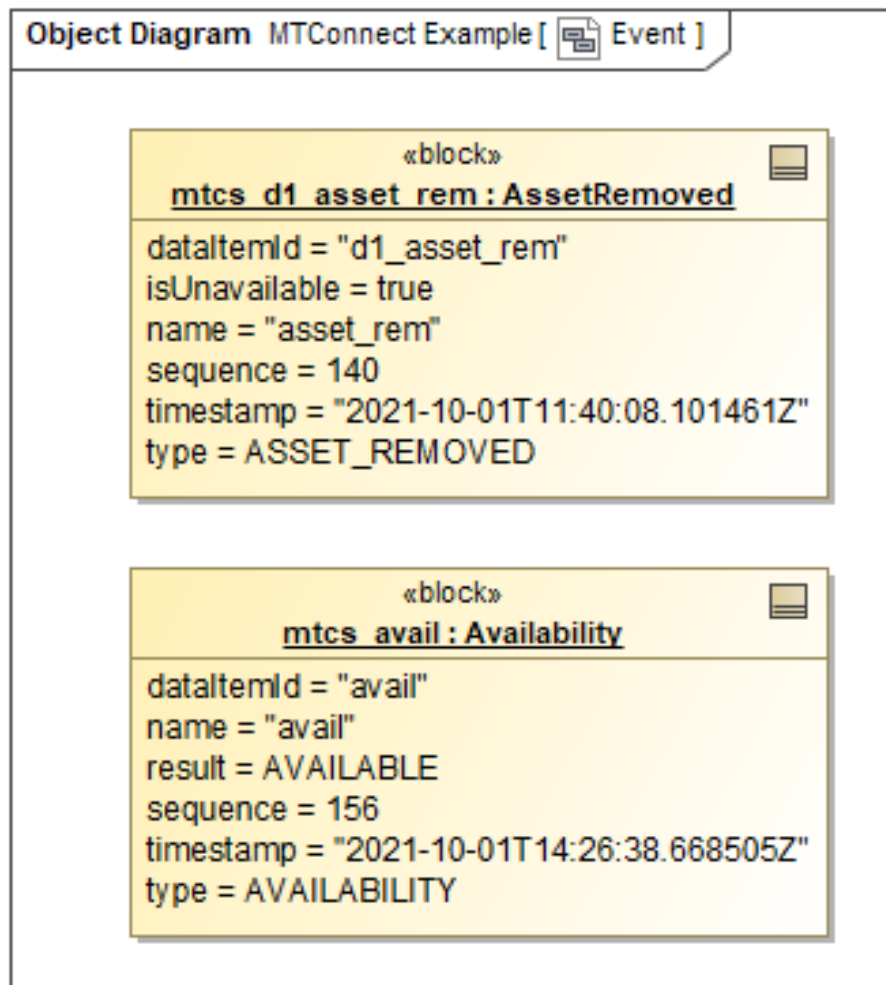
1036 attributes for `Event`.

**1037 3.5.1 Value Properties of Event**

1038 *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

1039 Descriptions for Value Properties of Event:

1040 • resetTriggered

1041 identifies when a reported value has been reset and what has caused that reset to  
1042 occur for those DataItem entities that may be periodically reset to an initial value.

1043 resetTriggered **MUST** only be provided for the specific occurrence of a DataItem  
1044 reported in the *MTConnectStreams Response Document* when the reset occurred.

1045 The value of resetTriggered **MUST** be one of the ResetTriggeredEnum  
1046 enumeration.

## 1047 3.6 Condition

1048 Observation that provides the condition of a piece of equipment or a *Component*.

1049 It provides the information and data reported from a piece of equipment for those `DataItem`  
 1050 entities defined with a `category, DataItem` property of `CONDITION` in the *MTCon-*  
 1051 *nectDevices Response Document*.

1052 Figure 6 shows `Condition` type examples for various state: Normal (`dataItemId`  
 1053 = `path_system`) and Warning (`dataItemId` = `logic_cond`). It also shows an  
 1054 example for when the state is not available (`dataItemId` = `cont_system`).

1055 Note: See *Example 6* for the XML representation of the same example.

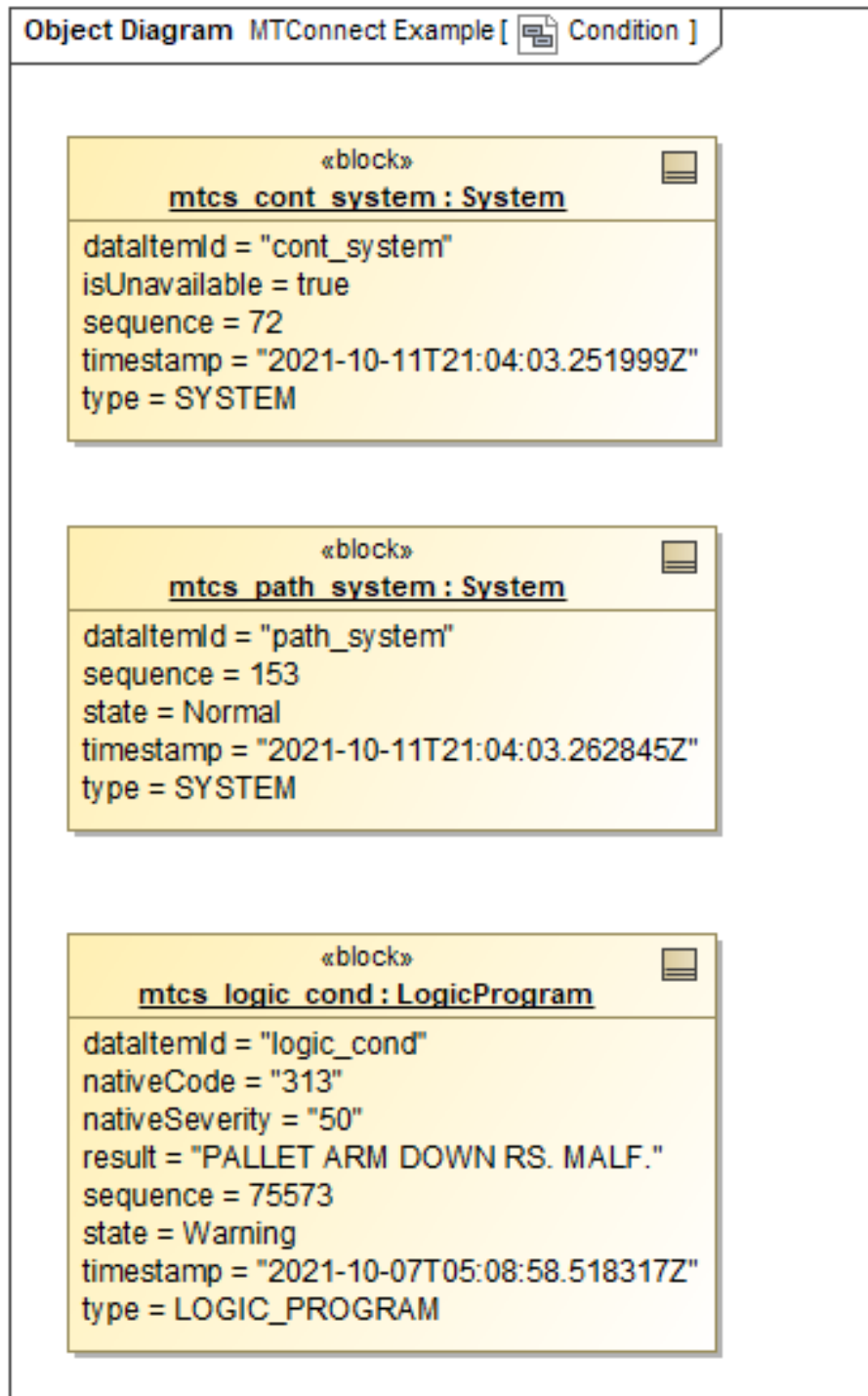
1056 The following *Section 3.6.1 - Value Properties of Condition* lists the additional and/or  
 1057 updated attributes for `Condition`.

### 1058 3.6.1 Value Properties of Condition

1059 *Table 8* lists the Value Properties of `Condition`.

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

**Table 8:** Value Properties of `Condition`



**Figure 6:** Condition Example



## 1060 Descriptions for Value Properties of Condition:

- 1061     • `nativeCode`  
 1062       native code (usually an alpha-numeric value) generated by the controller of a piece  
 1063       of equipment providing a reference identifier for a `Condition`.  
 1064       This is the same information an operator or maintenance personnel may see as a  
 1065       reference code designating a specific fault code provided by the piece of equipment.
- 1066     • `nativeSeverity`  
 1067       severity information to a client software application if the piece of equipment designates a severity level to a fault.  
 1068
- 1069     • `qualifier`  
 1070       additional information regarding a condition state associated with the measured  
 1071       value of a process variable.  
 1072       `qualifier` defines whether the condition state represented indicates a measured  
 1073       value that is above or below an expected value of a process variable.  
 1074       `QualifierEnum` Enumeration:
- 1075       – HIGH  
 1076         measured value is greater than the expected value for a process variable.
- 1077       – LOW  
 1078         measured value is less than the expected value for a process variable.
- 1079     • `statistic`  
 1080       `statistic` provides additional information describing the meaning of the `Condition`  
 1081       element.  
 1082       `statistic` **MUST** match the `statistic` attribute of the `DataItem` entity  
 1083       defined in the *MTConnectDevices Response Document*.  
 1084       The value of `statistic` **MUST** be one of the `StatisticEnum` enumeration.
- 1085     • `xs:lang`  
 1086       specifies the language of the `result` returned for the `Condition`.  
 1087       See *Ref IETF RFC 4646* (<http://www.ietf.org/rfc/rfc4646.txt>).
- 1088     • `state`  
 1089       condition state of the piece of equipment or `Component`.  
 1090       `ConditionStateEnum` Enumeration:

- 1091       – Fault
- 1092           condition state that requires intervention to continue operation to function properly.
- 1093
- 1094       – Normal
- 1095           condition state that indicates operation within specified limits.
- 1096
- 1096       – Warning
- 1097           condition state that requires concern and supervision and may become hazardous if no action is taken.
- 1098

## 1099 4 Representations

1100 This section provides semantic information for the Representation model.

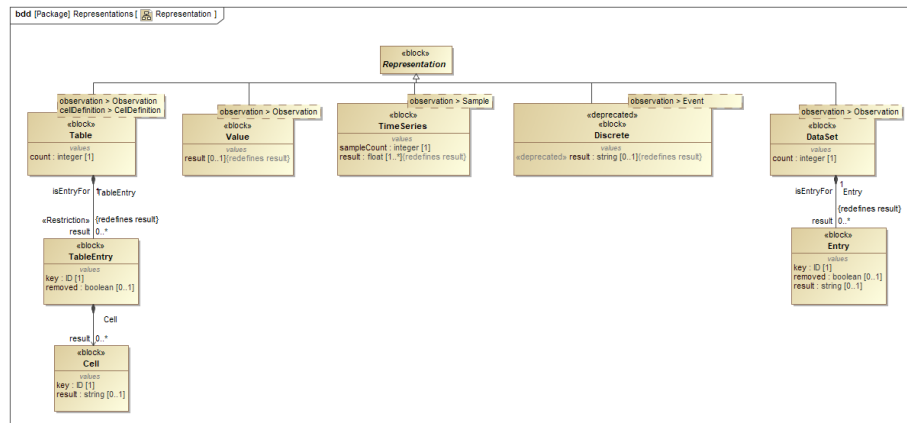


Figure 7: Representation

1101 Note: See Section B.2 - Representation Schema Diagrams for XML schema.

### 1102 4.1 Representation

1103 specifies the format and structure of the result for an Observation.

1104 The Representation type for an Observation is defined by the associated DataItem's  
1105 property representation in the *MTConnectDevices Response Document*.

1106 Value is the default Representation type for all Observation types.

1107 The name of the Observation type is modified for all Representation types other  
1108 than Value by appending the pascal case of the Representation type.

1109 Example: The name for Sample Observation type Temperaturewith Repr-  
1110 sentation type of TimeSeries becomes TemperatureTimeSeries.

### 1111 4.2 Value

1112 default Representation type for all Observation types where result of the  
1113 Observation types is an MTConnect data type. See Section 6.1 - DataTypes.

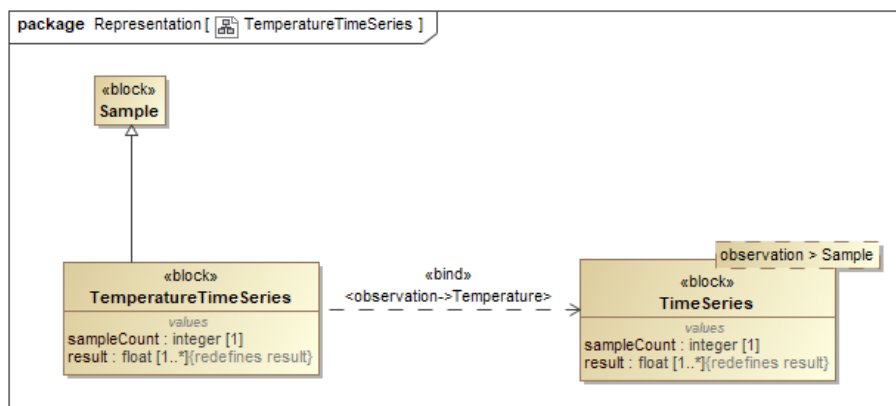
### 1114 4.3 TimeSeries

1115 Representation for an Observation composed of a series of sampled data.

1116 TimeSeries for an Observation is defined by the associated DataItem's property  
1117 representation as TIME\_SERIES.

1118 DataItem with TIME\_SERIES representation **MUST** have a category of  
1119 SAMPLE.

1120 Figure 8 shows the model for Temperature (Sample type) with a Representation  
1121 type of TimeSeries.



**Figure 8:** TemperatureTimeSeries

1122 Note: See Section B.2 - Representation Schema Diagrams for XML schema.

1123 TimeSeries **MUST** report multiple values at fixed intervals in a single Observation.  
1124 At minimum, one of DataItem or Observation **MUST** specify the sampleRate in  
1125 Hertz(values/second); fractional rates are permitted. When the Observation and the  
1126 DataItem specify the sampleRate, the Observation sampleRate supersedes  
1127 the DataItem.

1128 The Observation **MUST** set the timestamp to the time the last value was observed.  
1129 The duration **MAY** indicate the time interval from the first to the last value in the series.

1130 Section 4.3.1 - Value Properties of TimeSeries defines additional attributes for an Obser-  
1131 vation with TimeSeries Representation type.

### 1132 4.3.1 Value Properties of TimeSeries

1133 *Table 9* lists the Value Properties of TimeSeries.

Value Property name	Value Property type	Multiplicity
sampleCount	integer	1

**Table 9:** Value Properties of TimeSeries

1134 Descriptions for Value Properties of TimeSeries:

- 1135 • sampleCount
- 1136 number of values given for the Observation.

### 1137 4.4 <<deprecated>>Discrete

1138 **DEPRECATED** Representation for an Observation where each discrete occur-  
 1139 rence of the data may have the same value as the previous occurrence of the data.

1140 Discrete for an Observation is defined by the associated DataItem's property  
 1141 representation as DISCRETE.

1142 DataItem with DISCRETE representation **MUST** have a category of EVENT.

1143 *MTConnect Version 1.5* replaced representation DISCRETE with a discrete  
 1144 property for DataItem.

1145 Each occurrence of the Observation **MAY** have the same value as the previous occur-  
 1146 rence, and **MUST NOT** suppress duplicates.

1147 Examples of Discrete: A PartCount reporting the completion of each part using a 1  
 1148 to indicate completion of a single part, a Message that occurs each time a door opens.

### 1149 4.5 DataSet

1150 Representation for an Observation composed of value(s) represented as a set of  
 1151 *key-value pairs*.

1152 DataSet for an Observation is defined by the associated DataItem's property  
 1153 representation as DATA\_SET.

1154 DataItem with DATA\_SET representation **MUST** have a category of SAM-  
 1155 PLE or EVENT.

1156 Figure 9 shows the model for Variable (Event type) with a Representation type  
 1157 of DataSet.

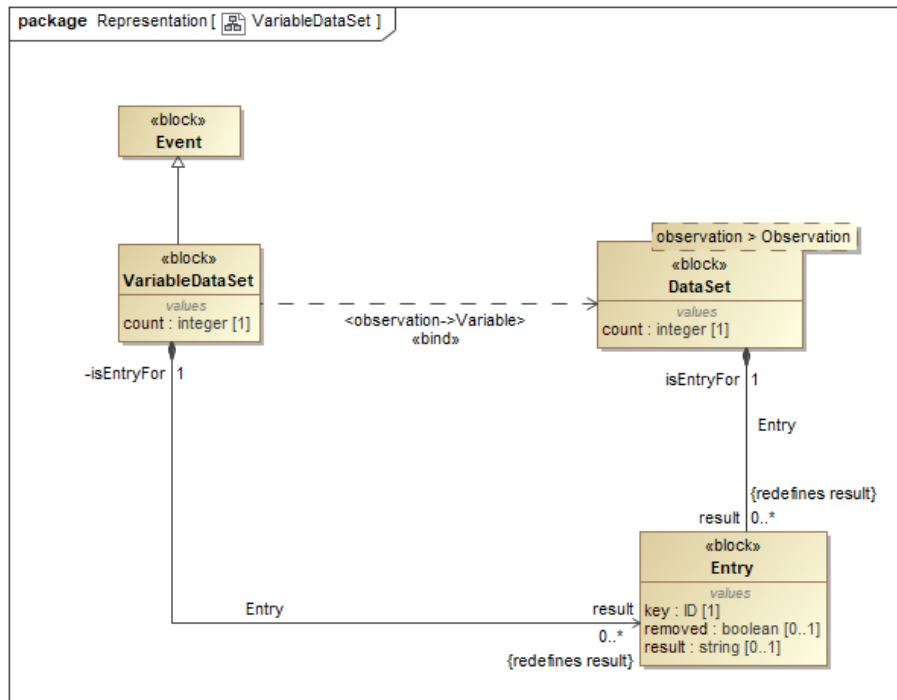


Figure 9: VariableDataSet

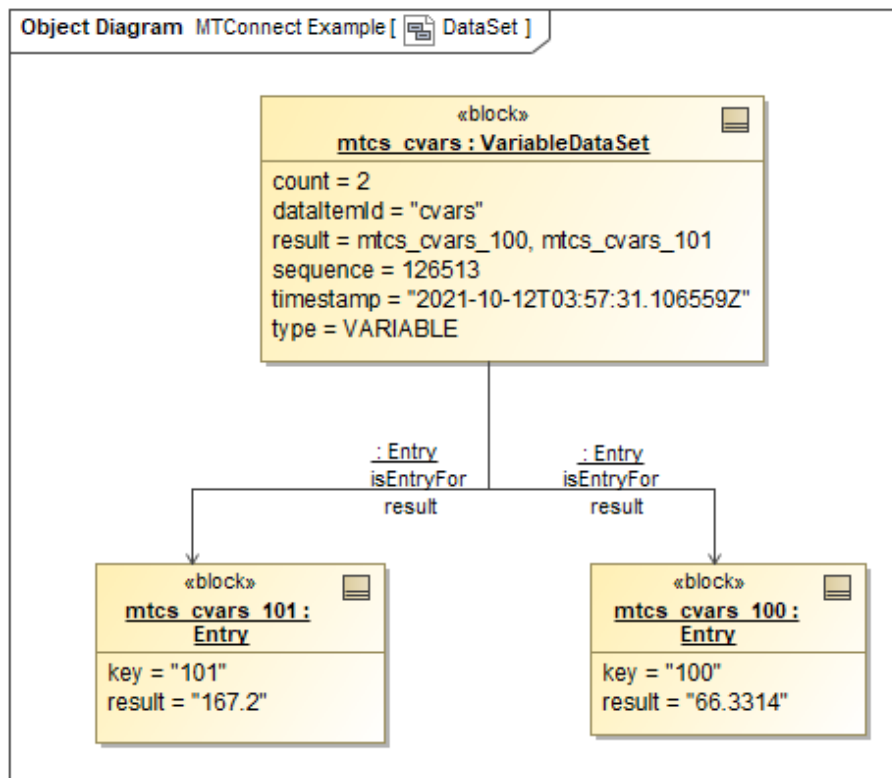
1158 DataSet reports multiple values as a set of *key-value pair* where each *key* **MUST** be  
 1159 unique. The representation of the *key-value pair* is an Entry. The value of each En-  
 1160 try **MUST** have the same constraints and format as the Observation defined for the  
 1161 VALUE representation for the DataItem type (See Value).

1162 The meaning of each Entry **MAY** be provided as the DataItem EntryDefinition.

1163 Figure 10 shows Event Observation type Variable with a Representation  
 1164 type of DataSet.

1165 Note: See Example 7 for the XML representation of the same example.

#### 1166 4.5.0.1 Management of Data Set Observations



**Figure 10:** DataSet Example

1167 An agent **MUST** maintain the current state of the DataSet as described in *MTConnect*  
 1168 *Standard Part 1.0 - Fundamentals*.

1169 One or more *key-value pairs* **MAY** be added, removed, or changed in an Observation.  
 1170 An agent **MUST** publish the changes to one or more *key-value pairs* as a single Obser-  
 1171 vation. An agent **MUST** indicate the removal of a *key-value pair* from a DataSet  
 1172 using the removed attribute equal true.

1173 When the `DataItem discrete`, `DataItem` attribute is false or is not present, an  
 1174 agent in response to a *sample request* **MUST** only publish the changed *key-value pair*  
 1175 since the previous state of the DataSet.

1176 When the `DataItem discrete`, `DataItem` attribute is true, an agent, in response  
 1177 to a *sample request*, **MUST** report all *key-value pairs* ignoring the state of the DataSet.

1178 When an agent responds to a *current request*, the *response document* **MUST** include the  
 1179 full set of *key-value pairs*. If the *current request* includes an `at` query parameter, the agent  
 1180 **MUST** provide the set of *key-value pairs* at the *sequence number*.

1181 When an Observation *reset* occurs, the DataSet **MUST** remove all *key-value pairs*  
 1182 making the set empty. The Observation **MAY** simultaneously populate the DataSet  
 1183 with new *key-value pairs*. The previous entries **MUST NOT** be included and **MUST NOT**  
 1184 have removed attribute equal true.

1185 When the Observation is UNAVAILABLE the DataSet **MUST** remove all *key-value*  
 1186 *pairs* making the set empty.

### 1187 4.5.1 Value Properties of DataSet

1188 *Table 10* lists the Value Properties of DataSet.

Value Property name	Value Property type	Multiplicity
count	integer	1

**Table 10:** Value Properties of DataSet

1189 Descriptions for Value Properties of DataSet:

- 1190 • count
- 1191     number of Entry elements for the Observation.

### 1192 4.5.2 Part Properties of DataSet

1193 *Table 11* lists the Part Properties of DataSet.

Part Property name	Multiplicity
Entry	0..*

**Table 11:** Part Properties of DataSet

1194 Descriptions for Part Properties of DataSet:

- 1195 • Entry
- 1196     *key-value pair* published as part of a DataSet.
- 1197     See *Section 4.7 - Entry*.



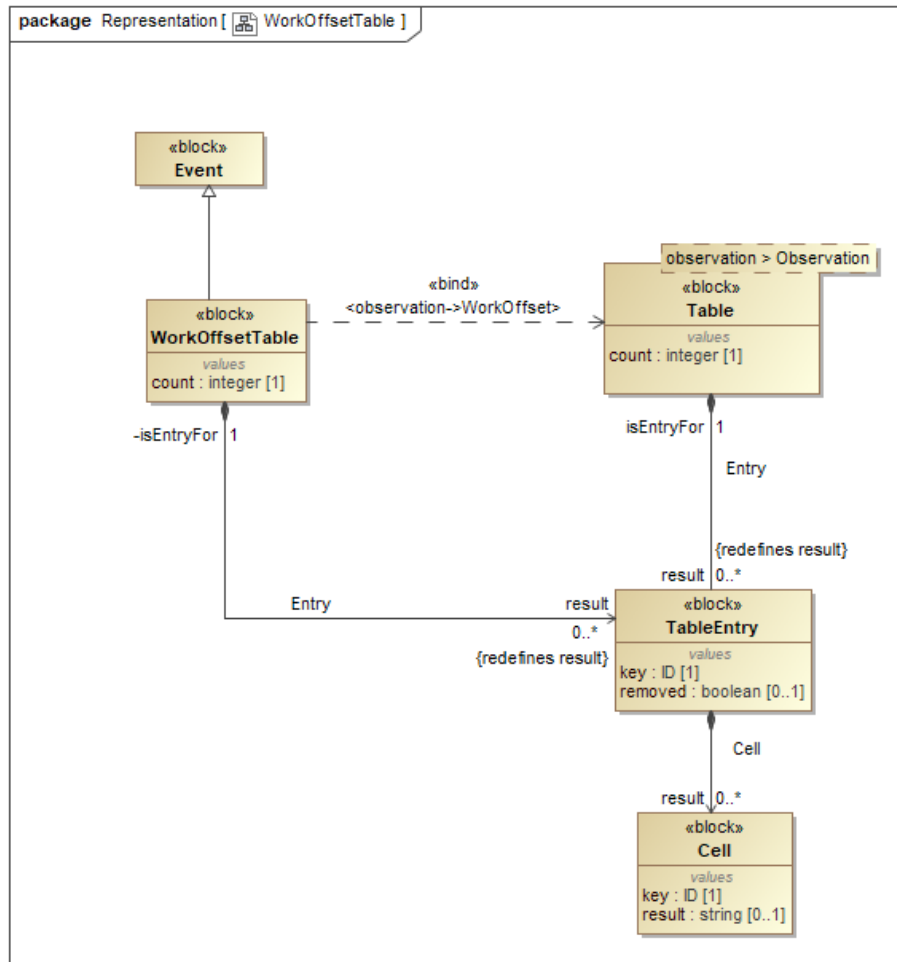
1198 **4.6 Table**

1199 Representation for an Observation composed of two-dimensional sets of *key-*  
 1200 *value pairs* where the Entry represents rows containing sets of *key-value pairs* given by  
 1201 Cell entities.

1202 Table for an Observation is defined by the associated DataItem’s property rep-  
 1203 resentation as TABLE.

1204 DataItem with TABLE representation **MUST** have a category of SAMPLE or  
 1205 EVENT.

1206 Figure 11 shows the model for WorkOffset (Event type) with a Representation  
 1207 type of Table.



**Figure 11: WorkOffsetTable**

1208 Note: See Section B.2 - Representation Schema Diagrams for XML schema.

1209 Table has the same behavior as the DataSet for change tracking, clearing, and history.  
1210 When an Entry changes, all Cell entities update at the same time; they are not tracked  
1211 separately like Entry.

1212 The meaning of each Entry and Cell MAY be provided as the DataItem Entry-  
1213 Definition and CellDefinition.

1214 The Entry key attribute MUST be the unique identity of the Entry within an Obser-  
1215 vation. The Cell key attribute MUST be the unique identity of the Cell within an  
1216 Entry.

1217 Figure 12 shows Event Observation type WorkOffset with a Representation  
1218 type of Table.

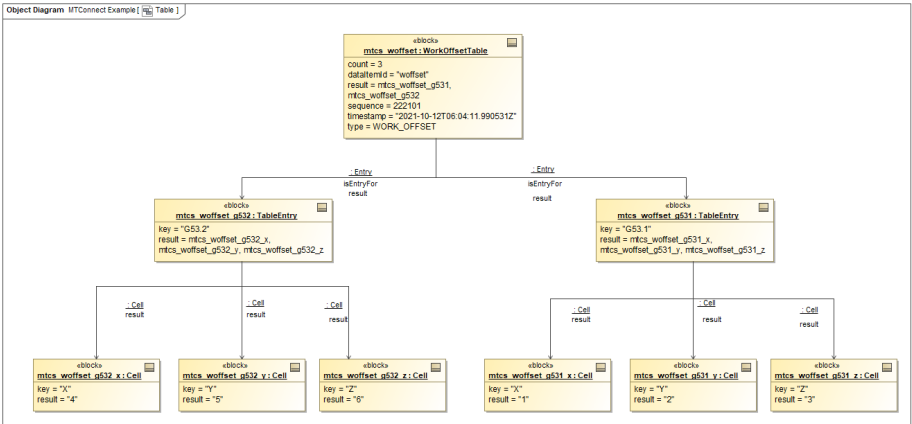


Figure 12: Table Example

1219 Note: See Example 8 for the XML representation of the same example.

1220 4.6.1 Value Properties of Table

1221 Table 12 lists the Value Properties of Table.

Value Property name	Value Property type	Multiplicity
count	integer	1

Table 12: Value Properties of Table

1222 Descriptions for Value Properties of Table:

- 1223     • count  
 1224         number of *key-value pairs* represented as `Entry` entities.

## 1225 4.6.2 Part Properties of Table

1226 *Table 13* lists the Part Properties of `Table`.

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

**Table 13:** Part Properties of `Table`

1227 Descriptions for Part Properties of `Table`:

- 1228     • `TableEntry`  
 1229         *key-value pair* published as part of a `Table`.  
 1230             Note: In the XML representation, `TableEntry` **MUST** appear as `En-`  
 1231             try.  
 1232         See *Section 4.8 - TableEntry*.

## 1233 4.7 Entry

1234 *key-value pair* published as part of a `DataSet`.

### 1235 4.7.0.1 Constraints for Entry Values

1236 The value of each `Entry` **MUST** have the same restrictions as the value of an observation  
 1237 with representation of `VALUE`.

1238 An `Entry` **MAY** be further constrained by the `DataItem` definition (see *MTConnect*  
 1239 *Standard: Part 2.0 - Device Information Model*), for example a `VariableDataSet`  
 1240 having a string value **MAY** have a floating-point `Temperature` value. A restriction  
 1241 **MUST NOT** be broadened or removed, for example, the value “READY” **MUST NOT**  
 1242 occur with a `TemperatureDataSet` constrained to floating-point numbers.

1243 The *MTConnect Standard: Part 2.0 - Device Information Model* `DataItem` Defini-  
 1244 tion **MAY** provide the type and units of an `Entry` for a key.

## 1245 4.7.1 Value Properties of Entry

1246 *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

1247 Descriptions for Value Properties of Entry:

- 1248 • key
- 1249     unique identifier for each *key-value pair*.
- 1250 • removed
- 1251     removal indicator of a *key-value pair*.

## 1252 4.8 TableEntry

1253 *key-value pair* published as part of a Table.

1254     Note: In the XML representation, TableEntry **MUST** appear as Entry.

### 1255 4.8.1 Value Properties of TableEntry

1256 *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

1257 Descriptions for Value Properties of TableEntry:

- 1258 • key
- 1259     unique identifier for each *key-value pair*.

- 1260     • removed
- 1261       removal indicator of a *key-value pair*.

## 1262 4.8.2 Part Properties of TableEntry

1263 *Table 16* lists the Part Properties of TableEntry.

Part Property name	Multiplicity
Cell	0..*

**Table 16:** Part Properties of TableEntry

1264 Descriptions for Part Properties of TableEntry:

- 1265     • Cell
- 1266       *key-value pair* published as part of a TableEntry.
- 1267       See *Section 4.9 - Cell*.

## 1268 4.9 Cell

1269 *key-value pair* published as part of a TableEntry.

### 1270 4.9.0.1 Constraints for Cell Values

1271 The value of each Cell **MUST** have the same restrictions as the value of an observation  
1272 with representation of VALUE.

1273 An Cell **MAY** be further constrained by the DataItem definition (see *MTConnect Stan-*  
1274 *dard: Part 2.0 - Device Information Model*), for example a VariableDataSet having  
1275 a string value **MAY** have a floating-point Temperature value. A restriction **MUST**  
1276 **NOT** be broadened or removed, for example, the value READY **MUST NOT** occur with a  
1277 TemperatureDataSet constrained limited to floating-point numbers.

1278 The *MTConnect Standard: Part 2.0 - Device Information Model* DataItem Defini-  
1279 tion **MAY** provide the type and units of a Cell for a key.

**1280 4.9.1 Value Properties of Cell**

1281 *Table 17* lists the Value Properties of Cell.

Value Property name	Value Property type	Multiplicity
key	ID	1

**Table 17:** Value Properties of Cell

1282 Descriptions for Value Properties of Cell:

- 1283 • key
- 1284 unique identifier for each *key-value pair*.

## 1285 **5 Observation Types**

1286 This section provides semantic information for the `Observation` types.

1287 `Observation` entities are instantiated as `Sample`, `Event` or `Condition` entities  
1288 based upon the `category` and `type` attributes defined for the corresponding `DataItem`.

1289 See Figure 2 for examples on how the `Observation` types are organized within `Com-`  
1290 `ponentStream`.

### 1291 **5.1 Condition Types**

1292 This section provides semantic information for `Condition` types.

1293 `Condition` types are reported differently from other `Observation` types. They are  
1294 reported based on the condition state for each `Condition`.

1295 The `type` and `subType` (where applicable) properties for a `Condition` **MAY** be any  
1296 of the `type` and `subType` attributes defined for `SAMPLE` category or `EVENT` cat-  
1297 egory `DataItem` listed in the *Device Information Model*.

1298 This section lists additional `Condition` types that have been defined to represent the  
1299 health and fault status of *Components*.

#### 1300 **5.1.1 Actuator**

1301 indication of a fault associated with an actuator.

#### 1302 **5.1.2 Communications**

1303 indication that the piece of equipment has experienced a communications failure.

#### 1304 **5.1.3 DataRange**

1305 indication that the value of the data associated with a measured value or a calculation is  
1306 outside of an expected range.

#### 1307 **5.1.4 LogicProgram**

1308 indication that an error occurred in the logic program or programmable logic controller  
1309 (PLC) associated with a piece of equipment.

#### 1310 **5.1.5 MotionProgram**

1311 indication that an error occurred in the motion program associated with a piece of equip-  
1312 ment.

#### 1313 **5.1.6 System**

1314 general purpose indication associated with an electronic component of a piece of equip-  
1315 ment or a controller that represents a fault that is not associated with the operator, program,  
1316 or hardware.

### 1317 **5.2 Event Types**

1318 This section provides semantic information for Event types.

#### 1319 **5.2.1 ActivationCount**

1320 accumulation of the number of times a function has attempted to, or is planned to attempt  
1321 to, activate or be performed.

1322 The default subType of ActivationCount is ALL.

1323 The value of ActivationCount **MUST** be integer.

##### 1324 **5.2.1.1 Subtypes of ActivationCount**

1325 • ABORTED

1326 accumulation of actions or activities that were attempted, but terminated before they  
1327 could be completed.



- 1328 • ALL
- 1329 accumulation of all actions, items, or activities being counted independent of the
- 1330 outcome.
- 1331 • BAD
- 1332 accumulation of actions, items, or activities being counted that do not conform to
- 1333 specification or expectation.
- 1334 • COMPLETE
- 1335 accumulation of actions, items, or activities that have been completed, independent
- 1336 of the outcome.
- 1337 • FAILED
- 1338 accumulation of actions or activities that were attempted, but failed to complete or
- 1339 resulted in an unexpected or unacceptable outcome.
- 1340 • GOOD
- 1341 accumulation of actions, items, or activities being counted that conform to specifi-
- 1342 cation or expectation.
- 1343 • REMAINING
- 1344 accumulation of actions, items, or activities yet to be counted.
- 1345 • TARGET
- 1346 goal of the operation or process.

### 1347 **5.2.2 ActiveAxes**

- 1348 set of axes currently associated with a `Path` or `Controller`.
- 1349 The value of `ActiveAxes` **MUST** be a list of string of size `0..*`.

### 1350 **5.2.3 ActuatorState**

- 1351 operational state of an apparatus for moving or controlling a mechanism or system.
- 1352 `ActuatorStateEnum` Enumeration:

- 1353     • ACTIVE
- 1354     Actuator is operating.
- 1355     • INACTIVE
- 1356     Actuator is not operating.

**1357 5.2.4 AdapterSoftwareVersion**

1358 originator’s software version of the *adapter*.

**1359 5.2.5 AdapterURI**

1360 Uniform Resource Identifier (URI) of the *adapter*.

**1361 5.2.6 <<deprecated>>Alarm**

1362 **DEPRECATED:** Replaced with CONDITION category data items in Version 1.1.0.

**1363 5.2.6.1 Value Properties of Alarm**

1364 *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

1365 Descriptions for Value Properties of Alarm:

- 1366     • <<deprecated>> code
- 1367     type of alarm.

- 1368 • <<deprecated>> severity
- 1369 severity of the alarm.
- 1370 • <<deprecated>> nativeCode
- 1371 native code for the piece of equipment.
- 1372 • <<deprecated>> state
- 1373 state of the alarm.
- 1374 • <<deprecated>> lang
- 1375 specifies the language of the alarm text.
- 1376 See *Ref IETF RFC 4646* (<http://www.ietf.org/rfc/rfc4646.txt>).

### 1377 **5.2.7 AlarmLimit**

1378 set of limits used to trigger warning or alarm indicators.

1379 The Entry key **MUST** be one or more from the AlarmLimitResult keys.

1380 AlarmLimitResult keys:

- 1381 • UpperLimit
- 1382 upper conformance boundary for a variable.
- 1383 Note: immediate concern or action may be required.
- 1384 The value of UpperLimit **MUST** be float.
- 1385 • UpperWarning
- 1386 upper boundary indicating increased concern and supervision may be required.
- 1387 The value of UpperWarning **MUST** be float.
- 1388 • LowerWarning
- 1389 lower boundary indicating increased concern and supervision may be required.
- 1390 The value of LowerWarning **MUST** be float.
- 1391 • LowerLimit
- 1392 lower conformance boundary for a variable.
- 1393 Note: immediate concern or action may be required.
- 1394 The value of LowerLimit **MUST** be float.

## 1395 **5.2.8 Application**

1396 application on a Component.

1397 A subType **MUST** always be specified.

### 1398 **5.2.8.1 Subtypes of Application**

1399 • INSTALL\_DATE

1400 date the hardware or software was installed.

1401 The value of Application **MUST** be datetime. See *Section 6.1.5 - datetime*.

1402 datetime Enumeration:

1403 • LICENSE

1404 license code to validate or activate the hardware or software.

1405 • MANUFACTURER

1406 corporate identity for the maker of the hardware or software.

1407 • RELEASE\_DATE

1408 date the hardware or software was released for general use.

1409 The value of Application **MUST** be datetime. See *Section 6.1.5 - datetime*.

1410 datetime Enumeration:

1411 • VERSION

1412 version of the hardware or software.

## 1413 **5.2.9 AssetChanged**

1414 assetId of the *Asset* that has been added or changed.

### 1415 **5.2.9.1 Value Properties of AssetChanged**

1416 *Table 19* lists the Value Properties of AssetChanged.

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

**Table 19:** Value Properties of AssetChanged

1417 Descriptions for Value Properties of AssetChanged:

- 1418     • assetType  
1419         type of Asset changed. See *MTCConnect Standard: Part 4.0 - Asset Information*  
1420         *Model* for details on the Asset model.
- 1421     • hash  
1422         condensed message digest from a secure one-way hash function. *Ref FIPS PUB*  
1423         *180-4*

## 1424 5.2.10 AssetCount

- 1425 *data set* of the number of *Assets* of a given type for a *Device*.
- 1426 The value of AssetCount **MUST** be integer.

## 1427 5.2.11 AssetRemoved

- 1428 *assetId* of the *Asset* that has been removed.

### 1429 5.2.11.1 Value Properties of AssetRemoved

1430 *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

1431 Descriptions for Value Properties of AssetRemoved:

- 1432 • `assetType`
- 1433 type of `Asset` removed. See *MTConnect Standard: Part 4.0 - Asset Information*
- 1434 *Model* for details on the `Asset` model.
- 1435 • `hash`
- 1436 condensed message digest from a secure one-way hash function. *Ref FIPS PUB*
- 1437 *180-4*

### 1438 5.2.12 Availability

1439 *agent's* ability to communicate with the data source.

1440 `AvailabilityEnum` Enumeration:

- 1441 • `AVAILABLE`
- 1442 data source is active and capable of providing data.
- 1443 • `UNAVAILABLE`
- 1444 data source is either inactive or not capable of providing data.

### 1445 5.2.13 AxisCoupling

1446 describes the way the axes will be associated to each other.

1447 This is used in conjunction with `COUPLED_AXES` to indicate the way they are interacting.

1448 `AxisCouplingEnum` Enumeration:

- 1449 • `MASTER`
- 1450 axis is the master of the `CoupledAxes`.
- 1451 • `SLAVE`
- 1452 axis is a slave to the `CoupledAxes`.
- 1453 • `SYNCHRONOUS`
- 1454 axes are not physically connected to each other but are operating together in lock-
- 1455 step.

- 1456 • TANDEM
- 1457 axes are physically connected to each other and operate as a single unit.

## 1458 5.2.14 AxisFeedrateOverride

1459 value of a signal or calculation issued to adjust the feedrate of an individual linear type  
1460 axis.

1461 The value of `AxisFeedrateOverride` **MUST** be float.

### 1462 5.2.14.1 Subtypes of AxisFeedrateOverride

- 1463 • JOG
- 1464 relating to momentary activation of a function or a movement.
- 1465 **DEPRECATION WARNING:** May be deprecated in the future.
- 1466 When the JOG subtype of `AxisFeedrateOverride` is applied, the resulting
- 1467 commanded feedrate for the axis is limited to the value of the original JOG subtype
- 1468 of the `AxisFeedrate` multiplied by the value of the JOG subtype of `AxisFee-`
- 1469 `drateOverride`.
- 1470 • PROGRAMMED
- 1471 directive value without offsets and adjustments.
- 1472 • RAPID
- 1473 performing an operation faster or in less time than nominal rate.

## 1474 5.2.15 AxisInterlock

1475 state of the axis lockout function when power has been removed and the axis is allowed to  
1476 move freely.

1477 `AxisInterlockEnum` Enumeration:

- 1478 • ACTIVE
- 1479 axis lockout function is activated, power has been removed from the axis, and the
- 1480 axis is allowed to move freely.

- 1481     • INACTIVE
- 1482     axis lockout function has not been activated, the axis may be powered, and the axis
- 1483     is capable of being controlled by another component.

## 1484 5.2.16 AxisState

1485 state of a Linear or Rotary component representing an axis.

1486 AxisStateEnum Enumeration:

- 1487     • HOME
- 1488     axis is in its home position.
- 1489     • PARKED
- 1490     axis has been moved to a fixed position and is being maintained in that position
- 1491     either electrically or mechanically.
- 1492     Action is required to release the axis from this position.
- 1493     • STOPPED
- 1494     axis is stopped.
- 1495     • TRAVEL
- 1496     axis is in motion.

## 1497 5.2.17 BatteryState

1498 present status of the battery.

1499 BatteryStateEnum Enumeration:

- 1500     • CHARGED
- 1501     Component is at it's maximum rated charge level.
- 1502     • CHARGING
- 1503     Component's charge is increasing.



- 1504 • DISCHARGED
- 1505 Component is at it's minimum charge level.
- 1506 • DISCHARGING
- 1507 Component's charge is decreasing.

### 1508 **5.2.18 Block**

1509 line of code or command being executed by a Controller entity.

### 1510 **5.2.19 BlockCount**

1511 total count of the number of blocks of program code that have been executed since execu-  
1512 tion started.

1513 The value of BlockCount **MUST** be integer.

### 1514 **5.2.20 CharacteristicPersistentId**

1515 Universally Unique Identifier (UUID) of the *characteristic*.

1516 The value of CharacteristicPersistentId **MUST** be ID. See *Section 6.1.2 - ID*.

### 1517 **5.2.21 CharacteristicStatus**

1518 pass/fail result of the measurement.

1519 CharacteristicStatusEnum Enumeration:

- 1520 • BASIC\_OR\_THEORETIC\_EXACT\_DIMENSION
- 1521 nominal provided without tolerance limits. *Ref QIF 3:2018 5.10.2.6*
- 1522 • FAIL
- 1523 measurement is not within acceptable tolerances.

- 1524 • INDETERMINATE
- 1525 measurement cannot be determined.
- 1526 • NOT\_ANALYZED
- 1527 measurement cannot be evaluated.
- 1528 • PASS
- 1529 measurement is within acceptable tolerances.
- 1530 • REWORK
- 1531 failed, but acceptable constraints achievable by utilizing additional manufacturing
- 1532 processes.
- 1533 • SYSTEM\_ERROR
- 1534 measurement is indeterminate due to an equipment failure.
- 1535 • UNDEFINED
- 1536 status of measurement cannot be determined.

## 1537 **5.2.22 ChuckInterlock**

1538 state of an interlock function or control logic state intended to prevent the associated  
1539 Chuck component from being operated.

1540 ChuckInterlockEnum Enumeration:

- 1541 • ACTIVE
- 1542 chuck cannot be unclamped.
- 1543 • INACTIVE
- 1544 chuck can be unclamped.

### 1545 **5.2.22.1 Subtypes of ChuckInterlock**

- 1546 • MANUAL\_UNCLAMP
- 1547 indication of the state of an operator controlled interlock that can inhibit the ability
- 1548 to initiate an unclamp action of an electronically controlled chuck.

1549       When `ChuckInterlockManualUnclamp` is `ACTIVE`, it is expected that a chuck  
1550 cannot be unclamped until `ChuckInterlockManualUnclamp` is set to `INAC-`  
1551 `TIVE`.

### 1552 **5.2.23 ChuckState**

1553 operating state of a mechanism that holds a part or stock material during a manufacturing  
1554 process.

1555 It may also represent a mechanism that holds any other mechanism in place within a piece  
1556 of equipment.

1557 `ChuckStateEnum` Enumeration:

1558       • `CLOSED`

1559       Chuck is closed to the point of a positive confirmation.

1560       • `OPEN`

1561       Chuck is open to the point of a positive confirmation.

1562       • `UNLATCHED`

1563       Chuck is not closed to the point of a positive confirmation and not open to the point  
1564 of a positive confirmation.

1565       It is in an intermediate position.

### 1566 **5.2.24 ClockTime**

1567 time provided by a timing device at a specific point in time.

1568 The value of `ClockTime` **MUST** be `datetime`. See *Section 6.1.5 - datetime*.

### 1569 **5.2.25 <<deprecated>>Code**

1570 programmatic code being executed.

1571 **DEPRECATED** in *Version 1.1*.

## 1572 5.2.26 ComponentData

1573 tabular Event that represents a Component where the EntryDefinition identi-  
 1574 fies the Component and the CellDefinitions define the Component's observed  
 1575 DataItems.

1576 If the Component multiplicity can be determined, the device model **MUST** use a fixed  
 1577 set of Components.

1578 ComponentData **MUST** provide a DataItem Definition.

## 1579 5.2.27 CompositionState

1580 operating state of a mechanism represented by a Composition entity.

1581 A subType **MUST** always be specified.

### 1582 5.2.27.1 Subtypes of CompositionState

1583 • ACTION

1584 indication of the operating state of a mechanism.

1585 CompositionStateActionEnum Enumeration:

1586 – ACTIVE

1587 Composition is operating.

1588 – INACTIVE

1589 Composition is not operating.

1590 • LATERAL

1591 indication of the position of a mechanism that may move in a lateral direction.

1592 CompositionStateLateralEnum Enumeration:

1593 – LEFT

1594 position of the Composition is oriented to the left to the point of a positive  
 1595 confirmation.

1596 – RIGHT

1597 position of the Composition is oriented to the right to the point of a positive  
 1598 confirmation.

- 1599           – TRANSITIONING  
 1600           position of the `Composition` is not oriented to the right to the point of a  
 1601           positive confirmation and is not oriented to the left to the point of a positive  
 1602           confirmation.  
 1603           It is in an intermediate position.
- 1604       • MOTION  
 1605           indication of the open or closed state of a mechanism.  
 1606           `CompositionStateMotionEnum` Enumeration:
- 1607           – CLOSED  
 1608           position of the `Composition` is closed to the point of a positive confirmation.  
 1609           – OPEN  
 1610           position of the `Composition` is open to the point of a positive confirmation.  
 1611           – UNLATCHED  
 1612           position of the `Composition` is not open to the point of a positive confirma-  
 1613           tion and is not closed to the point of a positive confirmation.  
 1614           It is in an intermediate position.
- 1615       • SWITCHED  
 1616           indication of the activation state of a mechanism.  
 1617           `CompositionStateSwitchedEnum` Enumeration:
- 1618           – OFF  
 1619           activation state of the `Composition` is in an OFF condition, it is not operat-  
 1620           ing, or it is not powered.  
 1621           – ON  
 1622           activation state of the `Composition` is in an ON condition, it is operating, or  
 1623           it is powered.
- 1624       • VERTICAL  
 1625           indication of the position of a mechanism that may move in a vertical direction.  
 1626           `CompositionStateVerticalEnum` Enumeration:
- 1627           – DOWN  
 1628           position of the `Composition` element is oriented in a downward direction to  
 1629           the point of a positive confirmation.

- 1630           – TRANSITIONING  
1631           position of the `Composition` element is not oriented in an upward direc-  
1632           tion to the point of a positive confirmation and is not oriented in a downward  
1633           direction to the point of a positive confirmation.  
1634           It is in an intermediate position.  
1635           – UP  
1636           position of the `Composition` element is oriented in an upward direction to  
1637           the point of a positive confirmation.

### 1638 5.2.28 `ConnectionStatus`

1639 status of the connection between an *adapter* and an *agent*.

1640 `ConnectionStatusEnum` Enumeration:

- 1641       • CLOSED  
1642       no connection at all.
- 1643       • ESTABLISHED  
1644       open connection.  
1645       The normal state for the data transfer phase of the connection.
- 1646       • LISTEN  
1647       *agent* is waiting for a connection request from an *adapter*.

### 1648 5.2.29 `ControlLimit`

1649 set of limits used to indicate whether a process variable is stable and in control.

1650 The `Entry` key **MUST** be one or more from the `ControlLimitResult` keys.

1651 `ControlLimitResult` keys:

- 1652       • `UpperLimit`  
1653       upper conformance boundary for a variable.

1654           Note: immediate concern or action may be required.

- 1655       The value of `UpperLimit` **MUST** be float.
- 1656       • `UpperWarning`
- 1657       upper boundary indicating increased concern and supervision may be required.
- 1658       The value of `UpperWarning` **MUST** be float.
- 1659       • `Nominal`
- 1660       ideal or desired value for a variable.
- 1661       The value of `Nominal` **MUST** be float.
- 1662       • `LowerWarning`
- 1663       lower boundary indicating increased concern and supervision may be required.
- 1664       The value of `LowerWarning` **MUST** be float.
- 1665       • `LowerLimit`
- 1666       lower conformance boundary for a variable.
- 1667               Note: immediate concern or action may be required.
- 1668       The value of `LowerLimit` **MUST** be float.

### 1669 **5.2.30 ControllerMode**

- 1670   current mode of the `Controller` component.
- 1671   `ControllerModeEnum` Enumeration:
- 1672       • `AUTOMATIC`
- 1673       `Controller` is configured to automatically execute a program.
- 1674       • `EDIT`
- 1675       `Controller` is currently functioning as a programming device and is not capable
- 1676       of executing an active program.
- 1677       • `<<deprecated>> FEED_HOLD`
- 1678       axes of the device are commanded to stop, but the spindle continues to function.

- 1679 • MANUAL
- 1680 Controller is not executing an active program.
- 1681 It is capable of receiving instructions from an external source – typically an operator.
- 1682 The Controller executes operations based on the instructions received from the
- 1683 external source.
- 1684 • MANUAL\_DATA\_INPUT
- 1685 operator can enter a series of operations for the Controller to perform.
- 1686 The Controller will execute this specific series of operations and then stop.
- 1687 • SEMI\_AUTOMATIC
- 1688 Controller is operating in a mode that restricts the active program from process-
- 1689 ing its next process step without operator intervention.

### 1690 5.2.31 ControllerModeOverride

1691 setting or operator selection that changes the behavior of a piece of equipment.

1692 ControllerModeOverrideEnum Enumeration:

- 1693 • OFF
- 1694 ControllerModeOverride is in the OFF state and the mode override is inac-
- 1695 tive.
- 1696 • ON
- 1697 ControllerModeOverride is in the ON state and the mode override is active.

1698 A subType **MUST** always be specified.

#### 1699 5.2.31.1 Subtypes of ControllerModeOverride

- 1700 • DRY\_RUN
- 1701 setting or operator selection used to execute a test mode to confirm the execution of
- 1702 machine functions.
- 1703 When DRY\_RUN is ON, the equipment performs all of its normal functions, except
- 1704 no part or product is produced. If the equipment has a spindle, spindle operation is
- 1705 suspended.



- 1706 • MACHINE\_AXIS\_LOCK  
 1707 setting or operator selection that changes the behavior of the controller on a piece of  
 1708 equipment.  
 1709 When MACHINE\_AXIS\_LOCK is ON, program execution continues normally, but  
 1710 no equipment motion occurs.
- 1711 • OPTIONAL\_STOP  
 1712 setting or operator selection that changes the behavior of the controller on a piece of  
 1713 equipment.  
 1714 The program execution is stopped after a specific program block is executed when  
 1715 OPTIONAL\_STOP is ON.  
 1716 In the case of a G-Code program, a program block containing a M01 code designates  
 1717 the command for an OPTIONAL\_STOP.  
 1718 Execution **MUST** change to OPTIONAL\_STOP after a program block speci-  
 1719 fying an optional stop is executed and the ControllerModeOverride OP-  
 1720 TIONAL\_STOP selection is ON.
- 1721 • SINGLE\_BLOCK  
 1722 setting or operator selection that changes the behavior of the controller on a piece of  
 1723 equipment.  
 1724 Program execution is paused after each block of code is executed when SINGLE\_BLOCK  
 1725 is ON.  
 1726 When SINGLE\_BLOCK is ON, Execution **MUST** change to INTERRUPTED  
 1727 after completion of each block of code.
- 1728 • TOOL\_CHANGE\_STOP  
 1729 setting or operator selection that changes the behavior of the controller on a piece of  
 1730 equipment.  
 1731 Program execution is paused when a command is executed requesting a cutting tool  
 1732 to be changed.  
 1733 Execution **MUST** change to INTERRUPTED after completion of the command  
 1734 requesting a cutting tool to be changed and TOOL\_CHANGE\_STOP is ON.

### 1735 5.2.32 CoupledAxes

1736 set of associated axes.

1737 The value of CoupledAxes **MUST** be a list of string of size 0..\*.

**1738 5.2.33 CycleCount**

1739 accumulation of the number of times a cyclic function has attempted to, or is planned to  
 1740 attempt to execute.

1741 The default subType of CycleCount is ALL.

1742 The value of CycleCount **MUST** be integer.

**1743 5.2.33.1 Subtypes of CycleCount**

1744 • ABORTED  
 1745 accumulation of actions or activities that were attempted, but terminated before they  
 1746 could be completed.

1747 • ALL  
 1748 accumulation of all actions, items, or activities being counted independent of the  
 1749 outcome.

1750 • BAD  
 1751 accumulation of actions, items, or activities being counted that do not conform to  
 1752 specification or expectation.

1753 • COMPLETE  
 1754 accumulation of actions, items, or activities that have been completed, independent  
 1755 of the outcome.

1756 • FAILED  
 1757 accumulation of actions or activities that were attempted, but failed to complete or  
 1758 resulted in an unexpected or unacceptable outcome.

1759 • GOOD  
 1760 accumulation of actions, items, or activities being counted that conform to specifi-  
 1761 cation or expectation.

1762 • REMAINING  
 1763 accumulation of actions, items, or activities yet to be counted.

1764 • TARGET  
 1765 goal of the operation or process.

## 1766 **5.2.34 DateCode**

1767 time and date code associated with a material or other physical item.

1768 The value of `DateCode` **MUST** be `datetime`. See *Section 6.1.5 - datetime*.

### 1769 **5.2.34.1 Subtypes of DateCode**

1770 • EXPIRATION

1771 time and date code relating to the expiration or end of useful life for a material or  
1772 other physical item.

1773 • FIRST\_USE

1774 time and date code relating the first use of a material or other physical item.

1775 • MANUFACTURE

1776 time and date code relating to the production of a material or other physical item.

## 1777 **5.2.35 DeactivationCount**

1778 accumulation of the number of times a function has attempted to, or is planned to attempt  
1779 to, deactivate or cease.

1780 The default `subType` of `DeactivationCount` is `ALL`.

1781 The value of `DeactivationCount` **MUST** be integer.

### 1782 **5.2.35.1 Subtypes of DeactivationCount**

1783 • ABORTED

1784 accumulation of actions or activities that were attempted, but terminated before they  
1785 could be completed.

1786 • ALL

1787 accumulation of all actions, items, or activities being counted independent of the  
1788 outcome.

- 1789     • BAD
- 1790         accumulation of actions, items, or activities being counted that do not conform to
- 1791         specification or expectation.
- 1792     • COMPLETE
- 1793         accumulation of actions, items, or activities that have been completed, independent
- 1794         of the outcome.
- 1795     • FAILED
- 1796         accumulation of actions or activities that were attempted, but failed to complete or
- 1797         resulted in an unexpected or unacceptable outcome.
- 1798     • GOOD
- 1799         accumulation of actions, items, or activities being counted that conform to specifi-
- 1800         cation or expectation.
- 1801     • REMAINING
- 1802         accumulation of actions, items, or activities yet to be counted.
- 1803     • TARGET
- 1804         goal of the operation or process.

**1805 5.2.36 DeviceAdded**

1806 UUID of new device added to an *MTConnect Agent*.

**1807 5.2.36.1 Value Properties of DeviceAdded**

1808 *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*

1809 Descriptions for Value Properties of *DeviceAdded*:

- 1810     • hash
- 1811         condensed message digest from a secure one-way hash function. *Ref FIPS PUB*
- 1812         *180-4*

**1813 5.2.37 DeviceChanged**

1814 UUID of the device whose *metadata* has changed.

**1815 5.2.37.1 Value Properties of DeviceChanged**

1816 *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`

1817 Descriptions for Value Properties of `DeviceChanged`:

- 1818 • hash
- 1819 condensed message digest from a secure one-way hash function. *Ref FIPS PUB*
- 1820 *180-4*

**1821 5.2.38 DeviceRemoved**

1822 UUID of a device removed from an *MTConnect Agent*.

**1823 5.2.38.1 Value Properties of DeviceRemoved**

1824 *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`

1825 Descriptions for Value Properties of `DeviceRemoved`:

- 1826 • hash
- 1827 condensed message digest from a secure one-way hash function. *Ref FIPS PUB*
- 1828 *180-4*

**1829 5.2.39 DeviceUuid**

1830 identifier of another piece of equipment that is temporarily associated with a component  
 1831 of this piece of equipment to perform a particular function.

**1832 5.2.40 Direction**

1833 direction of motion.

1834 <<deprecated>> DirectionEnum Enumeration:

- 1835 • <<deprecated>> CLOCKWISE
- 1836     clockwise rotation using the right-hand rule.
- 1837 • <<deprecated>> COUNTER\_CLOCKWISE
- 1838     counter-clockwise rotation using the right-hand rule.
- 1839 • <<deprecated>> NEGATIVE
- 1840 • <<deprecated>> POSITIVE

1841 A subType **MUST** always be specified.

**1842 5.2.40.1 Subtypes of Direction**

- 1843 • LINEAR
- 1844     direction of motion of a linear motion.
- 1845     DirectionLinearEnum Enumeration:
- 1846         – NEGATIVE
- 1847             linear position is decreasing.
- 1848         – NONE
- 1849             no direction.
- 1850         – POSITIVE
- 1851             linear position is increasing.

- 1852     • ROTARY
- 1853         rotational direction of a rotary motion using the right hand rule convention.
- 1854         DirectionRotaryEnum Enumeration:
- 1855             – CLOCKWISE
- 1856                 clockwise rotation using the right-hand rule.
- 1857             – COUNTER\_CLOCKWISE
- 1858                 counter-clockwise rotation using the right-hand rule.
- 1859             – NONE
- 1860                 no direction.

#### 1861 **5.2.41 DoorState**

1862 operational state of a Door component or composition element.

1863 DoorStateEnum Enumeration:

- 1864     • CLOSED
- 1865         Door is closed to the point of a positive confirmation.
- 1866     • OPEN
- 1867         Door is open to the point of a positive confirmation.
- 1868     • UNLATCHED
- 1869         Door is not closed to the point of a positive confirmation and not open to the point
- 1870         of a positive confirmation.
- 1871         It is in an intermediate position.

#### 1872 **5.2.42 EmergencyStop**

1873 state of the emergency stop signal for a piece of equipment, controller path, or any other

1874 component or subsystem of a piece of equipment.

1875 EmergencyStopEnum Enumeration:

- 1876 • ARMED
- 1877 emergency stop circuit is complete and the piece of equipment, component, or com-
- 1878 position is allowed to operate.
- 1879 • TRIGGERED
- 1880 operation of the piece of equipment, component, or composition is inhibited.

### 1881 **5.2.43 EndOfBar**

1882 indication of whether the end of a piece of bar stock being feed by a bar feeder has been  
1883 reached.

1884 EndOfBarEnum Enumeration:

- 1885 • NO
- 1886 EndOfBar has not been reached.
- 1887 • YES
- 1888 EndOfBar has been reached.

1889 The default subType of EndOfBar is PRIMARY.

#### 1890 **5.2.43.1 Subtypes of EndOfBar**

- 1891 • AUXILIARY
- 1892 when multiple locations on a piece of bar stock are referenced as the indication for
- 1893 the EndOfBar, the additional location(s) **MUST** be designated as AUXILIARY
- 1894 indication(s) for the EndOfBar.
- 1895 • PRIMARY
- 1896 specific applications **MAY** reference one or more locations on a piece of bar stock
- 1897 as the indication for the EndOfBar.
- 1898 The main or most important location **MUST** be designated as the PRIMARY indica-
- 1899 tion for the EndOfBar.
- 1900 If no subType is specified, PRIMARY **MUST** be the default EndOfBar indica-
- 1901 tion.



## 1902 **5.2.44 EquipmentMode**

1903 indication that a piece of equipment, or a sub-part of a piece of equipment, is performing  
1904 specific types of activities.

1905 `EquipmentModeEnum` Enumeration:

1906     • OFF  
1907         equipment is not functioning in the mode designated by the `subType`.

1908     • ON  
1909         equipment is functioning in the mode designated by the `subType`.

1910 A `subType` **MUST** always be specified.

### 1911 **5.2.44.1 Subtypes of EquipmentMode**

1912     • DELAY  
1913         elapsed time of a temporary halt of action.

1914     • LOADED  
1915         indication that the sub-parts of a piece of equipment are under load.

1916         Example: For traditional machine tools, this is an indication that the cutting tool is  
1917         assumed to be engaged with the part.

1918     • OPERATING  
1919         indication that the major sub-parts of a piece of equipment are powered or perform-  
1920         ing any activity whether producing a part or product or not.

1921         Example: For traditional machine tools, this includes when the piece of equipment  
1922         is `WORKING` or it is idle.

1923     • POWERED  
1924         indication that primary power is applied to the piece of equipment and, as a min-  
1925         imum, the controller or logic portion of the piece of equipment is powered and  
1926         functioning or components that are required to remain on are powered.

1927         Example: Heaters for an extrusion machine that required to be powered even when  
1928         the equipment is turned off.

- 1929 • WORKING
- 1930 indication that a piece of equipment is performing any activity, the equipment is
- 1931 active and performing a function under load or not.
- 1932 Example: For traditional machine tools, this includes when the piece of equipment
- 1933 is LOADED, making rapid moves, executing a tool change, etc.

## 1934 5.2.45 Execution

1935 execution status of the Component.

1936 ExecutionEnum Enumeration:

- 1937 • ACTIVE
- 1938 Component is actively executing an instruction.
- 1939 • FEED\_HOLD
- 1940 motion of the active axes are commanded to stop at their current position.
- 1941 • INTERRUPTED
- 1942 Component suspends the execution of the program due to an external signal.
- 1943 Action is required to resume execution.
- 1944 • OPTIONAL\_STOP
- 1945 command from the program has intentionally interrupted execution.
- 1946 The Component **MAY** have another state that indicates if the execution is inter-
- 1947 rupted or the execution ignores the interrupt instruction.
- 1948 • PROGRAM\_COMPLETED
- 1949 program completed execution.
- 1950 • <<deprecated>> PROGRAM\_OPTIONAL\_STOP
- 1951 program has been intentionally optionally stopped using an M01 or similar code.
- 1952 **DEPRECATED** in *version 1.4* and replaced with OPTIONAL\_STOP.
- 1953 • PROGRAM\_STOPPED
- 1954 command from the program has intentionally interrupted execution.
- 1955 Action is required to resume execution.

- 1956     • READY
- 1957       Component is ready to execute instructions.
- 1958       It is currently idle.
- 1959     • STOPPED
- 1960       Component program is not READY to execute.
- 1961     • WAIT
- 1962       Component suspends execution while a secondary operation executes.
- 1963       Execution resumes automatically once the secondary operation completes.

## 1964 5.2.46 FeatureMeasurement

- 1965 tabular representation of assessing elements of a *feature*.
- 1966 FeatureMeasurement **MAY** include a *characteristic* in which case it **MAY** include a
- 1967 CHARACTERISTIC\_STATUS.
- 1968 The Entry key **MUST** be one or more from the FeatureMeasurementResult
- 1969 keys.
- 1970 FeatureMeasurementResult keys:
- 1971     • MeasurementId
- 1972       identifier of this measurement.
- 1973       The value of MeasurementId **MUST** be ID. See *Section 6.1.2 - ID*.
- 1974     • FeaturePersistentId
- 1975       UUID of the feature.
- 1976       The value of FeaturePersistentId **MUST** be ID. See *Section 6.1.2 - ID*.
- 1977     • CharacteristicPersistentId
- 1978       UUID of the characteristic.
- 1979       The value of CharacteristicPersistentId **MUST** be ID. See *Section 6.1.2*
- 1980       - *ID*.

- 1981 • MeasurementType  
 1982 class of measurement being performed. *Ref QIF 3:2018 Section 6.3*  
 1983 **Examples:** POINT, RADIUS, ANGLE, LENGTH, etc.  
 1984 **The value of MeasurementType MUST be string.**
- 1985 • MeasurementValue  
 1986 measurement based on the measurement type.  
 1987 **The value of MeasurementValue MUST be float.**
- 1988 • MeasurementUnits  
 1989 engineering units of the measurement.  
 1990 **The value of MeasurementUnits MUST be string.**
- 1991 • CharacteristicStatus  
 1992 pass/fail result of the measurement.  
 1993 **The value of CharacteristicStatus MUST be one of the Characteris-**  
 1994 **ticStatusEnum enumeration.**  
 1995 **CharacteristicStatusEnum Enumeration:**
- 1996 – BASIC\_OR\_THEORETIC\_EXACT\_DIMENSION  
 1997 nominal provided without tolerance limits. *Ref QIF 3:2018 5.10.2.6*
  - 1998 – FAIL  
 1999 measurement is not within acceptable tolerances.
  - 2000 – INDETERMINATE  
 2001 measurement cannot be determined.
  - 2002 – NOT\_ANALYZED  
 2003 measurement cannot be evaluated.
  - 2004 – PASS  
 2005 measurement is within acceptable tolerances.
  - 2006 – REWORK  
 2007 failed, but acceptable constraints achievable by utilizing additional manufac-  
 2008 turing processes.
  - 2009 – SYSTEM\_ERROR  
 2010 measurement is indeterminate due to an equipment failure.
  - 2011 – UNDEFINED  
 2012 status of measurement cannot be determined.

- 2013     • UncertaintyType  
2014       method used to compute *standard uncertainty*.  
2015       UncertaintyTypeEnum Enumeration:  
2016         – COMBINED  
2017           *combined standard uncertainty*.  
2018         – MEAN  
2019           *standard uncertainty* using arithmetic mean or average the observations. *Ref JCGM*  
2020           *100:2008 4.2*
- 2021     • Uncertainty  
2022       *uncertainty* specified by UNCERTAINTY\_TYPE.  
2023       The value of Uncertainty **MUST** be float.

## 2024 **5.2.47 FeaturePersistantId**

- 2025    UUID of a *feature*. *Ref ISO 10303 AP 242/239*.  
2026    The value of FeaturePersistantId **MUST** be ID. See *Section 6.1.2 - ID*.

## 2027 **5.2.48 Firmware**

- 2028    embedded software of a Component .  
2029    A subType **MUST** always be specified.

### 2030 **5.2.48.1 Subtypes of Firmware**

- 2031     • INSTALL\_DATE  
2032       date the hardware or software was installed.  
2033       The value of Firmware **MUST** be datetime. See *Section 6.1.5 - datetime*.  
2034       datetime Enumeration:  
2035     • LICENSE  
2036       license code to validate or activate the hardware or software.

- 2037 • MANUFACTURER
- 2038 corporate identity for the maker of the hardware or software.
- 2039 • RELEASE\_DATE
- 2040 date the hardware or software was released for general use.
- 2041 The value of `Firmware` **MUST** be `datetime`. See *Section 6.1.5 - datetime*.
- 2042 `datetime` Enumeration:
- 2043 • VERSION
- 2044 version of the hardware or software.

#### 2045 **5.2.49 FixtureId**

2046 identifier for a fixture.

#### 2047 **5.2.50 FunctionalMode**

2048 current intended production status of the `Component`.

2049 `FunctionalModeEnum` Enumeration:

- 2050 • MAINTENANCE
- 2051 `Component` is not currently producing product.
- 2052 It is currently being repaired, waiting to be repaired, or has not yet been returned to
- 2053 a normal production status after maintenance has been performed.
- 2054 • PROCESS\_DEVELOPMENT
- 2055 `Component` is being used to prove-out a new process, testing of equipment or
- 2056 processes, or any other active use that does not result in the production of product.
- 2057 • PRODUCTION
- 2058 `Component` is currently producing product, ready to produce product, or its current
- 2059 intended use is to be producing product.
- 2060 • SETUP
- 2061 `Component` is not currently producing product.
- 2062 It is being prepared or modified to begin production of product.

- 2063 • TEARDOWN
- 2064 Component is not currently producing product.
- 2065 Typically, it has completed the production of a product and is being modified or
- 2066 returned to a neutral state such that it may then be prepared to begin production of a
- 2067 different product.

## 2068 **5.2.51 Hardness**

- 2069 hardness of a material.
- 2070 The value of `Hardness` **MUST** be float.
- 2071 A `subType` **MUST** always be specified.

### 2072 **5.2.51.1 Subtypes of Hardness**

- 2073 • BRINELL
- 2074 scale to measure the resistance to deformation of a surface.
- 2075 • LEEB
- 2076 scale to measure the elasticity of a surface.
- 2077 • MOHS
- 2078 scale to measure the resistance to scratching of a surface.
- 2079 • ROCKWELL
- 2080 scale to measure the resistance to deformation of a surface.
- 2081 • SHORE
- 2082 scale to measure the resistance to deformation of a surface.
- 2083 • VICKERS
- 2084 scale to measure the resistance to deformation of a surface.

## 2085 **5.2.52 Hardware**

2086 hardware of a Component.

2087 A subType **MUST** always be specified.

### 2088 **5.2.52.1 Subtypes of Hardware**

2089 • INSTALL\_DATE

2090 date the hardware or software was installed.

2091 The value of Hardware **MUST** be datetime. See *Section 6.1.5 - datetime*.

2092 datetime Enumeration:

2093 • LICENSE

2094 license code to validate or activate the hardware or software.

2095 • MANUFACTURER

2096 corporate identity for the maker of the hardware or software.

2097 • RELEASE\_DATE

2098 date the hardware or software was released for general use.

2099 The value of Hardware **MUST** be datetime. See *Section 6.1.5 - datetime*.

2100 datetime Enumeration:

2101 • VERSION

2102 version of the hardware or software.

## 2103 **5.2.53 HostName**

2104 name of the host computer supplying data.

## 2105 **5.2.54 LeakDetect**

2106 indication designating whether a leak has been detected.

2107 LeakDetectEnum Enumeration:



- 2108 • DETECTED
- 2109 leak is currently being detected.
- 2110 • NOT\_DETECTED
- 2111 leak is currently not being detected.

## 2112 **5.2.55 Library**

- 2113 software library on a Component
- 2114 A subType **MUST** always be specified.

### 2115 **5.2.55.1 Subtypes of Library**

- 2116 • INSTALL\_DATE
- 2117 date the hardware or software was installed.
- 2118 The value of Library **MUST** be datetime. See *Section 6.1.5 - datetime*.
- 2119 datetime Enumeration:
- 2120 • LICENSE
- 2121 license code to validate or activate the hardware or software.
- 2122 • MANUFACTURER
- 2123 corporate identity for the maker of the hardware or software.
- 2124 • RELEASE\_DATE
- 2125 date the hardware or software was released for general use.
- 2126 The value of Library **MUST** be datetime. See *Section 6.1.5 - datetime*.
- 2127 datetime Enumeration:
- 2128 • VERSION
- 2129 version of the hardware or software.

2130 **5.2.56 <<deprecated>>Line**

2131 current line of code being executed.

2132 **DEPRECATED** in *Version 1.4.0*.

2133 **5.2.56.1 Subtypes of Line**

2134 • MAXIMUM

2135 maximum line number of the code being executed.

2136 • MINIMUM

2137 minimum line number of the code being executed.

2138 **5.2.57 LineLabel**

2139 identifier for a Block of code in a Program.

2140 **5.2.58 LineNumber**

2141 position of a block of program code within a control program.

2142 The value of LineNumber **MUST** be integer.

2143 **5.2.58.1 Subtypes of LineNumber**

2144 • ABSOLUTE

2145 position of a block of program code relative to the beginning of the control program.

2146 • INCREMENTAL

2147 position of a block of program code relative to the occurrence of the last LineLa-  
2148 bel encountered in the control program.

## 2149 **5.2.59 LoadCount**

2150 accumulation of the number of times an operation has attempted to, or is planned to attempt  
2151 to, load materials, parts, or other items.

2152 The default `subType` of `LoadCount` is `ALL`.

2153 The value of `LoadCount` **MUST** be `integer`.

### 2154 **5.2.59.1 Subtypes of LoadCount**

2155 • `ABORTED`

2156 accumulation of actions or activities that were attempted, but terminated before they  
2157 could be completed.

2158 • `ALL`

2159 accumulation of all actions, items, or activities being counted independent of the  
2160 outcome.

2161 • `BAD`

2162 accumulation of actions, items, or activities being counted that do not conform to  
2163 specification or expectation.

2164 • `COMPLETE`

2165 accumulation of actions, items, or activities that have been completed, independent  
2166 of the outcome.

2167 • `FAILED`

2168 accumulation of actions or activities that were attempted, but failed to complete or  
2169 resulted in an unexpected or unacceptable outcome.

2170 • `GOOD`

2171 accumulation of actions, items, or activities being counted that conform to specifi-  
2172 cation or expectation.

2173 • `REMAINING`

2174 accumulation of actions, items, or activities yet to be counted.

2175 • `TARGET`

2176 goal of the operation or process.

## 2177 **5.2.60 LockState**

2178 state or operating mode of a Lock.

2179 LockStateEnum Enumeration:

2180     • LOCKED

2181         mechanism is engaged and preventing the associated Component from being opened  
2182         or operated.

2183     • UNLOCKED

2184         mechanism is disengaged and the associated Component is able to be opened or  
2185         operated.

## 2186 **5.2.61 MTConnectVersion**

2187 reference version of the MTConnect Standard supported by the *adapter*.

## 2188 **5.2.62 MaintenanceList**

2189 actions or activities to be performed in support of a piece of equipment.

2190 If the INTERVAL key is not provided, it is assumed ABSOLUTE.

2191 If the DIRECTION key is not provided, it is assumed UP.

2192 If the UNITS key is not provided, it is assumed to be COUNT.

2193 The Entry key **MUST** be one or more from the MaintenanceListResult keys.

2194 MaintenanceListResult keys:

2195     • Value

2196         current interval value of the activity.

2197         The value of Value **MUST** be float.

2198     • Interval

- 2199 interval of the value observed.
- 2200 MaintenanceListIntervalEnum Enumeration:
- 2201     – ABSOLUTE
- 2202     – INCREMENTAL
- 2203 • NextServiceDate
- 2204     next date/time stamp that maintenance should be performed.
- 2205     The value of NextServiceDate **MUST** be datetime. See *Section 6.1.5 -*
- 2206     *datetime*.
- 2207 • Reset
- 2208     last date/time stamp of the observation was reset.
- 2209     The value of Reset **MUST** be datetime. See *Section 6.1.5 - datetime*.
- 2210 • Severity
- 2211     level of severity on a scale of 1-10.
- 2212     The value of Severity **MUST** be integer.
- 2213 • Direction
- 2214     direction of the value observed.
- 2215 MaintenanceListDirectionEnum Enumeration:
- 2216     – DOWN
- 2217     – UP
- 2218 • Name
- 2219     identifier of the maintenance activity.
- 2220     The value of Name **MUST** be string.
- 2221 • LastServiceDate
- 2222     last date/time stamp that maintenance was performed.
- 2223     The value of LastServiceDate **MUST** be datetime. See *Section 6.1.5 -*
- 2224     *datetime*.
- 2225 • Units
- 2226     same as DataItem units. See *MTCConnect Standard: Part 2.0 - Device Informa-*
- 2227     *tion Model*.
- 2228     The value of Units **MUST** be one of the UnitEnum enumeration.

- 2229     • Target
- 2230     target value of the next maintenance.
- 2231     The value of Target **MUST** be float.

### 2232 **5.2.63 Material**

- 2233 identifier of a material used or consumed in the manufacturing process.

### 2234 **5.2.64 MaterialLayer**

- 2235 identifies the layers of material applied to a part or product as part of an additive manufacturing process.
- 2236
- 2237 The value of MaterialLayer **MUST** be integer.

#### 2238 **5.2.64.1 Subtypes of MaterialLayer**

- 2239     • ACTUAL
- 2240     measured or reported value of an observation.
- 2241     • TARGET
- 2242     goal of the operation or process.

### 2243 **5.2.65 MeasurementType**

- 2244 class of measurement being performed. *Ref QIF 3:2018 Section 6.3*
- 2245 Examples: POINT, RADIUS, ANGLE, LENGTH, etc.

### 2246 **5.2.66 MeasurementUnits**

- 2247 engineering units of the measurement.

**2248 5.2.67 MeasurementValue**

2249 measurement based on the measurement type.

2250 The value of MeasurementValue **MUST** be double.

**2251 5.2.68 Message**

2252 information to be transferred from a piece of equipment to a client software application.

**2253 5.2.68.1 Value Properties of Message**

2254 *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

2255 Descriptions for Value Properties of Message:

2256 • nativeCode

2257 control system local identification of the information being transferred.

**2258 5.2.69 Network**

2259 network details of a Component.

2260 A subType **MUST** always be specified.

**2261 5.2.69.1 Subtypes of Network**

2262 • GATEWAY

2263 Gateway for the component network.

- 2264 • IPV4\_ADDRESS
- 2265 IPV4 network address of the component.
- 2266 • IPV6\_ADDRESS
- 2267 IPV6 network address of the component.
- 2268 • MAC\_ADDRESS
- 2269 Media Access Control Address.
- 2270 The unique physical address of the network hardware.
- 2271 • SUBNET\_MASK
- 2272 SubNet mask for the component network.
- 2273 • VLAN\_ID
- 2274 layer2 Virtual Local Network (VLAN) ID for the component network.
- 2275 • WIRELESS
- 2276 identifies whether the connection type is wireless.
- 2277 NetworkWirelessEnum Enumeration:
  - 2278 – NO
  - 2279 – YES

## 2280 5.2.70 NetworkPort

- 2281 number of the TCP/IP or UDP/IP port for the connection endpoint.
- 2282 The value of NetworkPort **MUST** be integer.

## 2283 5.2.71 OperatingMode

- 2284 state of Component or Composition that describes the automatic or manual operation
- 2285 of the entity.
- 2286 OperatingModeEnum Enumeration:
  - 2287 • AUTOMATIC
  - 2288 automatically execute instructions from a recipe or program.



2289 Note: Setpoint comes from a recipe.

2290 • MANUAL

2291 execute instructions from an external agent or person.

2292 Note 1 to entry: Valve or switch is manipulated by an agent/person.

2293 Note 2 to entry: Direct control of the PID output. % of the range: A user  
2294 manually sets the % output, not the setpoint.

2295 • SEMI\_AUTOMATIC

2296 executes a single instruction from a recipe or program.

2297 Note 1 to entry: Setpoint is entered and fixed, but the PID is controlling.

2298 Note 2 to entry: Still goes through the PID control system.

2299 Note 3 to entry: Manual fixed entry from a recipe.

## 2300 5.2.72 OperatingSystem

2301 Operating System (OS) of a Component.

2302 A subType **MUST** always be specified.

### 2303 5.2.72.1 Subtypes of OperatingSystem

2304 • INSTALL\_DATE

2305 date the hardware or software was installed.

2306 The value of OperatingSystem **MUST** be datetime. See *Section 6.1.5 -*  
2307 *datetime*.

2308 datetime Enumeration:

2309 • LICENSE

2310 license code to validate or activate the hardware or software.

2311 • MANUFACTURER

2312 corporate identity for the maker of the hardware or software.

- 2313     • RELEASE\_DATE  
2314     date the hardware or software was released for general use.  
2315     The value of OperatingSystem **MUST** be datetime. See *Section 6.1.5 -*  
2316     *datetime*.  
2317     datetime Enumeration:  
2318     • VERSION  
2319     version of the hardware or software.

### 2320 **5.2.73 OperatorId**

2321 identifier of the person currently responsible for operating the piece of equipment.

### 2322 **5.2.74 PalletId**

2323 identifier for a pallet.

### 2324 **5.2.75 PartCount**

2325 aggregate count of parts.

2326 The value of PartCount **MUST** be integer.

2327 The default subType of PartCount is ALL.

#### 2328 **5.2.75.1 Subtypes of PartCount**

- 2329     • ABORTED  
2330     accumulation of actions or activities that were attempted, but terminated before they  
2331     could be completed.  
2332     • ALL  
2333     accumulation of all actions, items, or activities being counted independent of the  
2334     outcome.

- 2335 • BAD
- 2336 accumulation of actions, items, or activities being counted that do not conform to
- 2337 specification or expectation.
- 2338 • COMPLETE
- 2339 accumulation of actions, items, or activities that have been completed, independent
- 2340 of the outcome.
- 2341 • FAILED
- 2342 accumulation of actions or activities that were attempted, but failed to complete or
- 2343 resulted in an unexpected or unacceptable outcome.
- 2344 • GOOD
- 2345 accumulation of actions, items, or activities being counted that conform to specifi-
- 2346 cation or expectation.
- 2347 • REMAINING
- 2348 accumulation of actions, items, or activities yet to be counted.
- 2349 • TARGET
- 2350 goal of the operation or process.

### 2351 **5.2.76 PartCountType**

2352 interpretation of PART\_COUNT.

2353 PartCountTypeEnum Enumeration:

- 2354 • BATCH
- 2355 pre-specified group of items.
- 2356 • EACH
- 2357 count is of individual items.

### 2358 **5.2.77 PartDetect**

2359 indication designating whether a part or work piece has been detected or is present.

2360 PartDetectEnum Enumeration:

- 2361     • NOT\_PRESENT
- 2362     part or work piece is not detected or is not present.
- 2363     • PRESENT
- 2364     part or work piece is detected or is present.

## 2365 **5.2.78 PartGroupId**

- 2366 identifier given to a collection of individual parts.
- 2367 If no subType is specified, UUID is default.
- 2368 The default subType of PartGroupId is UUID.

### 2369 **5.2.78.1 Subtypes of PartGroupId**

- 2370     • BATCH
- 2371     identifier that references a group of parts produced in a batch.
- 2372     • HEAT\_TREAT
- 2373     identifier used to reference a material heat number.
- 2374     • LOT
- 2375     identifier that references a group of parts tracked as a lot.
- 2376     • RAW\_MATERIAL
- 2377     material that is used to produce parts.
- 2378     • UUID
- 2379     universally unique identifier as specified in ISO 11578 or RFC 4122.

## 2380 **5.2.79 PartId**

- 2381 identifier of a part in a manufacturing operation.

## 2382 **5.2.80 PartKindId**

2383 identifier given to link the individual occurrence to a class of parts, typically distinguished  
2384 by a particular part design.

2385 If no subType is specified, UUID is default.

2386 The default subType of PartKindId is UUID.

### 2387 **5.2.80.1 Subtypes of PartKindId**

2388 • PART\_FAMILY

2389 identifier given to a group of parts having similarities in geometry, manufacturing  
2390 process, and/or functions.

2391 • PART\_NAME

2392 word or set of words by which a part is known, addressed, or referred to.

2393 • PART\_NUMBER

2394 identifier of a particular part design or model.

2395 • UUID

2396 universally unique identifier as specified in ISO 11578 or RFC 4122.

## 2397 **5.2.81 <<deprecated>>PartNumber**

2398 identifier of a part or product moving through the manufacturing process.

2399 **DEPRECATED** in *Version 1.7*. PART\_NUMBER is now a subType of PART\_KIND\_ID.

## 2400 **5.2.82 PartProcessingState**

2401 particular condition of the part occurrence at a specific time.

2402 PartProcessingStateEnum Enumeration:

- 2403 • IN\_PROCESS
- 2404 part occurrence is actively being processed.
- 2405 • IN\_TRANSIT
- 2406 part occurrence is being transported to its destination.
- 2407 • NEEDS\_PROCESSING
- 2408 part occurrence is not actively being processed, but the processing has not ended.
- 2409 Processing requirements exist that have not yet been fulfilled. This is the default
- 2410 entry state when the part occurrence is originally received. In some cases, the part
- 2411 occurrence may return to this state while it waits for additional processing to be
- 2412 performed.
- 2413 • PROCESSING\_ENDED
- 2414 part occurrence is no longer being processed.
- 2415 A general state when the reason for termination is unknown.
- 2416 • PROCESSING\_ENDED\_ABORTED
- 2417 processing of the part occurrence has come to a premature end.
- 2418 • PROCESSING\_ENDED\_COMPLETE
- 2419 part occurrence has completed processing successfully.
- 2420 • PROCESSING\_ENDED\_LOST
- 2421 terminal state when the part occurrence has been removed from the equipment by
- 2422 an external entity and it no longer exists at the equipment.
- 2423 • PROCESSING\_ENDED\_REJECTED
- 2424 part occurrence has been processed completely. However, the processing may have
- 2425 a problem.
- 2426 • PROCESSING\_ENDED\_SKIPPED
- 2427 part occurrence has been skipped for processing on the piece of equipment.
- 2428 • PROCESSING\_ENDED\_STOPPED
- 2429 process has been stopped during the processing.
- 2430 The part occurrence will require special treatment.
- 2431 • TRANSIT\_COMPLETE
- 2432 part occurrence has been placed at its designated destination.
- 2433 • WAITING\_FOR\_TRANSIT
- 2434 part occurrence is waiting for transit.

### 2435 **5.2.83 PartStatus**

2436 state or condition of a part.

2437 If unique identifier is given, part status is for that individual. If group identifier is given  
2438 without a unique identifier, then the status is assumed to be for the whole group.

2439 `PartStatusEnum` Enumeration:

2440     • FAIL

2441         part does not conform to some given requirements.

2442     • PASS

2443         part conforms to given requirements.

### 2444 **5.2.84 PartUniqueId**

2445 identifier given to a distinguishable, individual part.

2446 If no `subType` is specified, `UUID` is default.

2447 The default `subType` of `PartUniqueId` is `UUID`.

#### 2448 **5.2.84.1 Subtypes of PartUniqueId**

2449     • `RAW_MATERIAL`

2450         material that is used to produce parts.

2451     • `SERIAL_NUMBER`

2452         serial number that uniquely identifies a specific part.

2453     • `UUID`

2454         universally unique identifier as specified in ISO 11578 or RFC 4122.

## 2455 **5.2.85 PathFeedrateOverride**

2456 value of a signal or calculation issued to adjust the feedrate for the axes associated with a  
2457 Path component that may represent a single axis or the coordinated movement of multiple  
2458 axes.

2459 The value of PathFeedrateOverride **MUST** be float.

### 2460 **5.2.85.1 Subtypes of PathFeedrateOverride**

- 2461 • JOG  
2462 relating to momentary activation of a function or a movement.  
2463 **DEPRECATION WARNING:** May be deprecated in the future.
- 2464 • PROGRAMMED  
2465 directive value without offsets and adjustments.
- 2466 • RAPID  
2467 performing an operation faster or in less time than nominal rate.

## 2468 **5.2.86 PathMode**

2469 describes the operational relationship between a Path entity and another Path entity for  
2470 pieces of equipment comprised of multiple logical groupings of controlled axes or other  
2471 logical operations.

2472 PathModeEnum Enumeration:

- 2473 • INDEPENDENT  
2474 path is operating independently and without the influence of another path.
- 2475 • MASTER  
2476 path provides information or state values that influences the operation of other DataItem  
2477 of similar type.
- 2478 • MIRROR  
2479 axes associated with the path are mirroring the motion of the MASTER path.



- 2480 • SYNCHRONOUS
- 2481 physical or logical parts which are not physically connected to each other but are
- 2482 operating together.

### 2483 **5.2.87 PowerState**

2484 indication of the status of the source of energy for an entity to allow it to perform its  
 2485 intended function or the state of an enabling signal providing permission for the entity to  
 2486 perform its functions.

2487 `PowerStateEnum` Enumeration:

- 2488 • OFF
- 2489 source of energy for an entity or the enabling signal providing permission for the
- 2490 entity to perform its function(s) is not present or is disconnected.
- 2491 • ON
- 2492 source of energy for an entity or the enabling signal providing permission for the
- 2493 entity to perform its function(s) is present and active.

#### 2494 **5.2.87.1 Subtypes of PowerState**

- 2495 • CONTROL
- 2496 state of the enabling signal or control logic that enables or disables the function or
- 2497 operation of the entity.
- 2498 • LINE
- 2499 state of the power source for the entity.

### 2500 **5.2.88 <<deprecated>>PowerStatus**

2501 status of the Component.

2502 **DEPRECATED** in *Version 1.1.0*.

2503 <<deprecated>> `PowerStatusEnum` Enumeration:

2504 • <<deprecated>> OFF

2505 • <<deprecated>> ON

## 2506 **5.2.89 ProcessAggregateId**

2507 identifier given to link the individual occurrence to a group of related occurrences, such as  
2508 a process step in a process plan.

### 2509 **5.2.89.1 Subtypes of ProcessAggregateId**

2510 • ORDER\_NUMBER

2511 identifier of the authorization of the process occurrence. Synonyms include “job id”,  
2512 “work order”.

2513 • PROCESS\_PLAN

2514 identifier of the process plan that this occurrence belongs to. Synonyms include  
2515 “routing id”, “job id”.

2516 • PROCESS\_STEP

2517 identifier of the step in the process plan that this occurrence corresponds to. Syn-  
2518 onyms include “operation id”.

## 2519 **5.2.90 ProcessKindId**

2520 identifier given to link the individual occurrence to a class of processes or process defini-  
2521 tion.

### 2522 **5.2.90.1 Subtypes of ProcessKindId**

2523 • ISO\_STEP\_EXECUTABLE

2524 reference to a ISO 10303 Executable.

2525 • PROCESS\_NAME

2526 word or set of words by which a process being executed (process occurrence) by the  
2527 device is known, addressed, or referred to.

- 2528 • UUID
- 2529     universally unique identifier as specified in ISO 11578 or RFC 4122.

### 2530 **5.2.91 ProcessOccurrenceId**

2531 identifier of a process being executed by the device.

#### 2532 **5.2.91.1 Subtypes of ProcessOccurrenceId**

- 2533 • ACTIVITY
- 2534     phase or segment of a recipe or program.
- 2535 • OPERATION
- 2536     step of a discrete manufacturing process.
- 2537 • RECIPE
- 2538     process as part of product production; can be a subprocess of a larger process.
- 2539 • SEGMENT
- 2540     phase of a recipe process.

### 2541 **5.2.92 ProcessState**

2542 particular condition of the process occurrence at a specific time.

2543 `ProcessStateEnum` Enumeration:

- 2544 • ABORTED
- 2545     process occurrence has come to a premature end and cannot be resumed.
- 2546 • ACTIVE
- 2547     process occurrence is actively executing.
- 2548 • COMPLETE
- 2549     process occurrence is now finished.

- 2550     • INITIALIZING
- 2551     device is preparing to execute the process occurrence.
- 2552     • INTERRUPTED
- 2553     process occurrence has been stopped and may be resumed.
- 2554     • READY
- 2555     process occurrence is ready to be executed.

## 2556 **5.2.93 ProcessTime**

- 2557     time and date associated with an activity or event.
- 2558     A subType **MUST** always be specified.

### 2559 **5.2.93.1 Subtypes of ProcessTime**

- 2560     • COMPLETE
- 2561     time and date associated with the completion of an activity or event.
- 2562     • START
- 2563     boundary when an activity or an event commences.
- 2564     • TARGET\_COMPLETION
- 2565     projected time and date associated with the end or completion of an activity or event.

## 2566 **5.2.94 Program**

- 2567     name of the logic or motion program being executed by the Controller component.

### 2568 **5.2.94.1 Subtypes of Program**

- 2569     • ACTIVE
- 2570     identity of the logic or motion program currently executing.

- 2571 • ACTIVITY
- 2572 phase or segment of a recipe or program.
- 2573 • MAIN
- 2574 identity of the primary logic or motion program currently being executed.
- 2575 It is the starting nest level in a call structure and may contain calls to sub programs.
- 2576 • OPERATION
- 2577 step of a discrete manufacturing process.
- 2578 • RECIPE
- 2579 process as part of product production; can be a subprocess of a larger process.
- 2580 • SCHEDULE
- 2581 identity of a control program that is used to specify the order of execution of other
- 2582 programs.
- 2583 • SEGMENT
- 2584 phase of a recipe process.

## 2585 **5.2.95 ProgramComment**

2586 comment or non-executable statement in the control program.

### 2587 **5.2.95.1 Subtypes of ProgramComment**

- 2588 • ACTIVE
- 2589 identity of the logic or motion program currently executing.
- 2590 • MAIN
- 2591 identity of the primary logic or motion program currently being executed.
- 2592 It is the starting nest level in a call structure and may contain calls to sub programs.
- 2593 • SCHEDULE
- 2594 identity of a control program that is used to specify the order of execution of other
- 2595 programs.

## 2596 **5.2.96 ProgramEdit**

2597 indication of the status of the Controller components program editing mode.

2598 A program may be edited while another is executed.

2599 ProgramEditEnum Enumeration:

2600     • ACTIVE

2601         Controller is in the program edit mode.

2602     • NOT\_READY

2603         Controller is being inhibited by a function from entering the program edit mode.

2604     • READY

2605         Controller is capable of entering the program edit mode and no function is  
2606         inhibiting a change to that mode.

## 2607 **5.2.97 ProgramEditName**

2608 name of the program being edited.

2609 This is used in conjunction with ProgramEdit when in ACTIVE state.

## 2610 **5.2.98 ProgramHeader**

2611 non-executable header section of the control program.

2612 The default subType of ProgramHeader is MAIN.

### 2613 **5.2.98.1 Subtypes of ProgramHeader**

2614     • ACTIVE

2615         identity of the logic or motion program currently executing.

- 2616 • MAIN
- 2617 identity of the primary logic or motion program currently being executed.
- 2618 It is the starting nest level in a call structure and may contain calls to sub programs.
- 2619 • SCHEDULE
- 2620 identity of a control program that is used to specify the order of execution of other
- 2621 programs.

## 2622 5.2.99 ProgramLocation

2623 URI for the source file associated with Program.

### 2624 5.2.99.1 Subtypes of ProgramLocation

- 2625 • ACTIVE
- 2626 identity of the logic or motion program currently executing.
- 2627 • MAIN
- 2628 identity of the primary logic or motion program currently being executed.
- 2629 It is the starting nest level in a call structure and may contain calls to sub programs.
- 2630 • SCHEDULE
- 2631 identity of a control program that is used to specify the order of execution of other
- 2632 programs.

## 2633 5.2.100 ProgramLocationType

2634 defines whether the logic or motion program defined by Program is being executed from  
2635 the local memory of the controller or from an outside source.

2636 ProgramLocationTypeEnum Enumeration:

- 2637 • EXTERNAL
- 2638 not managed by the controller.
- 2639 • LOCAL
- 2640 managed by the controller.

2641 **5.2.100.1 Subtypes of ProgramLocationType**

- 2642     • ACTIVE
- 2643         identity of the logic or motion program currently executing.
- 2644     • MAIN
- 2645         identity of the primary logic or motion program currently being executed.
- 2646         It is the starting nest level in a call structure and may contain calls to sub programs.
- 2647     • SCHEDULE
- 2648         identity of a control program that is used to specify the order of execution of other
- 2649         programs.

2650 **5.2.101 ProgramNestLevel**

- 2651 indication of the nesting level within a control program that is associated with the code or
- 2652 instructions that is currently being executed.
- 2653 If an initial value is not defined, the nesting level associated with the highest or initial
- 2654 nesting level of the program **MUST** default to zero (0).
- 2655 The value of ProgramNestLevel **MUST** be integer.

2656 **5.2.102 RotaryMode**

- 2657 current operating mode for a Rotary type axis.
- 2658 RotaryModeEnum Enumeration:
- 2659     • CONTOUR
  - 2660         position of the axis is being interpolated.
  - 2661     • INDEX
  - 2662         axis is configured to index.
  - 2663     • SPINDLE
  - 2664         axis is functioning as a spindle.



### 2665 **5.2.103 RotaryVelocityOverride**

2666 percentage change to the velocity of the programmed velocity for a Rotary axis.

2667 This command represents a percentage change to the velocity calculated by a logic or  
2668 motion program or set by a switch for a Rotary type axis.

2669 The value of RotaryVelocityOverride **MUST** be float.

### 2670 **5.2.104 Rotation**

2671 three space angular displacement of an object or coordinate system relative to a *cartesian*  
2672 *coordinate system*.

2673 The units of Rotation **MUST** be DEGREE\_3D.

2674 The value of Rotation **MUST** be a list of float of size 0 . . 3.

### 2675 **5.2.105 SensorAttachment**

2676 *attachment* between a sensor and an entity.

2677 The Entry key **MUST** be one or more from the SensorAttachmentResult keys.

2678 SensorAttachmentResult keys:

2679     • SensorId

2680         identity of a sensor used to observe some measurement of an item.

2681         The value of SensorId **MUST** be ID. See *Section 6.1.2 - ID*.

### 2682 **5.2.106 SensorState**

2683 detection result of a sensor.

2684 The default subType of SensorState is BINARY.

### 2685 **5.2.106.1 Subtypes of SensorState**

2686 • BINARY

2687 detection result of a sensor.

2688 where the state is observed as a binary data type.

2689 The value of `SensorState` **MUST** be binary. See *Section 6.1.18 - binary*.

2690 binary Enumeration:

2691 • BOOLEAN

2692 detection result of a sensor.

2693 where the state is observed as a boolean data type.

2694 The value of `SensorState` **MUST** be boolean.

2695 boolean Enumeration:

2696 • DETECT

2697 detection result of a sensor.

2698 where the state is indicated by the presence or existence of something.

2699 The value of `SensorState` **MUST** be one of the `CompositionStateSwitchedE-`  
2700 `num` enumeration.

2701 • ENUMERATED

2702 detection result of a sensor.

2703 where the state is observed as a set containing a restricted number of discrete values  
2704 where each discrete value is named and unique. *Ref ISO 21961:2003, 013*

2705 The value of `SensorState` **MUST** be integer.

2706 integer Enumeration:

### 2707 **5.2.107 SerialNumber**

2708 serial number associated with a `Component`, `Asset`, or `Device`.

## 2709 **5.2.108 SpecificationLimit**

2710 set of limits defining a range of values designating acceptable performance for a variable.

2711 The Entry key **MUST** be one or more from the SpecificationLimitResult  
2712 keys.

2713 SpecificationLimitResult keys:

2714 • UpperLimit

2715 upper conformance boundary for a variable.

2716 Note: immediate concern or action may be required.

2717 The value of UpperLimit **MUST** be float.

2718 • Nominal

2719 ideal or desired value for a variable.

2720 The value of Nominal **MUST** be float.

2721 • LowerLimit

2722 lower conformance boundary for a variable.

2723 Note: immediate concern or action may be required.

2724 The value of LowerLimit **MUST** be float.

## 2725 **5.2.109 SpindleInterlock**

2726 indication of the status of the spindle for a piece of equipment when power has been  
2727 removed and it is free to rotate.

2728 SpindleInterlockEnum Enumeration:

2729 • ACTIVE

2730 power has been removed and the spindle cannot be operated.

2731 • INACTIVE

2732 spindle has not been deactivated.

**2733 5.2.110 ToolAssetId**

2734 identifier of an individual tool asset.

**2735 5.2.111 ToolGroup**

2736 identifier for the tool group associated with a specific tool. Commonly used to designate  
2737 spare tools.

**2738 5.2.112 <<deprecated>>ToolId**

2739 identifier of the tool currently in use for a given Path.

2740 **DEPRECATED** in *Version 1.2.0*. See TOOL\_ASSET\_ID.

**2741 5.2.113 ToolNumber**

2742 identifier assigned by the Controller component to a cutting tool when in use by a  
2743 piece of equipment.

**2744 5.2.114 ToolOffset**

2745 reference to the tool offset variables applied to the active cutting tool.

2746 The value of ToolOffset **MUST** be float.

2747 A subType **MUST** always be specified.

**2748 5.2.114.1 Subtypes of ToolOffset**

2749 • LENGTH

2750 reference to a length type tool offset variable.

2751 • RADIAL

2752 reference to a radial type tool offset variable.

**2753 5.2.115 ToolOffsets**

2754 tabular representation of properties of each addressable tool offset.

**2755 5.2.116 TransferCount**

2756 accumulation of the number of times an operation has attempted to, or is planned to attempt  
2757 to, transfer materials, parts, or other items from one location to another.

2758 The default subType of TransferCount is ALL.

2759 The value of TransferCount **MUST** be integer.

**2760 5.2.116.1 Subtypes of TransferCount**

2761 • ABORTED

2762 accumulation of actions or activities that were attempted, but terminated before they  
2763 could be completed.

2764 • ALL

2765 accumulation of all actions, items, or activities being counted independent of the  
2766 outcome.

2767 • BAD

2768 accumulation of actions, items, or activities being counted that do not conform to  
2769 specification or expectation.

2770 • COMPLETE

2771 accumulation of actions, items, or activities that have been completed, independent  
2772 of the outcome.

2773 • FAILED

2774 accumulation of actions, items, or activities being counted that do not conform to  
2775 specification or expectation.

2776 • GOOD

2777 accumulation of actions, items, or activities being counted that conform to speci-  
2778 fication or expectation.

- 2779     • REMAINING
- 2780       accumulation of actions, items, or activities yet to be counted.
- 2781     • TARGET
- 2782       goal of the operation or process.

### 2783 **5.2.117 Translation**

2784 three space linear displacement of an object or coordinate system relative to a *cartesian*  
2785 *coordinate system*.

2786 The units of Translation **MUST** be MILLIMETER\_3D.

2787 The value of Translation **MUST** be a list of float of size 0..3.

### 2788 **5.2.118 Uncertainty**

2789 *uncertainty* specified by UncertaintyType.

2790 The value of Uncertainty **MUST** be double.

### 2791 **5.2.119 UncertaintyType**

2792 method used to compute *standard uncertainty*.

2793 The value of UncertaintyType **MUST** be one of the UncertaintyTypeEnum  
2794 enumeration.

2795 UncertaintyTypeEnum Enumeration:

- 2796     • COMBINED
- 2797       *combined standard uncertainty*.
- 2798     • MEAN
- 2799       *standard uncertainty* using arithmetic mean or average the observations. *Ref JCGM*  
2800       *100:2008 4.2*

**2801 5.2.120 UnloadCount**

2802 accumulation of the number of times an operation has attempted to, or is planned to attempt  
2803 to, unload materials, parts, or other items.

2804 The default subType of UnloadCount is ALL.

2805 The value of UnloadCount **MUST** be integer.

**2806 5.2.120.1 Subtypes of UnloadCount**

2807 • ABORTED

2808 accumulation of actions or activities that were attempted, but terminated before they  
2809 could be completed.

2810 • ALL

2811 accumulation of all actions, items, or activities being counted independent of the  
2812 outcome.

2813 • BAD

2814 accumulation of actions, items, or activities being counted that do not conform to  
2815 specification or expectation.

2816 • COMPLETE

2817 accumulation of actions, items, or activities that have been completed, independent  
2818 of the outcome.

2819 • FAILED

2820 accumulation of actions, items, or activities being counted that do not conform to  
2821 specification or expectation.

2822 • GOOD

2823 accumulation of actions, items, or activities being counted that conform to specifi-  
2824 cation or expectation.

2825 • REMAINING

2826 accumulation of actions, items, or activities yet to be counted.

2827 • TARGET

2828 goal of the operation or process.

## 2829 **5.2.121 User**

2830 identifier of the person currently responsible for operating the piece of equipment.

2831 A subType **MUST** always be specified.

### 2832 **5.2.121.1 Subtypes of User**

2833 • MAINTENANCE

2834 identifier of the person currently responsible for performing maintenance on the  
2835 piece of equipment.

2836 • OPERATOR

2837 identifier of the person currently responsible for operating the piece of equipment.

2838 • SET\_UP

2839 identifier of the person currently responsible for preparing a piece of equipment for  
2840 production or restoring the piece of equipment to a neutral state after production.

## 2841 **5.2.122 ValveState**

2842 state of a valve is one of open, closed, or transitioning between the states.

2843 ValveStateEnum Enumeration:

2844 • CLOSED

2845 ValveState where flow is not possible, the aperture is static, and the valve is  
2846 completely shut.

2847 • CLOSING

2848 valve is transitioning from an OPEN state to a CLOSED state.

2849 • OPEN

2850 ValveState where flow is allowed and the aperture is static.

2851 Note: For a binary value, OPEN indicates the valve has the maximum  
2852 possible aperture.

2853 • OPENING

2854 valve is transitioning from a CLOSED state to an OPEN state.



2855 **5.2.122.1 Subtypes of ValveState**

- 2856 • ACTUAL
- 2857 measured or reported value of an observation.
- 2858 • PROGRAMMED
- 2859 directive value without offsets and adjustments.

2860 **5.2.123 Variable**

2861 data whose meaning may change over time due to changes in the operation of a piece of  
2862 equipment or the process being executed on that piece of equipment.

2863 **5.2.124 WaitState**

2864 indication of the reason that Execution is reporting a value of WAIT.

2865 When Execution result is not WAIT, the isUnavailable property of Wait-  
2866 State **MUST** be true.

2867 WaitStateEnum Enumeration:

- 2868 • MATERIAL\_LOAD
- 2869 execution is waiting while material is being loaded.
- 2870 • MATERIAL\_UNLOAD
- 2871 execution is waiting while material is being unloaded.
- 2872 • PART\_LOAD
- 2873 execution is waiting while one or more discrete workpieces are being loaded.
- 2874 • PART\_UNLOAD
- 2875 execution is waiting while one or more discrete workpieces are being unloaded.
- 2876 • PAUSING
- 2877 execution is waiting while the equipment is pausing but the piece of equipment has  
2878 not yet reached a fully paused state.

- 2879     • POWERING\_DOWN
- 2880     execution is waiting while the equipment is powering down but has not fully reached
- 2881     a stopped state.
  
- 2882     • POWERING\_UP
- 2883     execution is waiting while the equipment is powering up and is not currently avail-
- 2884     able to begin producing parts or products.
  
- 2885     • RESUMING
- 2886     execution is waiting while the equipment is resuming the production cycle but has
- 2887     not yet resumed execution.
  
- 2888     • SECONDARY\_PROCESS
- 2889     execution is waiting while another process is completed before the execution can
- 2890     resume.
  
- 2891     • TOOL\_LOAD
- 2892     execution is waiting while a tool or tooling is being loaded.
  
- 2893     • TOOL\_UNLOAD
- 2894     execution is waiting while a tool or tooling is being unloaded.

**2895 5.2.125 Wire**

2896 identifier for the type of wire used as the cutting mechanism in Electrical Discharge Ma-  
2897 chining or similar processes.

**2898 5.2.126 WorkOffset**

2899 reference to offset variables for a work piece or part.

**2900 5.2.127 WorkOffsets**

2901 tabular representation of properties of each addressable work offset.

## 2902 **5.2.128 WorkholdingId**

2903 identifier for the current workholding or part clamp in use by a piece of equipment.

## 2904 **5.3 Sample Types**

2905 This section provides semantic information for Sample types.

### 2906 **5.3.1 Acceleration**

2907 positive rate of change of velocity.

2908 The units of Acceleration **MUST** be MILLIMETER/SECOND<sup>2</sup>.

2909 The default subType of Acceleration is ACTUAL.

#### 2910 **5.3.1.1 Subtypes of Acceleration**

2911 • ACTUAL

2912 measured or reported value of an observation.

2913 • COMMANDED

2914 directive value including adjustments such as an offset or overrides.

2915 • PROGRAMMED

2916 directive value without offsets and adjustments.

### 2917 **5.3.2 AccumulatedTime**

2918 accumulated time for an activity or event.

2919 The units of AccumulatedTime **MUST** be SECOND.

### 2920 **5.3.3 <<deprecated>>Amperage**

2921 strength of electrical current.

2922 **DEPRECATED** in *Version 1.6*. Replaced by AMPERAGE\_AC and AMPERAGE\_DC.

2923 The units of Amperage **MUST** be AMPERE.

#### 2924 **5.3.3.1 Subtypes of Amperage**

2925 • ACTUAL

2926 measured or reported value of an observation.

2927 **DEPRECATED** in *Version 1.6*.

2928 • ALTERNATING

2929 measurement of alternating voltage or current.

2930 If not specified further in statistic, defaults to RMS voltage.

2931 **DEPRECATED** in *Version 1.6*.

2932 • DIRECT

2933 measurement of DC current or voltage.

2934 **DEPRECATED** in *Version 1.6*.

2935 • TARGET

2936 goal of the operation or process.

2937 **DEPRECATED** in *Version 1.6*.

### 2938 **5.3.4 AmperageAC**

2939 electrical current that reverses direction at regular short intervals.

2940 The units of AmperageAC **MUST** be AMPERE.

2941 A subType **MUST** always be specified.

2942 **5.3.4.1 Subtypes of AmperageAC**

- 2943 • ACTUAL
- 2944 measured or reported value of an observation.
- 2945 • COMMANDED
- 2946 directive value including adjustments such as an offset or overrides.
- 2947 • PROGRAMMED
- 2948 directive value without offsets and adjustments.

2949 **5.3.5 AmperageDC**

- 2950 electric current flowing in one direction only.
- 2951 The units of AmperageDC **MUST** be AMPERE.
- 2952 A subType **MUST** always be specified.

2953 **5.3.5.1 Subtypes of AmperageDC**

- 2954 • ACTUAL
- 2955 measured or reported value of an observation.
- 2956 • COMMANDED
- 2957 directive value including adjustments such as an offset or overrides.
- 2958 • PROGRAMMED
- 2959 directive value without offsets and adjustments.

2960 **5.3.6 Angle**

- 2961 angular position.
- 2962 The units of Angle **MUST** be DEGREE.

2963 **5.3.6.1 Subtypes of Angle**

- 2964 • ACTUAL
- 2965 measured or reported value of an observation.
- 2966 • COMMANDED
- 2967 directive value including adjustments such as an offset or overrides.

2968 **5.3.7 AngularAcceleration**

- 2969 positive rate of change of angular velocity.
- 2970 The units of AngularAcceleration **MUST** be DEGREE/SECOND<sup>2</sup>.
- 2971 The default subType of AngularAcceleration is ACTUAL.

2972 **5.3.7.1 Subtypes of AngularAcceleration**

- 2973 • ACTUAL
- 2974 measured or reported value of an observation.
- 2975 • COMMANDED
- 2976 directive value including adjustments such as an offset or overrides.
- 2977 • PROGRAMMED
- 2978 directive value without offsets and adjustments.

2979 **5.3.8 AngularDeceleration**

- 2980 negative rate of change of angular velocity.
- 2981 The units of AngularDeceleration **MUST** be DEGREE/SECOND<sup>2</sup>.
- 2982 The default subType of AngularDeceleration is ACTUAL.

2983 **5.3.8.1 Subtypes of AngularDeceleration**

- 2984 • ACTUAL
- 2985 measured or reported value of an observation.
- 2986 • COMMANDED
- 2987 directive value including adjustments such as an offset or overrides.
- 2988 • PROGRAMMED
- 2989 directive value without offsets and adjustments.

2990 **5.3.9 AngularVelocity**

- 2991 rate of change of angular position.
- 2992 The units of AngularVelocity **MUST** be DEGREE/SECOND.

2993 **5.3.10 AssetUpdateRate**

- 2994 average rate of change of values for assets in the MTConnect streams.
- 2995 The average is computed over a rolling window defined by the implementation.
- 2996 The units of AssetUpdateRate **MUST** be COUNT/SECOND.

2997 **5.3.11 AxisFeedrate**

- 2998 feedrate of a linear axis.
- 2999 The units of AxisFeedrate **MUST** be MILLIMETER/SECOND.

3000 **5.3.11.1 Subtypes of AxisFeedrate**

- 3001 • ACTUAL
- 3002 measured or reported value of an observation.

- 3003 • COMMANDED
- 3004 directive value including adjustments such as an offset or overrides.
- 3005 • JOG
- 3006 relating to momentary activation of a function or a movement.
- 3007 **DEPRECATION WARNING:** May be deprecated in the future.
- 3008 • OVERRIDE
- 3009 operator's overridden value.
- 3010 • PROGRAMMED
- 3011 directive value without offsets and adjustments.
- 3012 • RAPID
- 3013 performing an operation faster or in less time than nominal rate.

### 3014 **5.3.12 BatteryCapacity**

- 3015 maximum rated charge a battery is capable of maintaining based on the battery discharging
- 3016 at a specified current over a specified time period.
- 3017 The units of BatteryCapacity **MUST** be COULOMB.
- 3018 The default subType of BatteryCapacity is ACTUAL.

#### 3019 **5.3.12.1 Subtypes of BatteryCapacity**

- 3020 • ACTUAL
- 3021 measured or reported value of an observation.
- 3022 • TARGET
- 3023 goal of the operation or process.



### 3024 **5.3.13 BatteryCharge**

3025 value of the battery's present capacity expressed as a percentage of the battery's maximum  
3026 rated capacity.

3027 The units of BatteryCharge **MUST** be PERCENT.

3028 The default subType of BatteryCharge is ACTUAL.

#### 3029 **5.3.13.1 Subtypes of BatteryCharge**

3030     • ACTUAL  
3031         measured or reported value of an observation.

3032     • TARGET  
3033         goal of the operation or process.

### 3034 **5.3.14 CapacityFluid**

3035 fluid capacity of an object or container.

3036 The units of CapacityFluid **MUST** be MILLILITER.

### 3037 **5.3.15 CapacitySpatial**

3038 geometric capacity of an object or container.

3039 The units of CapacitySpatial **MUST** be CUBIC\_MILLIMETER.

### 3040 **5.3.16 ChargeRate**

3041 value of the current being supplied to the Component for the purpose of charging.

3042 The units of ChargeRate **MUST** be AMPERE.

3043 The default subType of ChargeRate is ACTUAL.

3044 **5.3.16.1 Subtypes of ChargeRate**

- 3045 • ACTUAL
- 3046 measured or reported value of an observation.
- 3047 • TARGET
- 3048 goal of the operation or process.

3049 **5.3.17 Concentration**

- 3050 percentage of one component within a mixture of components.
- 3051 The units of Concentration **MUST** be PERCENT.

3052 **5.3.18 Conductivity**

- 3053 ability of a material to conduct electricity.
- 3054 The units of Conductivity **MUST** be SIEMENS/METER.

3055 **5.3.19 CuttingSpeed**

- 3056 speed difference (relative velocity) between the cutting mechanism and the surface of the
- 3057 workpiece it is operating on.
- 3058 The units of CuttingSpeed **MUST** be MILLIMETER/SECOND.

3059 **5.3.19.1 Subtypes of CuttingSpeed**

- 3060 • ACTUAL
- 3061 measured or reported value of an observation.
- 3062 • COMMANDED
- 3063 directive value including adjustments such as an offset or overrides.

- 3064 • PROGRAMMED
- 3065 directive value without offsets and adjustments.

### 3066 **5.3.20 Deceleration**

- 3067 negative rate of change of velocity.
- 3068 The units of Deceleration **MUST** be MILLIMETER/SECOND<sup>2</sup>.
- 3069 The default subType of Deceleration is ACTUAL.

#### 3070 **5.3.20.1 Subtypes of Deceleration**

- 3071 • ACTUAL
- 3072 measured or reported value of an observation.
- 3073 • COMMANDED
- 3074 directive value including adjustments such as an offset or overrides.
- 3075 • PROGRAMMED
- 3076 directive value without offsets and adjustments.

### 3077 **5.3.21 Density**

- 3078 volumetric mass of a material per unit volume of that material.
- 3079 The units of Density **MUST** be MILLIGRAM/CUBIC\_MILLIMETER.

### 3080 **5.3.22 DepositionAccelerationVolumetric**

- 3081 rate of change in spatial volume of material deposited in an additive manufacturing pro-
- 3082 cess.
- 3083 The units of DepositionAccelerationVolumetric **MUST** be CUBIC\_MILLIMETER/SECOND<sup>2</sup>

3084 **5.3.22.1 Subtypes of DepositionAccelerationVolumetric**

- 3085 • ACTUAL
- 3086 measured or reported value of an observation.
- 3087 • COMMANDED
- 3088 directive value including adjustments such as an offset or overrides.

3089 **5.3.23 DepositionDensity**

3090 density of the material deposited in an additive manufacturing process per unit of volume.  
3091 The units of DepositionDensity **MUST** be MILLIGRAM/CUBIC\_MILLIMETER.

3092 **5.3.23.1 Subtypes of DepositionDensity**

- 3093 • ACTUAL
- 3094 measured or reported value of an observation.
- 3095 • COMMANDED
- 3096 directive value including adjustments such as an offset or overrides.

3097 **5.3.24 DepositionMass**

3098 mass of the material deposited in an additive manufacturing process.  
3099 The units of DepositionMass **MUST** be MILLIGRAM.

3100 **5.3.24.1 Subtypes of DepositionMass**

- 3101 • ACTUAL
- 3102 measured or reported value of an observation.
- 3103 • COMMANDED
- 3104 directive value including adjustments such as an offset or overrides.

**3105 5.3.25 DepositionRateVolumetric**

3106 rate at which a spatial volume of material is deposited in an additive manufacturing pro-  
3107 cess.

3108 The units of `DepositionRateVolumetric` **MUST** be `CUBIC_MILLIMETER/SECOND`.

**3109 5.3.25.1 Subtypes of DepositionRateVolumetric**

3110 • ACTUAL

3111 measured or reported value of an observation.

3112 • COMMANDED

3113 directive value including adjustments such as an offset or overrides.

**3114 5.3.26 DepositionVolume**

3115 spatial volume of material to be deposited in an additive manufacturing process.

3116 The units of `DepositionVolume` **MUST** be `CUBIC_MILLIMETER`.

**3117 5.3.26.1 Subtypes of DepositionVolume**

3118 • ACTUAL

3119 measured or reported value of an observation.

3120 • COMMANDED

3121 directive value including adjustments such as an offset or overrides.

**3122 5.3.27 DewPoint**

3123 temperature at which moisture begins to condense, corresponding to saturation for a given  
3124 absolute humidity.

3125 The units of `DewPoint` **MUST** be `CELSIUS`.

### 3126 **5.3.28 Diameter**

3127 dimension of a diameter.

3128 The units of Diameter **MUST** be MILLIMETER.

### 3129 **5.3.29 DischargeRate**

3130 value of current being drawn from the Component.

3131 The units of DischargeRate **MUST** be AMPERE.

3132 The default subType of DischargeRate is ACTUAL.

#### 3133 **5.3.29.1 Subtypes of DischargeRate**

3134 • ACTUAL

3135 measured or reported value of an observation.

3136 • TARGET

3137 goal of the operation or process.

### 3138 **5.3.30 Displacement**

3139 change in position of an object.

3140 The units of Displacement **MUST** be MILLIMETER.

### 3141 **5.3.31 DisplacementAngular**

3142 absolute value of the change in angular position around a vector

3143 Note: The displacement vector **MAY** be defined by the motion of the owning  
3144 Component.

3145 The units of DisplacementAngular **MUST** be DEGREE.

### 3146 **5.3.32 DisplacementLinear**

3147 absolute value of the change in position along a vector.

3148 Note: The displacement vector **MAY** be defined by the motion of the owning  
3149 Component.

3150 The units of DisplacementLinear **MUST** be MILLIMETER.

### 3151 **5.3.33 ElectricalEnergy**

3152 Wattage used or generated by a component over an interval of time.

3153 The units of ElectricalEnergy **MUST** be WATT\_SECOND.

### 3154 **5.3.34 EquipmentTimer**

3155 amount of time a piece of equipment or a sub-part of a piece of equipment has performed  
3156 specific activities.

3157 The units of EquipmentTimer **MUST** be SECOND.

3158 A subType **MUST** always be specified.

#### 3159 **5.3.34.1 Subtypes of EquipmentTimer**

3160 • DELAY

3161 elapsed time of a temporary halt of action.

3162 • LOADED

3163 time that the sub-parts of a piece of equipment are under load.

3164 Example: For traditional machine tools, this is a measurement of the time that the  
3165 cutting tool is assumed to be engaged with the part.

- 3166 • OPERATING  
3167 time that the major sub-parts of a piece of equipment are powered or performing any  
3168 activity whether producing a part or product or not.  
3169 Example: For traditional machine tools, this includes WORKING, plus idle time.
- 3170 • POWERED  
3171 time that primary power is applied to the piece of equipment and, as a minimum, the  
3172 controller or logic portion of the piece of equipment is powered and functioning or  
3173 components that are required to remain on are powered.  
3174 Example: Heaters for an extrusion machine that are required to be powered even  
3175 when the equipment is turned off.
- 3176 • WORKING  
3177 time that a piece of equipment is performing any activity the equipment is active and  
3178 performing a function under load or not.  
3179 Example: For traditional machine tools, this includes LOADED, plus rapid moves,  
3180 tool changes, etc.

### 3181 5.3.35 FillLevel

3182 amount of a substance remaining compared to the planned maximum amount of that sub-  
3183 stance.

3184 The units of FillLevel **MUST** be PERCENT.

### 3185 5.3.36 Flow

3186 rate of flow of a fluid.

3187 The units of Flow **MUST** be LITER/SECOND.

### 3188 5.3.37 FollowingError

3189 difference between actual and commanded position at any specific point in time during a  
3190 motion.

3191 The units of FollowingError **MUST** be COUNT.



3192 **5.3.37.1 Subtypes of FollowingError**

- 3193 • ACTUAL
- 3194 measured or reported value of an observation.

3195 **5.3.38 FollowingErrorAngular**

3196 angular difference between the commanded encoder/resolver position and the actual en-  
3197 coder/resolver position at any specified point in time during a motion.

3198 The units of FollowingErrorAngular **MUST** be DEGREE.

3199 **5.3.38.1 Subtypes of FollowingErrorAngular**

- 3200 • ACTUAL
- 3201 measured or reported value of an observation.

3202 **5.3.39 FollowingErrorLinear**

3203 difference between the commanded encoder/resolver position and the actual encoder/re-  
3204 solver position at any specified point in time during a motion.

3205 The units of FollowingErrorLinear **MUST** be MILLIMETER.

3206 **5.3.39.1 Subtypes of FollowingErrorLinear**

- 3207 • ACTUAL
- 3208 measured or reported value of an observation.

3209 **5.3.40 Frequency**

3210 number of occurrences of a repeating event per unit time.

3211 The units of Frequency **MUST** be HERTZ.

3212 **5.3.41 <<deprecated>>GlobalPosition**

3213 position in three-dimensional space.

3214 **DEPRECATED** in Version 1.1.

3215 The units of `GlobalPosition` **MUST** be `MILLIMETER`.

3216 **5.3.41.1 Subtypes of GlobalPosition**

3217 • `ACTUAL`

3218 measured or reported value of an observation.

3219 • `COMMANDED`

3220 directive value including adjustments such as an offset or overrides.

3221 **5.3.42 GravitationalAcceleration**

3222 acceleration relative to Earth's gravity of 9.80665 `METER/SECOND`<sup>2</sup>.

3223 The units of `GravitationalAcceleration` **MUST** be `GRAVITATIONAL_ACCELERATION`.

3224 **5.3.43 GravitationalForce**

3225 force relative to earth's gravity.

3226 Note:  $Mass \times GravitationalAcceleration$

3227 The units of `GravitationalForce` **MUST** be `GRAVITATIONAL_FORCE`.

3228 **5.3.44 HumidityAbsolute**

3229 amount of water vapor expressed in grams per cubic meter.

3230 The units of `HumidityAbsolute` **MUST** be `GRAM/CUBIC_METER`.

3231 **5.3.44.1 Subtypes of HumidityAbsolute**

- 3232 • ACTUAL
- 3233 measured or reported value of an observation.
- 3234 • COMMANDED
- 3235 directive value including adjustments such as an offset or overrides.

3236 **5.3.45 HumidityRelative**

- 3237 amount of water vapor present expressed as a percent to reach saturation at the same tem-  
3238 perature.
- 3239 The units of HumidityRelative **MUST** be PERCENT.

3240 **5.3.45.1 Subtypes of HumidityRelative**

- 3241 • ACTUAL
- 3242 measured or reported value of an observation.
- 3243 • COMMANDED
- 3244 directive value including adjustments such as an offset or overrides.

3245 **5.3.46 HumiditySpecific**

- 3246 ratio of the water vapor present over the total weight of the water vapor and air present  
3247 expressed as a percent.
- 3248 The units of HumiditySpecific **MUST** be PERCENT.

3249 **5.3.46.1 Subtypes of HumiditySpecific**

- 3250 • ACTUAL
- 3251 measured or reported value of an observation.

- 3252 • COMMANDED
- 3253 directive value including adjustments such as an offset or overrides.

### 3254 **5.3.47 Length**

- 3255 length of an object.
- 3256 The units of Length **MUST** be MILLIMETER.

#### 3257 **5.3.47.1 Subtypes of Length**

- 3258 • REMAINING
- 3259 remaining total length of an object.
- 3260 • STANDARD
- 3261 standard or original length of an object.
- 3262 • USEABLE
- 3263 remaining usable length of an object.

### 3264 **5.3.48 <<deprecated>>Level**

- 3265 level of a resource.
- 3266 **DEPRECATED** in *Version 1.2*. See FILL\_LEVEL.
- 3267 The units of Level **MUST** be PERCENT.

### 3268 **5.3.49 LinearForce**

- 3269 *force* applied to a mass in one direction only.
- 3270 The units of LinearForce **MUST** be NEWTON.

**3271 5.3.50 Load**

3272 actual versus the standard rating of a piece of equipment.

3273 The units of Load **MUST** be PERCENT.

**3274 5.3.51 Mass**

3275 mass of an object(s) or an amount of material.

3276 The units of Mass **MUST** be KILOGRAM.

**3277 5.3.52 ObservationUpdateRate**

3278 average rate of change of values for data items in the MTConnect streams. The average is  
3279 computed over a rolling window defined by the implementation.

3280 The units of ObservationUpdateRate **MUST** be COUNT/SECOND.

**3281 5.3.53 Openness**

3282 percentage open where 100% is fully open and 0% is fully closed.

3283 The units of Openness **MUST** be PERCENT.

**3284 5.3.54 Orientation**

3285 angular position of a plane or vector relative to a *cartesian coordinate system*

3286 The units of Orientation **MUST** be DEGREE\_3D.

3287 The value of Orientation **MUST** be a list of float of size 0..3.

**3288 5.3.54.1 Subtypes of Orientation**

- 3289 • ACTUAL
- 3290 measured or reported value of an observation.
- 3291 • COMMANDED
- 3292 directive value including adjustments such as an offset or overrides.

### 3293 5.3.55 PH

- 3294 acidity or alkalinity of a solution.
- 3295 The units of PH **MUST** be PH.

### 3296 5.3.56 PathFeedrate

- 3297 feedrate for the axes, or a single axis, associated with a Path component.
- 3298 The units of PathFeedrate **MUST** be MILLIMETER/SECOND.

#### 3299 5.3.56.1 Subtypes of PathFeedrate

- 3300 • ACTUAL
- 3301 measured or reported value of an observation.
- 3302 • COMMANDED
- 3303 directive value including adjustments such as an offset or overrides.
- 3304 • JOG
- 3305 relating to momentary activation of a function or a movement.
- 3306 **DEPRECATION WARNING:** May be deprecated in the future.
- 3307 • OVERRIDE
- 3308 operator's overridden value.
- 3309 **DEPRECATED** in *Version 1.3*.
- 3310 • PROGRAMMED
- 3311 directive value without offsets and adjustments.

- 3312 • RAPID
- 3313 performing an operation faster or in less time than nominal rate.

### 3314 **5.3.57 PathFeedratePerRevolution**

3315 feedrate for the axes, or a single axis.

3316 The units of PathFeedratePerRevolution **MUST** be MILLIMETER/REVO-  
3317 LUTION.

#### 3318 **5.3.57.1 Subtypes of PathFeedratePerRevolution**

- 3319 • ACTUAL
- 3320 measured or reported value of an observation.
- 3321 • COMMANDED
- 3322 directive value including adjustments such as an offset or overrides.
- 3323 • PROGRAMMED
- 3324 directive value without offsets and adjustments.

### 3325 **5.3.58 PathPosition**

3326 position of a control point associated with a Controller or a Path.

3327 The units of PathPosition **MUST** be MILLIMETER\_3D.

3328 The value of PathPosition **MUST** be a list of float of size 0..3.

#### 3329 **5.3.58.1 Subtypes of PathPosition**

- 3330 • ACTUAL
- 3331 measured or reported value of an observation.
- 3332 • COMMANDED
- 3333 directive value including adjustments such as an offset or overrides.

- 3334 • PROBE
- 3335 position provided by a measurement probe.
- 3336 **DEPRECATION WARNING:** May be deprecated in the future.
- 3337 • TARGET
- 3338 goal of the operation or process.

### 3339 5.3.59 Position

- 3340 point along an axis in a *cartesian coordinate system*.
- 3341 The units of Position **MUST** be MILLIMETER.

#### 3342 5.3.59.1 Subtypes of Position

- 3343 • ACTUAL
- 3344 measured or reported value of an observation.
- 3345 • COMMANDED
- 3346 directive value including adjustments such as an offset or overrides.
- 3347 • PROGRAMMED
- 3348 directive value without offsets and adjustments.
- 3349 • TARGET
- 3350 goal of the operation or process.

### 3351 5.3.60 PositionCartesian

- 3352 point in a *cartesian coordinate system*.
- 3353 The units of PositionCartesian **MUST** be MILLIMETER\_3D.
- 3354 The value of PositionCartesian **MUST** be a list of float of size 0..3.



**3355 5.3.61 PowerFactor**

3356 ratio of real power flowing to a load to the apparent power in that AC circuit.

3357 The units of PowerFactor **MUST** be PERCENT.

**3358 5.3.62 Pressure**

3359 force per unit area measured relative to atmospheric pressure.

3360 Commonly referred to as gauge pressure.

3361 The units of Pressure **MUST** be PASCAL.

**3362 5.3.63 PressureAbsolute**

3363 The force per unit area measured relative to a vacuum.

3364 The units of PressureAbsolute **MUST** be PASCAL.

**3365 5.3.64 PressurizationRate**

3366 change of pressure per unit time.

3367 The units of PressurizationRate **MUST** be PASCAL/SECOND.

**3368 5.3.64.1 Subtypes of PressurizationRate**

3369 • ACTUAL

3370 measured or reported value of an observation.

3371 • COMMANDED

3372 directive value including adjustments such as an offset or overrides.

3373 • PROGRAMMED

3374 directive value without offsets and adjustments.

### 3375 **5.3.65 ProcessTimer**

3376 amount of time a piece of equipment has performed different types of activities associated  
3377 with the process being performed at that piece of equipment.

3378 The units of `ProcessTimer` **MUST** be `SECOND`.

3379 A `subType` **MUST** always be specified.

#### 3380 **5.3.65.1 Subtypes of ProcessTimer**

3381 • `DELAY`

3382 elapsed time of a temporary halt of action.

3383 • `PROCESS`

3384 time from the beginning of production of a part or product on a piece of equipment  
3385 until the time that production is complete for that part or product on that piece of  
3386 equipment.

3387 This includes the time that the piece of equipment is running, producing parts or  
3388 products, or in the process of producing parts.

### 3389 **5.3.66 Resistance**

3390 degree to which a substance opposes the passage of an electric current.

3391 The units of `Resistance` **MUST** be `OHM`.

### 3392 **5.3.67 RotaryVelocity**

3393 rotational speed of a rotary axis.

3394 The units of `RotaryVelocity` **MUST** be `REVOLUTION/MINUTE`.

#### 3395 **5.3.67.1 Subtypes of RotaryVelocity**

- 3396     • ACTUAL
- 3397        measured or reported value of an observation.
- 3398     • COMMANDED
- 3399        directive value including adjustments such as an offset or overrides.
- 3400     • OVERRIDE
- 3401        The operators overridden value.
- 3402     • PROGRAMMED
- 3403        directive value without offsets and adjustments.

### 3404 **5.3.68 SettlingError**

- 3405    difference between actual and commanded position at the end of a motion.
- 3406    The units of `SettlingError` **MUST** be COUNT.

#### 3407 **5.3.68.1 Subtypes of SettlingError**

- 3408     • ACTUAL
- 3409        measured or reported value of an observation.

### 3410 **5.3.69 SettlingErrorAngular**

- 3411    angular difference between the commanded encoder/resolver position, and the actual en-
- 3412    coder/resolver position when motion is complete.
- 3413    The units of `SettlingErrorAngular` **MUST** be DEGREE.

#### 3414 **5.3.69.1 Subtypes of SettlingErrorAngular**

- 3415     • ACTUAL
- 3416        measured or reported value of an observation.

**3417 5.3.70 SettlingErrorLinear**

3418 difference between the commanded encoder/resolver position, and the actual encoder/re-  
3419 solver position when motion is complete.

3420 The units of `SettlingErrorLinear` **MUST** be `MILLIMETER`.

**3421 5.3.70.1 Subtypes of SettlingErrorLinear**

- 3422 • `ACTUAL`
- 3423 measured or reported value of an observation.

**3424 5.3.71 SoundLevel**

3425 sound level or sound pressure level relative to atmospheric pressure.

3426 The units of `SoundLevel` **MUST** be `DECIBEL`.

3427 The default `subType` of `SoundLevel` is `A_SCALE`.

**3428 5.3.71.1 Subtypes of SoundLevel**

- 3429 • `A_SCALE`
- 3430 A Scale weighting factor. This is the default weighting factor if no factor is specified
- 3431 • `B_SCALE`
- 3432 B Scale weighting factor
- 3433 • `C_SCALE`
- 3434 C Scale weighting factor
- 3435 • `D_SCALE`
- 3436 D Scale weighting factor
- 3437 • `NO_SCALE`
- 3438 No weighting factor on the frequency scale

3439 **5.3.72 <<deprecated>>SpindleSpeed**

3440 rotational speed of the rotary axis.

3441 **DEPRECATED** in *Version 1.2*. Replaced by ROTARY\_VELOCITY.

3442 The units of SpindleSpeed **MUST** be REVOLUTION/MINUTE.

3443 **5.3.72.1 Subtypes of SpindleSpeed**

3444 • ACTUAL

3445 measured or reported value of an observation.

3446 **DEPRECATED** in *Version 1.3*.

3447 • COMMANDED

3448 directive value including adjustments such as an offset or overrides.

3449 **DEPRECATED** in *Version 1.3*.

3450 • OVERRIDE

3451 operator's overridden value.

3452 **DEPRECATED** in *Version 1.3*.

3453 **5.3.73 Strain**

3454 amount of deformation per unit length of an object when a load is applied.

3455 The units of Strain **MUST** be PERCENT.

3456 **5.3.74 Temperature**

3457 degree of hotness or coldness measured on a definite scale.

3458 The units of Temperature **MUST** be CELSIUS.

**3459 5.3.75 Tension**

3460 force that stretches or elongates an object.

3461 The units of Tension **MUST** be NEWTON.

**3462 5.3.76 Tilt**

3463 angular displacement.

3464 The units of Tilt **MUST** be MICRO\_RADIAN.

**3465 5.3.77 Torque**

3466 turning force exerted on an object or by an object.

3467 The units of Torque **MUST** be NEWTON\_METER.

**3468 5.3.78 Velocity**

3469 rate of change of position of a Component.

3470 The units of Velocity **MUST** be MILLIMETER/SECOND.

**3471 5.3.79 Viscosity**

3472 fluid's resistance to flow.

3473 The units of Viscosity **MUST** be PASCAL\_SECOND.

**3474 5.3.80 VoltAmpere**

3475 apparent power in an electrical circuit, equal to the product of root-mean-square (RMS)

3476 voltage and RMS current (commonly referred to as VA).

3477 The units of VoltAmpere **MUST** be VOLT\_AMPERE.

### 3478 **5.3.81 VoltAmpereReactive**

3479 reactive power in an AC electrical circuit (commonly referred to as VAR).

3480 The units of VoltAmpereReactive **MUST** be VOLT\_AMPERE\_REACTIVE.

### 3481 **5.3.82 <<deprecated>>Voltage**

3482 electrical potential between two points.

3483 **DEPRECATED** in *Version 1.6*. Replaced by VOLTAGE\_AC and VOLTAGE\_DC.

3484 The units of Voltage **MUST** be VOLT.

#### 3485 **5.3.82.1 Subtypes of Voltage**

3486 • ACTUAL

3487 measured or reported value of an observation.

3488 **DEPRECATED** in *Version 1.6*.

3489 • ALTERNATING

3490 alternating voltage or current.

3491 If not specified further in statistic, defaults to RMS voltage.

3492 **DEPRECATED** in *Version 1.6*.

3493 • DIRECT

3494 DC current or voltage.

3495 **DEPRECATED** in *Version 1.6*.

3496 • TARGET

3497 goal of the operation or process.

3498 **DEPRECATED** in *Version 1.6*.

**3499 5.3.83 VoltageAC**

3500 electrical potential between two points in an electrical circuit in which the current period-  
3501 ically reverses direction.

3502 The units of VoltageAC **MUST** be VOLT.

3503 A subType **MUST** always be specified.

**3504 5.3.83.1 Subtypes of VoltageAC**

- 3505 • ACTUAL  
3506 measured or reported value of an observation.
- 3507 • COMMANDED  
3508 directive value including adjustments such as an offset or overrides.
- 3509 • PROGRAMMED  
3510 directive value without offsets and adjustments.

**3511 5.3.84 VoltageDC**

3512 electrical potential between two points in an electrical circuit in which the current is uni-  
3513 directional.

3514 The units of VoltageDC **MUST** be VOLT.

3515 A subType **MUST** always be specified.

**3516 5.3.84.1 Subtypes of VoltageDC**

- 3517 • ACTUAL  
3518 measured or reported value of an observation.
- 3519 • COMMANDED  
3520 directive value including adjustments such as an offset or overrides.
- 3521 • PROGRAMMED  
3522 directive value without offsets and adjustments.



**3523 5.3.85 VolumeFluid**

3524 fluid volume of an object or container.

3525 The units of `VolumeFluid` **MUST** be `MILLILITER`.

**3526 5.3.85.1 Subtypes of VolumeFluid**

3527 • ACTUAL

3528 measured or reported value of an observation.

3529 • CONSUMED

3530 reported or measured value of the amount used in the manufacturing process.

3531 • ENDED

3532 boundary when an activity or an event terminates.

3533 • PART

3534 reported or measured value of amount included in the *part*.

3535 • START

3536 boundary when an activity or an event commences.

3537 • WASTE

3538 reported or measured value of the amount discarded.

**3539 5.3.86 VolumeSpatial**

3540 geometric volume of an object or container.

3541 The units of `VolumeSpatial` **MUST** be `CUBIC_MILLIMETER`.

**3542 5.3.86.1 Subtypes of VolumeSpatial**

3543 • ACTUAL

3544 measured or reported value of an observation.

- 3545 • CONSUMED
- 3546 reported or measured value of the amount used in the manufacturing process.
- 3547 • ENDED
- 3548 boundary when an activity or an event terminates.
- 3549 • PART
- 3550 reported or measured value of amount included in the *part*.
- 3551 • START
- 3552 boundary when an activity or an event commences.
- 3553 • WASTE
- 3554 reported or measured value of the amount discarded

### 3555 **5.3.87 Wattage**

- 3556 power flowing through or dissipated by an electrical circuit or piece of equipment.
- 3557 The units of Wattage **MUST** be WATT.

#### 3558 **5.3.87.1 Subtypes of Wattage**

- 3559 • ACTUAL
- 3560 measured or reported value of an observation.
- 3561 • TARGET
- 3562 goal of the operation or process.

### 3563 **5.3.88 XDimension**

- 3564 dimension of an entity relative to the X direction of the referenced coordinate system.
- 3565 The units of XDimension **MUST** be MILLIMETER.

**3566 5.3.89 YDimension**

3567 dimension of an entity relative to the Y direction of the referenced coordinate system.

3568 The units of YDimension **MUST** be MILLIMETER.

**3569 5.3.90 ZDimension**

3570 dimension of an entity relative to the Z direction of the referenced coordinate system.

3571 The units of ZDimension **MUST** be MILLIMETER.

## 3572 6 Profile

3573 MTConnect Profile is a *profile* that extends the Systems Modeling Language (SysML)  
 3574 metamodel for the MTConnect domain using additional data types and *stereotypes*.

### 3575 6.1 DataTypes

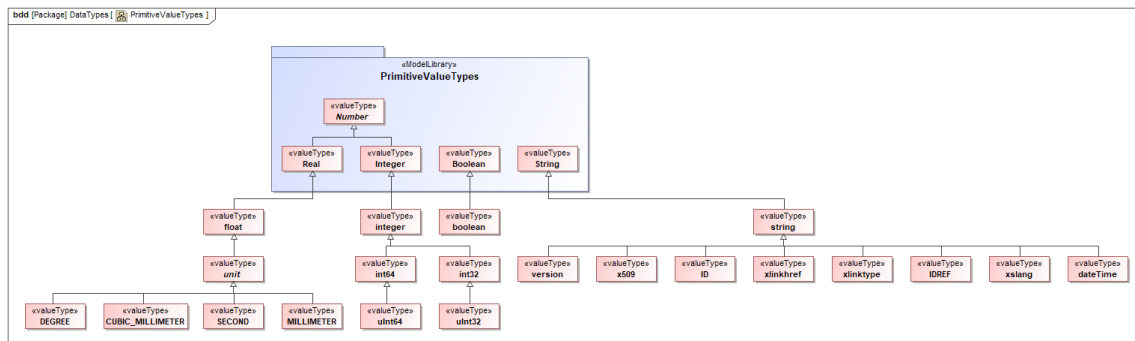


Figure 13: DataTypes

#### 3576 6.1.1 boolean

3577 primitive type.

#### 3578 6.1.2 ID

3579 string that represents an identifier (ID).

#### 3580 6.1.3 string

3581 primitive type.

#### 3582 6.1.4 float

3583 primitive type.

**3584 6.1.5 datetime**

3585 string that represents timestamp in ISO 8601 format.

**3586 6.1.6 integer**

3587 primitive type.

**3588 6.1.7 xlinktype**

3589 string that represents the type of an XLink element. See <https://www.w3.org/TR/xlink11/>.

**3591 6.1.8 xslang**

3592 string that represents a language tag. See <http://www.ietf.org/rfc/rfc4646.txt>.

**3594 6.1.9 SECOND**

3595 float that represents time in seconds.

**3596 6.1.10 IDREF**

3597 string that represents a reference to an ID.

**3598 6.1.11 xlinkhref**

3599 string that represents the locator attribute of an XLink element. See <https://www.w3.org/TR/xlink11/>.

**3601 6.1.12 x509**

3602 string that represents an x509 data block. *Ref ISO/IEC 9594-8:2020.*

**3603 6.1.13 int32**

3604 32-bit integer.

**3605 6.1.14 int64**

3606 64-bit integer.

**3607 6.1.15 version**

3608 series of four numeric values, separated by a decimal point, representing a *major*, *minor*,  
3609 and *revision* number of the MTConnect Standard and the revision number of a specific  
3610 *schema*.

**3611 6.1.16 uint32**

3612 32-bit unsigned integer.

**3613 6.1.17 uint64**

3614 64-bit unsigned integer.

**3615 6.1.18 binary**

3616 base-2 numeral system or binary numeral system represented by two digits: “0” and “1”.

**3617 6.1.19 double**

3618 primitive type.

**3619 6.2 Stereotypes**

**3620 6.2.1 organizer**

3621 element that *organizes* other elements of a type.

**3622 6.2.2 deprecated**

3623 element that has been deprecated.

**3624 6.2.3 extensible**

3625 enumeration that can be extended.

**3626 6.2.4 informative**

3627 element that is descriptive and non-normative.

**3628 6.2.5 valueType**

3629 extends SysML <<ValueType>> to include `Class` as a value type.

**3630 6.2.6 normative**

3631 element that has been added to the standard.

**3632 6.2.7 observes**

3633 association in which a *Component* makes *Observations* about an observable *DataItem*.



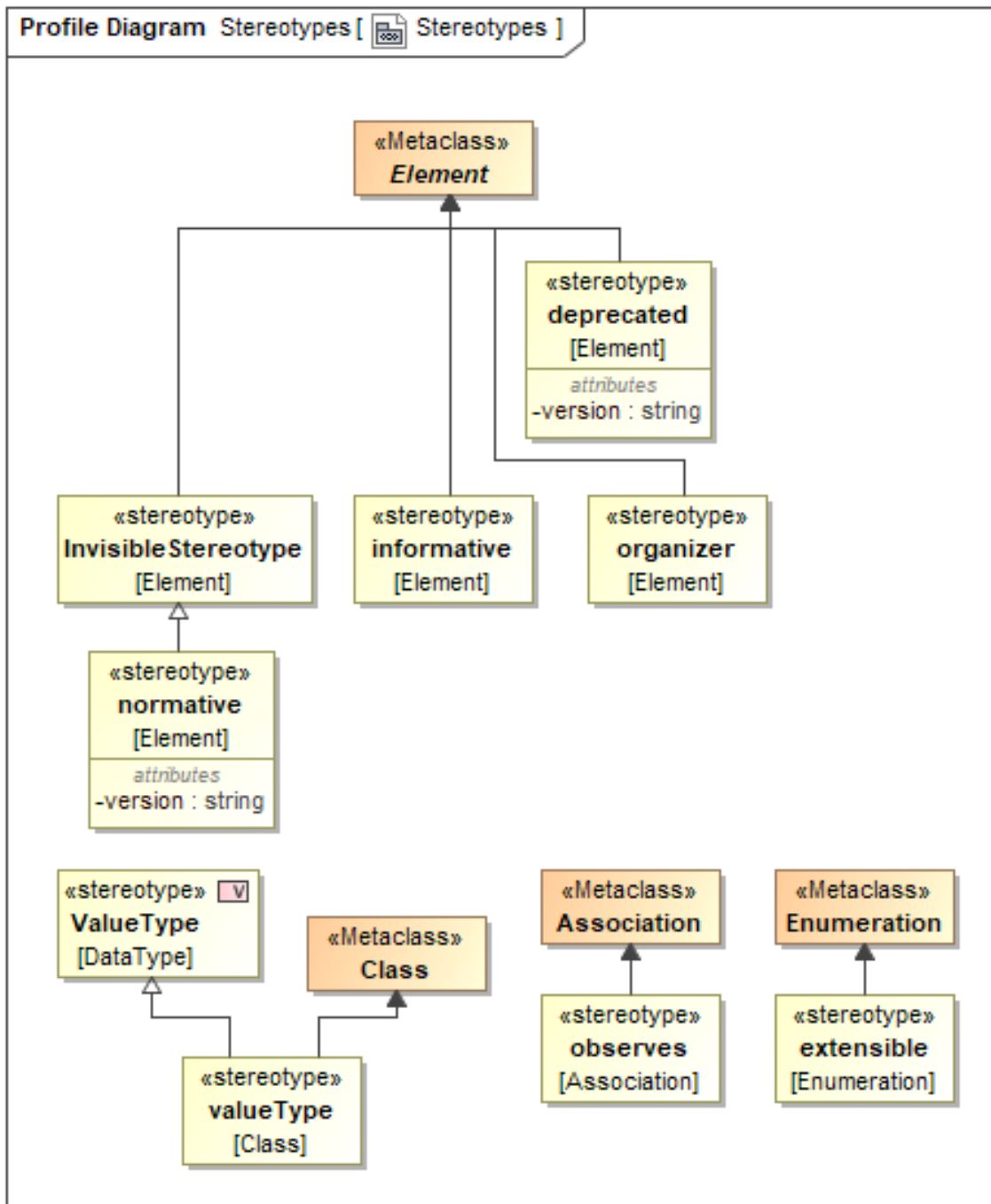


Figure 14: Stereotypes

## 3634 Appendices

### 3635 A Bibliography

- 3636 Engineering Industries Association. EIA Standard - EIA-274-D, Interchangeable Variable,  
3637 Block Data Format for Positioning, Contouring, and Contouring/Positioning Numerically  
3638 Controlled Machines. Washington, D.C. 1979.
- 3639 ISO TC 184/SC4/WG3 N1089. ISO/DIS 10303-238: Industrial automation systems and  
3640 integration Product data representation and exchange Part 238: Application Protocols: Ap-  
3641 plication interpreted model for computerized numerical controllers. Geneva, Switzerland,  
3642 2004.
- 3643 International Organization for Standardization. ISO 14649: Industrial automation sys-  
3644 tems and integration – Physical device control – Data model for computerized numerical  
3645 controllers – Part 10: General process data. Geneva, Switzerland, 2004.
- 3646 International Organization for Standardization. ISO 14649: Industrial automation sys-  
3647 tems and integration – Physical device control – Data model for computerized numerical  
3648 controllers – Part 11: Process data for milling. Geneva, Switzerland, 2000.
- 3649 International Organization for Standardization. ISO 6983/1 – Numerical Control of ma-  
3650 chines – Program format and definition of address words – Part 1: Data format for posi-  
3651 tioning, line and contouring control systems. Geneva, Switzerland, 1982.
- 3652 Electronic Industries Association. ANSI/EIA-494-B-1992, 32 Bit Binary CL (BCL) and  
3653 7 Bit ASCII CL (ACL) Exchange Input Format for Numerically Controlled Machines.  
3654 Washington, D.C. 1992.
- 3655 National Aerospace Standard. Uniform Cutting Tests - NAS Series: Metal Cutting Equip-  
3656 ment Specifications. Washington, D.C. 1969.
- 3657 International Organization for Standardization. ISO 10303-11: 1994, Industrial automa-  
3658 tion systems and integration Product data representation and exchange Part 11: Descrip-  
3659 tion methods: The EXPRESS language reference manual. Geneva, Switzerland, 1994.
- 3660 International Organization for Standardization. ISO 10303-21: 1996, Industrial automa-  
3661 tion systems and integration – Product data representation and exchange – Part 21: Imple-  
3662 mentation methods: Clear text encoding of the exchange structure. Geneva, Switzerland,  
3663 1996.
- 3664 H.L. Horton, F.D. Jones, and E. Oberg. Machinery's Handbook. Industrial Press, Inc.

- 3665 New York, 1984.
- 3666 International Organization for Standardization. ISO 841-2001: Industrial automation systems and integration - Numerical control of machines - Coordinate systems and motion  
3667 nomenclature. Geneva, Switzerland, 2001.  
3668
- 3669 ASME B5.57: Methods for Performance Evaluation of Computer Numerically Controlled  
3670 Lathes and Turning Centers, 1998.
- 3671 ASME/ANSI B5.54: Methods for Performance Evaluation of Computer Numerically Controlled Machining Centers. 2005.  
3672
- 3673 OPC Foundation. OPC Unified Architecture Specification, Part 1: Concepts Version 1.00.  
3674 July 28, 2006.
- 3675 IEEE STD 1451.0-2007, Standard for a Smart Transducer Interface for Sensors and Actuators – Common Functions, Communication Protocols, and Transducer Electronic Data  
3676 Sheet (TEDS) Formats, IEEE Instrumentation and Measurement Society, TC-9, The Institute of Electrical and Electronics Engineers, Inc., New York, N.Y. 10016, SH99684,  
3677  
3678 October 5, 2007.  
3679
- 3680 IEEE STD 1451.4-1994, Standard for a Smart Transducer Interface for Sensors and Actuators – Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet  
3681 (TEDS) Formats, IEEE Instrumentation and Measurement Society, TC-9, The Institute of  
3682  
3683 Electrical and Electronics Engineers, Inc., New York, N.Y. 10016, SH95225, December  
3684 15, 2004.

## 3685 B XML Schema Diagrams

### 3686 B.1 Observations Schema Diagrams

3687 See `Streams` element in `MTConnectStreams` schema.

3688 See `DeviceStream` element in `MTConnectStreams` schema.

3689 See `ComponentStream` element in `MTConnectStreams` schema.

3690 See `Sample` element in `MTConnectStreams` schema.

3691 See `Event` element in `MTConnectStreams` schema.

3692 See `Condition` element in `MTConnectStreams` schema.

### 3693 B.2 Representation Schema Diagrams

3694 See `AbsTimeSeries` element in `MTConnectStreams` schema.

3695 See `PartCountDiscrete` element in `MTConnectStreams` schema.

3696 See `VariableDataSet` element in `MTConnectStreams` schema.

3697 See `Entry` element in `MTConnectStreams` schema.

3698 See `WorkOffsetTable` element in `MTConnectStreams` schema.

## 3699 C XML Examples

### 3700 C.1 DeviceStream Example

#### Example 1: DeviceStream Example

```

3701 1 <Streams>
3702 2   <DeviceStream name="M12346" uuid="M8010W4194N">
3703 3     <ComponentStream component="Device" name="M12346" componentId="
3704     dl">
3705 4       <Events>
3706 5         <Availability dataItemId="avail" sequence="156" timestamp="
3707         2021-10-01T14:26:38.668505Z">AVAILABLE</Availability>

```

```

3708 6      <AssetChanged assetType="CuttingTool" dataItemId="d1\
3709          textunderscore_asset\textunderscore_chg" sequence="75570"
3710          timestamp="2021-10-07T05:08:53.870206Z">M8010W4194N1
3711          .120</AssetChanged>
3712 7      <AssetRemoved assetType="CuttingTool" dataItemId="d1\
3713          textunderscore_asset\textunderscore_rem" sequence="140"
3714          timestamp="2021-10-01T11:40:08.101461Z">UNAVAILABLE</
3715          AssetRemoved>
3716 8      </Events>
3717 9      </ComponentStream>
3718 10     <ComponentStream component="Controller" name="controller"
3719          componentId="cont">
3720 11     <Events>
3721 12         <EmergencyStop dataItemId="estop" sequence="159" timestamp="
3722 2021-10-01T14:26:38.66869Z">ARMED</EmergencyStop>
3723 13     </Events>
3724 14     <Samples>
3725 15         <AccumulatedTime dataItemId="cut\textunderscore_time"
3726 1763070.0</AccumulatedTime>
3727 1763070.0</AccumulatedTime>
3728 16     </Samples>
3729 17     <Condition>
3730 18         <Unavailable dataItemId="cont\textunderscore_system"
3731 72" timestamp="2021-10-11T21:04:03.251999Z" type="
3732 SYSTEM"/>
3733 19         <Warning dataItemId="cont\textunderscore_system" nativecode=
3734 313" nativeSeverity="50" sequence="75573" timestamp="
3735 2021-10-07T05:08:58.518317Z" type="LOGIC\textunderscore_
3736 PROGRAM">PALLET ARM DOWN RS. MALEF.</Warning>
3737 20     </Condition>
3738 21     </ComponentStream>
3739 22     <ComponentStream component="Path" name="path" componentId="path1
3740     ">
3741 23     <Events>
3742 24         <Execution dataItemId="execution" name="execution" sequence=
3743 222623" timestamp="2021-10-12T06:04:32.761198Z">INTERRUPTED</
3744 Execution>
3745 25         <VariableDataSet count="2" dataItemId="cvars" sequence="
3746 126513" timestamp="2021-10-12T03:57:31.106559Z">
3747 26         <Entry key="100">66.3314</Entry>
3748 27         <Entry key="101">167.2</Entry>
3749 28         </VariableDataSet>
3750 29         <WorkOffsetTable count="2" dataItemId="woffset" sequence="
3751 222101" timestamp="2021-10-12T06:04:11.990531Z">
3752 30         <Entry key="G53.1">
3753 31         <Cell key="X">1</Cell>
3754 32         <Cell key="Y">2</Cell>
3755 33         <Cell key="Z">3</Cell>
3756 34         </Entry>
3757 35         <Entry key="G53.2">
3758 36         <Cell key="X">4</Cell>

```

```

3759 37      <Cell key="Y">5</Cell>
3760 38      <Cell key="Z">6</Cell>
3761 39      </Entry>
3762 40      </WorkOffsetTable>
3763 41      </Events>
3764 42      <Samples>
3765 43          <CuttingSpeed dataItemId="cspeed" sequence="112" timestamp="
3766 2021-10-07T05:08:28.221704Z" subType="ACTUAL">UNAVAILABLE</
3767 CuttingSpeed>
3768 44      </Samples>
3769 45      <Condition>
3770 46          <Normal dataItemId="path\textunderscore_system" sequence="
3771 153" timestamp="2021-10-11T21:04:03.262845Z" type="SYSTEM"/>
3772 47      </Condition>
3773 48      </ComponentStream>
3774 49      </DeviceStream>
3775 50      </Streams>

```

## 3776 C.2 Observations made for DataItem Example

### Example 2: MTConnectDevices Response Document

```

3777 1 <Components>
3778 2   <Controller id="cont" name="controller">
3779 3     <DataItems>
3780 4       <DataItem category="EVENT" id="estop" name="estop" type="
3781 EMERGENCY\textunderscore_STOP"/>
3782 5     </DataItems>
3783 6   </Controller>
3784 7 </Components>

```

### Example 3: MTConnectStreams Response Document

```

3785 1 <ComponentStream component="Controller" name="controller"
3786   componentId="cont">
3787 2   <Events>
3788 3     <EmergencyStop dataItemId="estop" sequence="159" timestamp="
3789 2021-10-01T14:26:38.66869Z">ARMED</EmergencyStop>
3790 4   </Events>
3791 5 </ComponentStream>

```

## 3792 C.3 Sample Example

### Example 4: Sample Example

```

3793 1 <Samples>

```

```

3794 2 <AccumulatedTime dataItemId="cut\textunderscore_time" sequence="
3795 75437" timestamp="2021-10-07T05:08:28.221704Z">1763070.0</
3796 AccumulatedTime>
3797 3 <CuttingSpeed dataItemId="cspeed" sequence="112" timestamp="
3798 2021-10-07T05:08:28.221704Z" subType="ACTUAL">UNAVAILABLE</
3799 CuttingSpeed>
3800 4 </Samples>

```

## 3801 C.4 Event Example

### Example 5: Event Example

```

3802 1 <Events>
3803 2 <Availability dataItemId="avail" sequence="156" timestamp="
3804 2021-10-01T14:26:38.668505Z">AVAILABLE</Availability>
3805 3 <AssetRemoved assetType="CuttingTool" dataItemId="d1\
3806 textunderscore_asset\textunderscore_rem" sequence="140"
3807 timestamp="2021-10-01T11:40:08.101461Z">UNAVAILABLE</
3808 AssetRemoved>
3809 4 </Events>

```

## 3810 C.5 Condition Example

3811 Condition types are represented differently in XML when compared to Sample and  
3812 Event types. The element name is the condition state of the Condition type in Pascal  
3813 Case. The name of the Condition type is represented by the attribute type.

3814 If the condition state is unavailable then the element name is represented by Unavail-  
3815 able.

### Example 6: Condition Example

```

3816 1 <Condition>
3817 2 <Unavailable dataItemId="cont\textunderscore_system" sequence="72"
3818 timestamp="2021-10-11T21:04:03.251999Z" type="SYSTEM"/>
3819 3 <Normal dataItemId="path\textunderscore_system" sequence="153"
3820 timestamp="2021-10-11T21:04:03.262845Z" type="SYSTEM"/>
3821 4 <Warning dataItemId="cont\textunderscore_system" nativecode="313"
3822 nativeSeverity="50" sequence="75573" timestamp="2021-10-07T05
3823 :08:58.518317Z" type="LOGIC\textunderscore_PROGRAM">PALLET ARM
3824 DOWN RS. MALF.</Warning>
3825 5 </Condition>

```

**3826 C.6 DataSet Example****Example 7: DataSet Example**

```

3827 1 <Events>
3828 2   <VariableDataSet count="2" dataItemId="cvars" sequence="126513"
3829   timestamp="2021-10-12T03:57:31.106559Z">
3830 3     <Entry key="100">66.3314</Entry>
3831 4     <Entry key="101">167.2</Entry>
3832 5   </VariableDataSet>
3833 6 </Events>

```

**3834 C.7 Table Example****Example 8: Table Example**

```

3835 1 <Events>
3836 2   <WorkOffsetTable count="2" dataItemId="woffset" sequence="222101"
3837   timestamp="2021-10-12T06:04:11.990531Z">
3838 3     <Entry key="G53.1">
3839 4       <Cell key="X">1</Cell>
3840 5       <Cell key="Y">2</Cell>
3841 6       <Cell key="Z">3</Cell>
3842 7     </Entry>
3843 8     <Entry key="G53.2">
3844 9       <Cell key="X">4</Cell>
3845 10      <Cell key="Y">5</Cell>
3846 11      <Cell key="Z">6</Cell>
3847 12    </Entry>
3848 13  </WorkOffsetTable>
3849 14 </Events>

```