Logo_FORCAM2010_200

FORCAM FORCE®

MTConnect

Document: FF\_MTConnect.docx

Created: 16.06.2015

Last changed: 17.09.15

Author: Peter Mueller

|  |  |
| --- | --- |
| D:\FORCAM\1_Produktmanagement\Release 5.0\Screenshots\FF_Cubes_RGB_RZ.gif | COPYRIGHT 2015 BY **FORCAM GMBH**  D-88214 Ravensburg  ALL RIGHTS RESERVED. COPY OR TRANSLATION, ALSO IN EXTRACTS ONLY WITH WRITTEN PERMISSION BY FORCAM GMBH |

Table of Contents

[1 Introduction 3](#_Toc431894826)

[1.1 Infrastructure 3](#_Toc431894827)

[2 Requirements 4](#_Toc431894828)

[2.1 Machine control 4](#_Toc431894829)

[2.2 Network 4](#_Toc431894830)

[2.3 Requirements on machine side 4](#_Toc431894831)

[2.4 Personnel capacities and know how 4](#_Toc431894832)

[2.5 Requirements overview 4](#_Toc431894833)

[3 Functionality 6](#_Toc431894834)

[3.1 Read in digital input signals 6](#_Toc431894835)

[3.2 Write digital output signals 6](#_Toc431894836)

[4 Software Configuration 7](#_Toc431894837)

[5 Factory Framework configuration 8](#_Toc431894838)

[5.1 Javis DCU configuration 8](#_Toc431894839)

[5.2 Signal configuration 9](#_Toc431894840)

[5.3 DACQ script example 10](#_Toc431894841)

[6 XML file structure 13](#_Toc431894842)

[6.1 Devices and Components 13](#_Toc431894843)

[6.2 Data Item 14](#_Toc431894844)

[6.3 Data Item Category 14](#_Toc431894845)

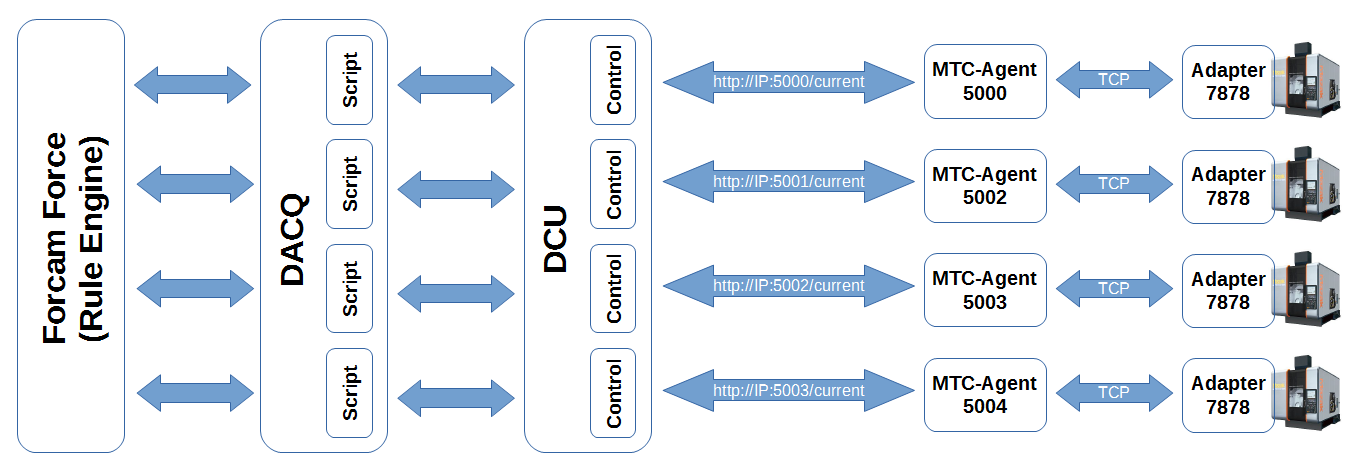
# Introduction

MTConnect is an industry standard for the production to get easier process information’s from numerically controlled machines. MTConnect is an open and extensible protocol for exchanging data between machine control systems and software applications like MES systems. In the current form, MTConnect is only able to read data from a machine control, not to write data to a control device.

The data are exchanged with MTConnect via XML files. The data transport is done by the HTTP protocol.

To retrieve data from an MTConnect agent, no session-oriented connection, and no registration or logoff sequence is required.

## Infrastructure



# Requirements

## Machine control

To connect a machine control the machine must be ready for MTConnect. The MTConnect adapter and agent should be installed on the controller.

This agent reads the data over port 7878 from the MTConnect adapter and generates an XML file that contains the recent machine data’s. The agent itself is accessible via port 5000 (or 5002 or 5003).

Note: Port 5001 is already in use by another application on the Forcam FORCE server!

Alternatively, the MTConnect agent can also be installed on the Factory Framework application server or another computer.

## Network

* Ethernet connection to the machine
* Customer specified IP address
* Free IP routing for TCP port 5000 (or 5002 or 5003) between Forcam DCU Server to the MTConnect agent.
* Routing for Telnet Port: 7878 in case, if the Agent is installed on a remote PC.

Note: MTConnect has no security features. Thus, when using MTConnect, the LAN must be well secured.

## Requirements on machine side

* Machine control which supports MTConnect
* Installed MTConnect adapter on the machine control, or on another dedicated computer in the network (HMI with Windows 95, open Telnet port: 7878 on the machine side will be required)
* Installed MTConnect agent on the machine control or on another computer in the network

## Personnel capacities and know how

* Install the MTConnect agent on the machine control or another computer by the customer (together with the machine manufacturer).
* Selection of machine signals holding the necessary information for the production state of the machine

## Requirements overview

* IP-Address
* Network connectivity
* Installed MTConnect adapter (Mazak)on the controller
* Data on Telnet Port 7878 between controller and remote PC
* MTC MTConnect agent receives the machine data
* Output via HTTP protocol (Internet)
* Client software collects and evaluates data

# Functionality

## Read in digital input signals

Signals can be read out of the XML-file which is generated by the MTConnect agent. An example address in the Internet is: <http://agent.mtconnect.org:80/>.



With the following address, you can see the recent information’s from the machine: <http://agent.mtconnect.org:80/current>:



## Write digital output signals

Actually it is not possible to write data over MTConnect into a machine control.

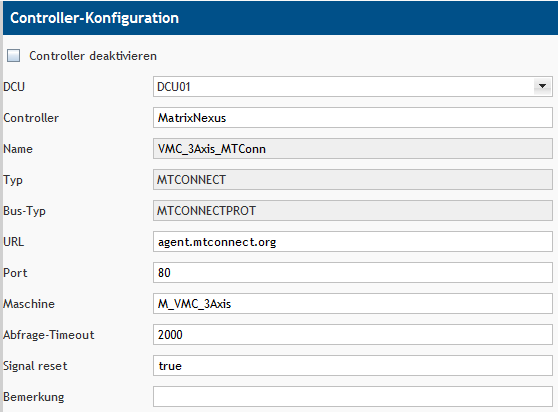
# Software Configuration

The controller must support MTConnect, it means on the control must be installed the MTConnect agent and if it’s possible also the MTConnect adapter.

The customer must specify which information’s are required from the machine for the MES system.

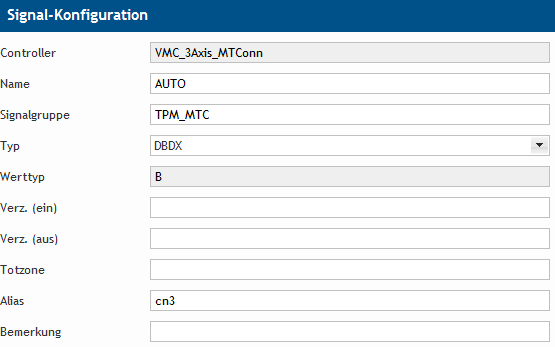
# Factory Framework configuration

## Javis DCU configuration



|  |  |  |
| --- | --- | --- |
| Input field | Description | Input values |
| **DCU** | Name of the DCU to which a controller belong to | Automatically filled in |
|  |  |  |
| **Controller** | Unique controller name | One word, digits are allowed without special characters |
|  |  |  |
| **Type** | MTCONNECT Plugin over HTTP | Input by selection |
|  |  |  |
| **Bus type** | MTCONNECTPROT bus type | Input by selection |
|  |  |  |
| **URL** | DNS-Name or IP-Address to the MTConnect agent | Several words or numbers, separated by a point |
|  |  |  |
| **Port** | Normally 5000, or 5002 or 5003 (5001 is already in use by another application) | Only numbers 5000-65535 |
|  |  |  |
| **Machine** | Name for the machine. This name must be identical to the machine name in the MTConnect XML file (**DeviceStream name**="[Machine name]") | One word without special characters |
|  |  |  |
| **Request timeout** | Timeout for data request | Only seconds without decimal |
|  |  |  |
| **Signal reset** | Reset the signals form control at every time | Boolean (true/false) |
|  |  |  |
| **Comment** | It is suggested to describe the connected machine, but can be left empty | Variable text |

## Signal configuration



|  |  |  |
| --- | --- | --- |
| Input field | Description | Input values |
| **Name** | Controller global unique signal name. | One word, digits are allowed without special characters |
|  |  |  |
| **Signal group** | Can be used to group signals, for example if more than one machine is connect with one controller, signals in different groups can have the same name | One word, digits are allowed without special characters |
|  |  |  |
| **Type** | Type of addressed value |  |
|  | Data block bit | DBDX |
|  | Data block double word | DBDW2 |
|  | Data block floating point decimal | DBREAL |
|  | Data block string | DBSTRING |
|  |  |  |
| **Value type** | Type of data |  |
|  | Boolean | B |
|  | Number | N |
|  | String | S |
|  |  |  |
| **Delay** | Delay a signal that switches to on until it is shown on (only **Number** and **String**) | Time in seconds |
|  |  |  |
| **Delay (on)** | Delay a signal that switches to on until it is shown on (only **Boolean**) | Time in seconds |
|  |  |  |
| **Delay (off)** | Delay a signal switches to off until it is shown off (only **Boolean**) | Time in seconds |
|  |  |  |
| **Dead band** | A signal change is not displayed, if the change is within (in this field) specified range (only Number) | The unit depends on the value to be measured |
|  |  |  |
| **Alias** | Alias name for the signal. This alias name must be identical to the signal name in the MTConnect XML File (**dataItemId**="[Signal name]") | One word without special characters |
|  |  |  |
| **Comment** | It is suggested to describe the signal, but can be left empty | Variable text |

## DACQ script example

|  |
| --- |
| //  // Task: Send machine state / strokes / quantities to runtime  // Created: 2015-07-20  // Version: 1.0  // Author: Daniel.Rinker@forcam.com  //  // --------------------------------------------------------------  //  // Incoming signals from DCU  // All variables must be checked if available on MTConnect machine interface  // M\_ON = machine on if "ON"  // M\_MODE = machine in automatic mode if "AUTOMATIC"  // M\_FEED\_OVR = machine feed override  // M\_FEED\_RATE = machine feed rate  // M\_SPINDLE\_SPEED = machine spindle speed  // ABS\_CNT1 = absolute counter 1 on PLC  //  // Outgoing information to rule engine  // state = machine state  // malfunction = malfunction reason  // counterSEND = machine strokes / quantity  //  // --------------------------------------------------------------  var\_local  begin  // GENERAL LOGIC VARIABLES  seconds: number;  // SCRIPT INIZIALIZING VARIABLES  initialized: boolean;  // PIECE COUNT VARIABLES  counter: number;  counterOLD: number;  counterSend: number;  i: number;  zaehlerAbs: number;  zaehlerAbsAlt: number;  FEED: number;  stat: string;  stat2: string;  oldstat: string;  paketnr: number;  palnr: number;  end;  //----- START CONFIGURE ----------------------------------------------------  begin  // INITIALIZE SCRIPT VARIABLES START  if not initialized and not offline(@|PLC|@)then  begin  // initialize counter  counter := @|PLC|@:ABS\_CNT1;  counterOld := @|PLC|@:ABS\_CNT1;  // set initialized to perform initializing once  initialized := true;  end  else if initialized then  begin  counter := @|PLC|@:ABS\_CNT1;  end;  // INITIALIZE SCRIPT VARIABLES END  // ACTIONS ONCE PER SECOND START  oncePerSecond  begin  seconds:= seconds + 1;  end;  // ACTIONS ONCE PER SECOND END  // LOGGING SIGNALS WHEN CHANGED START  logstring := "@|PLC|@ Signals : " + " @|PLC|@ offline : " + toString(offline(@|PLC|@))  + " M\_ON : " + "@|PLC|@:M\_ON"  + " M\_MODE : " + "@|PLC|@:M\_MODE"  + " M\_FEED\_OVR : " + "@|PLC|@:M\_FEED\_OVR"  + " M\_FEED\_RATE : " + "@|PLC|@:M\_FEED\_RATE"  + " M\_SPINDLE\_SPEED : " + "@|PLC|@:M\_SPINDLE\_SPEED"  + " ABS\_CNT1 : " + toString(@|PLC|@:ABS\_CNT1);  if logString <> logstringOld then  begin  stdlog("Javis-DACQ", "I", 0, logString);  logstringOld := logString;  end;  // LOGGING SIGNALS WHEN CHANGED END  // DEFINITION state malfunction START  if offline(@|PLC|@) then // check if MTConnect agent is online  begin  state := 1; // 1 = No production 2 = production  malfunction := 12; // 998 - no connection  end  else if (@|PLC|@:M\_SPINDLE\_SPEED="UNAVAILABLE") then // check if MTConnect agent is able to read Variables  begin  if seconds > 2 then // Variable unavailable longer than 2 seconds  begin  state := 1; // 1 = No production 2 = production  malfunction := 12; // 998 - no connection  seconds := 0; // reset seconds counter  end;  end  else if not (@|PLC|@:M\_ON = "ON") then  begin  state := 1; // 1 = No production 2 = production  malfunction := 9; // 901 - machine switched off  end  else if not (@|PLC|@:M\_AUTO = "AUTOMATIC") then  begin  state := 1; // 1 = No production 2 = production  malfunction := 7; // 995 - machine not in automatic mode  end  else if (@|PLC|@:M\_ON="ON")  and toNumber(@|PLC|@:M\_SPINDLE\_SPEED)>0 // Spindle speed greater 0  and toNumber(@|PLC|@:M\_FEED\_RATE)>0 // Feed rate greater 0  and toNumber(@|PLC|@:M\_FEED\_OVR)>80 // Feed override greater 80 percent  and (@|PLC|@:M\_MODE="AUTOMATIC") then  begin  state := 2; // 1 = No production 2 = production  malfunction := 0; // 0 = no malfunction  end  else  begin // all other cases  state := 1; // 1 = No production 2 = production  malfunction := 1; // 1 = 999 = undefined stoppage  end;  // DEFINITION state malfunction END  // DEFINITION COUNTER START  if counter > counterOLD then // counter on PLC is incremented  begin  counterSend := counter - counterOLD;  counterOLD := counter;  end  else if counter < counterOLD then // counter on PLC is reset  begin  counterSend := counter;  counterOLD := counter;  end  else  begin  counterSend := 0;  end;  // DEFINITION COUNTER END  // SEND state malfunction START  if (malfunction <> malfunctionOld) or (state <> stateOld) then  begin  stdlog("Javis-DACQ", "I", 0, "@|WPL|@ send state " + toString(state) + " reason " + toString(malfunction));  debugOut("@|WPL|@ send state " + toString(state) + " reason " + toString(malfunction)+ "\n");  sendStateWorkplace("@|WPL|@", state, malfunction);  malfunctionOld := malfunction;  stateOld := state;  end;  // SEND state malfunction END    // SEND STROKES / QUANTITY START  if counterSend > 0 then  begin  stdlog("Javis-DACQ", "I", 0, "@|WPL|@ send strokes or quantity : " + toString(counterSend));  debugOut("@|WPL|@ send strokes or quantity : " + toString(counterSend) + " \n");  sendStrokesWorkplace("@|WPL|@", counterSend, 1); //send strokes to workplace  // sendCountWorkplace("@|WPL|@", 1, counterSend); //send quantity to workplace  counterSend := 0;  end;  // SEND STROKES / QUANTITY END  end; |

# XML file structure

MTConnect is based on the modern ReSTful (XML) Web interface.

With MTConnect, requests are HTTP requests with an empty body. All relevant information's are included in the URL. The answering machine always returns an XML document. An MTConnect agent must support four different XML schemes:

* **probe**: to list the components and DataItems of a device
* **current**: for a snapshot of the last known values of an DataItems
* **sample**: to read out the data in the samples, events and conditions in a specified interval
* **asset**: to query the current status of an asset, associated with the device

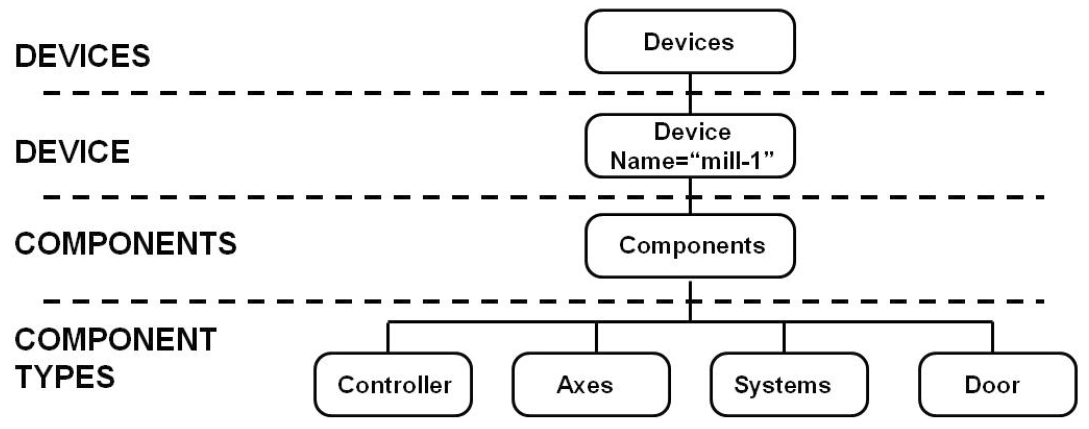
MTConnect determines for reasons of data economy, to transfer always only the changed values.

## Devices and Components

The following text and pictures are an excerpt from the MTConnect Wiki: http://mtcup.org/wiki/ Data\_Items. This documentation describes the possibilities of MTConnect.

MTConnect organized information and data from a machine control unit in an information model, which defines the relationship between each data element and the source of the data. An application can use this information model to interpret and process the received data from a machine.

The basic model of the entire information contains three primary container: Devices, devices and components. These containers are the building blocks in order to organize information about a piece of equipment. They also define how the various pieces of equipment are interrelated.



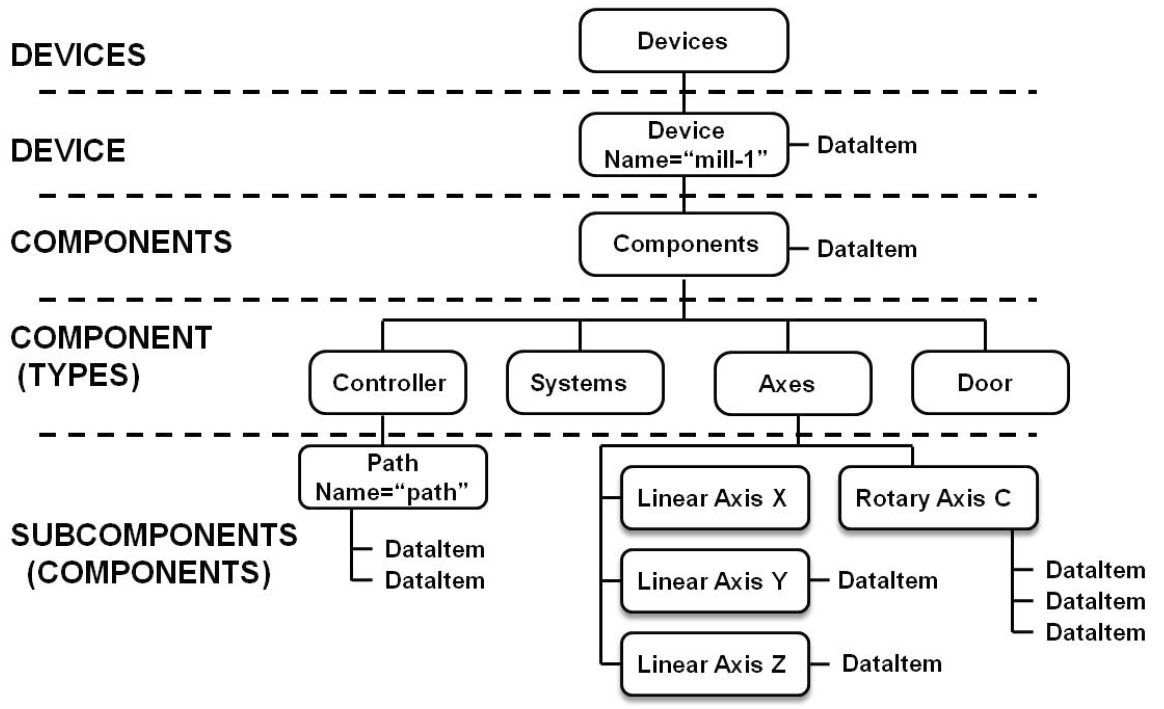
The first or highest container in the entire data structure is the equipment container. The devices container consists of minimum one device XML element. The devices container provides a mechanism to group data from multiple device elements that provide their data via a common MTConnect agent. Devices have no attributes, and serve only to group data from appliance elements.

The next level container is a device. A device typically represents a single piece of equipment or a machine. However, it can also represent any logical grouping of components that operate together to perform a function. Every device in MTConnect must have an availability data item. Availability represents the device’s ability to provide information about itself.

The third container in the entire data structure is the component container. Components provides a mechanism for grouping sub-elements of a device in logical groups that are associated to each other. Components have no attributes and they are used for grouping of elements.

## Data Item

The data item is a piece of information that can be collected from a device, component or subcomponent. A data item can maybe report both, a numerical value (a numerical size either reported as a sample or event category), as well as a condition status (as a condition category reported). A data item specifies the type of data collected and an array of optional attributes further defines these data. The data is provided in the response of the stream.



Source: http://mtcup.org/wiki/Devices\_and\_Components

## Data Item Category

MTConnect offers three different categories of data items - SAMPLE, EVENT, and CONDITION. The category indicates, where are the results in the XML document, in response to a sample or the current request.

**SAMPLE**

An example is the reading the value of a continuous variable or an analog data element. A continuous value can be scanned at any point in time and will always produce a result. An example of a continuous data element is the axis Linear X. A data item of category SAMPLE must have a unit.

**EVENT**

A data item of the category EVENT has discrete information from the device. There are two types of events: representing the state, with two or more discrete values; and those that contain plain text messages. An example of a state of type EVENT is a door status, which is either open, unlocked, or closed. Another example of an EVENT type is a message which contains a machine program with any valid string of characters. A data element of EVENT has no intermediate values that vary over time, as well as in samples.

**CONDITION**

A data element that communicates the condition and function of this unit. A data item of the class CONDITION can be one of: UNAVAILABLE, NORMAL, WARNING or a MISTAKE. A data item of class CONDITION can transmit multiple active states at the same time; where an item of data Class SAMPLE or EVENT can only have one value at any given time.

**Data Item Types for SAMPLE Category**

|  |  |  |
| --- | --- | --- |
| **Data Item type/subtype** | **Description** | **Units** |
| **ACCELERATION** | Rate of change of velocity | MILLIMETER/SECOND^2 |
| **ACCUMULATED\_TIME** | The measurement of accumulated time associated with a Component | SECOND |
| **ANGULAR\_ACCELERATION** | Rate of change of angular velocity. | DEGREE/SECOND^2 |
| **ANGULAR\_VELOCITY** | Rate of change of angular position. | DEGREE/SECOND |
| **AMPERAGE** | The measurement of AC Current or a DC current | AMPERE |
| ALTERNATING | The measurement of alternating current. If not specified further in statistic, defaults to RMS Current | AMPERE |
| DIRECT | The measurement of DC current | AMPERE |
| **ANGLE** | The angular position of a component relative to the parent. | DEGREE |
| ACTUAL | The angular position as read from the physical component. | DEGREE |
| COMMANDED | The angular position computed by the Controller. | DEGREE |
| **AXIS\_FEEDRATE** | The feedrate of a linear axis. | MILLIMETER/SECOND |
| ACTUAL | The actual federate of a linear axis. | MILLIMETER/SECOND |
| COMMANDED | The feedrate as specified in the program. | MILLIMETER/SECOND |
| OVERRIDE | The operator’s overridden value. Percent of commanded. | PERCENT |
| **CLOCK\_TIME** | The reading of a timing device at a specific point in time. Clock time **MUST** be reported in W3C ISO 8601 format. | YYYY-MM-DDThh:mm:ss.ffff |
| **CONCENTRATION** | Percentage of one component within a mixture of components | PERCENT |
| **CONDUCTIVITY** | The ability of a material to conduct electricity | SIEMENS/METER |
| **DISPLACEMENT** | The displacement as the change in position of an object | MILLIMETER |
| **ELECTRICAL\_ENERGY** | The measurement of electrical energy consumption by a component | WATT\_SECOND |
| **FILL\_LEVEL** | The measurement of the amount of a substance remaining compared to the planned maximum amount of that substance | PERCENT |
| **FLOW** | The rate of flow of a fluid | LITER/SECOND |
| **FREQUENCY** | The measurement of the number of occurrences of a repeating event per unit time | HERTZ |
| **LINEAR\_FORCE** | The measure of the push or pull introduced by an actuator or exerted on an object | NEWTON |
| **LOAD** | The measurement of the percentage of the standard rating of a device | PERCENT |
| **MASS** | The measurement of the mass of an object(s) or an amount of material | KILOGRAM |
| **PATH\_FEEDRATE** | The feedrate of the tool path. | MILLIMETER/SECOND |
| ACTUAL | The three-dimensional feedrate derived from the Controller. | MILLIMETER/SECOND |
| COMMANDED | The feedrate as specified in the program | MILLIMETER/SECOND |
| OVERRIDE | The operator’s overridden value. Percent of commanded. | PERCENT |
| **PATH\_POSITION** | The current program control point or program coordinate in WORK coordinates. The coordinate system will revert to MACHINE coordinates if WORK coordinates are not available. | MILLIMETER\_3D |
| ACTUAL | The position of the Component as read from the device. | MILLIMETER\_3D |
| COMMANDED | The position computed by the Controller. | MILLIMETER\_3D |
| TARGET | The target position for the movement. | MILLIMETER\_3D |
| PROBE | The position provided by a probe | MILLIMETER\_3D |
| **PH** | The measure of the acidity or alkalinity. | PH |
| **POSITION** | The position of the Component. Defaults to MACHINE coordinates. | MILLIMETER |
| ACTUAL | The position of the Component. | MILLIMETER |
| COMMANDED | The position as given by the Controller. | MILLIMETER |
| TARGET | The target position for the movement. | MILLIMETER |
| POWER\_FACTOR | The measurement of the ratio of real power flowing to a load to the apparent power in that AC circuit. | PERCENT |
| PRESSURE | The force per unit area exerted by a gas or liquid | PASCAL |
| RESISTANCE | The measurement of the degree to which an object opposes an electric current through it | OHM |
| ROTARY\_VELOCITY | The rotational speed of a rotary axis. | REVOLUTION/MINUTE |
| ACTUAL | The rotational speed the rotary axis is spinning at. ROTARY\_MODE MUST be SPINDLE. | REVOLUTION/MINUTE |
| COMMANDED | The rotational speed as specified in the program. | REVOLUTION/MINUTE |
| OVERRIDE | The operator’s overridden value. Percent of commanded. | PERCENT |
| **SOUND\_LEVEL** | Measurement of a sound level or sound pressure level relative to atmospheric pressure | DECIBEL |
| NO\_SCALE | No weighting factor on the frequency scale | DECIBEL |
| A\_SCALE | A Scale weighting factor. This is the default weighting factor if no factor is specified | DECIBEL |
| B\_SCALE | B Scale weighting factor | DECIBEL |
| C\_SCALE | C Scale weighting factor | DECIBEL |
| D\_SCALE | D Scale weighting factor | DECIBEL |
| **SPINDLE\_SPEED** | Replaced by ROTARY\_VELOCITY |  |
| ACTUAL | The rotational speed of a rotary axis. ROTARY\_MODE **MUST** be SPINDLE. | REVOLUTION/MINUTE |
| COMMANDED | The rotational speed the as specified in the program. | REVOLUTION/MINUTE |
| OVERRIDE | The operator’s overridden value. Percent of commanded. | PERCENT |
| **STRAIN** | Strain is the amount of deformation per unit length of an object when a load is applied. | PERCENT |
| **TEMPERATURE** | The measurement of temperature | CELSIUS |
| **TILT** | A measurement of angular displacement | MICRO\_RADIAN |
| **TORQUE** | The turning force exerted on an object or by an object | NEWTON\_METER |
| **VOLT\_AMPERE** | The measure of the apparent power in an electrical circuit, equal to the product of root-mean-square (RMS) voltage and RMS current’ (commonly referred to as VA) | VOLT\_AMPERE |
| **VOLT\_AMPERE\_REACTIVE** | The measurement of reactive power in an AC electrical circuit (commonly referred to as var) | VOLT\_AMPERE\_REACTIVE |
| **VELOCITY** | The rate of change of position. | MILLIMETER/SECOND |
| **VISCOSITY** | A measurement of a fluid’s resistance to flow | PASCAL\_SECOND |
| **VOLTAGE** | The measurement of electrical potential between two points | VOLT |
| ALTERNATING | The measurement of alternating voltage. If not specified further in statistic, defaults to RMS voltage | VOLT |
| DIRECT | The measurement of DC voltage | VOLT |
| **WATTAGE** | The measurement of power consumed or dissipated by an electrical circuit or device | WATT |

**Data Item Types for EVENT Category**

|  |  |
| --- | --- |
| **ACTUATOR\_STATE** | The state of the Actuator - ACTIVE or INACTIVE. |
| **ALARM** | DEPRECATED: Replaced with CONDITION category. *Rel. 1.1*. |
| **ACTIVE\_AXES** | The set of axes associated with a Path that the Controller is controlling. If this DataItem is not provided, it will be assumed the Controller is controlling all axes. |
| **AVAILABILITY** | Represents the ability of a Component to communicate. This **MUST** be provided for a Device and **MAY** be provided for any other Component. AVAILABLE or UNAVAILABLE. |
| **AXIS\_COUPLING** | Describes the way the axes will be associated to each other. This is used in conjunction with COUPLED\_AXES to indicate the way they are interacting. The possible values are: TANDEM, SYNCHRONOUS, MASTER, and SLAVE. The coupling **MUST** be viewed from the perspective of the axis, therefore a MASTER coupling indicates that this axis is the  master of the COUPLED\_AXES. |
| **BLOCK** | The block of code being executed. Block contains the entire expression for a line of program code. |
| **CONTROLLER\_MODE** | The current mode of the Controller. AUTOMATIC, MANUAL, MANUAL\_DATA\_INPUT, or SEMI\_AUTOMATIC. |
| **COUPLED\_AXES** | Refers to the set of associated axes. The value will be a space delimited set of axes names. |
| **DIRECTION** | The direction of motion. CLOCKWISE or COUNTER\_CLOCKWISE |
| **ROTARY** | The rotational direction of a rotary device using the right hand rule convention as defined in Appendix B. CLOCKWISE or COUNTER\_CLOCKWISE |
| **LINEAR** | The direction of motion of a linear device. POSTIVE or NEGATIVE |
| **DOOR\_STATE** | The opened or closed state of the door. OPEN, UNLATCHED, or CLOSED. |
| **EMERGENCY\_STOP** | The current state of the emergency stop actuator. ARMED (the circuit is complete and the device is operating) or TRIGGERED (the circuit is open and the device MUST cease operation). |
| **EXECUTION** | The execution status of the Controller. READY, ACTIVE, INTERRUPTED, FEED\_HOLD, or STOPPED |
| **LINE** | The current line of code being executed |
| **MAXIMUM** | The maximum line number of the code being executed. |
| **MINIMUM** | The minimum line number of the code being executed. |
| **MESSAGE** | An uninterpreted textual notification. |
| **PALLET\_ID** | The identifier for the pallet currently in use for a given Path |
| **PART\_COUNT** | The current count of parts produced as represented by the Controller. MUST be an integer value. |
| **ALL** | The count of all the parts produced. If the subtype is not given, this is the default. |
| **GOOD** | Indicates the count of correct parts made. |
| **BAD** | Indicates the count of incorrect parts produced. |
| **PART\_ID** | An identifier of the current part in the device |
| **PATH\_MODE** | The operational mode for this Path. SYNCHRONOUS, MIRROR, or INDEPENDENT. Default value is INDEPENDENT if not specified. |
| **POWER\_STATE** | The ON or OFF status of the Component. DEPRECATION WARNING: MAY be deprecated in the future. |
| **LINE** | The state of the high voltage line. |
| **CONTROL** | The state of the low power line. |
| **POWER\_STATUS** | DEPRECATED. Rel. 1.1. |
| **PROGRAM** | The name of the program being executed |
| **ROTARY\_MODE** | The mode for the Rotary axis. SPINDLE, INDEX, or CONTOUR. |
| **TOOL\_ID** | DEPRECATED in Rel. 1.2. See Tool\_ASSET\_ID. The identifier of the tool currently in use for a given Path |
| **TOOL\_ASSET\_ID** | The identifier of an individual tool asset. |
| **TOOL\_NUMBER** | The identifier of a tool provided by the device controller. |
| **WORKHOLDING\_ID** | The identifier for the workholding currently in use for a given Path |

**Data Item Types for CONDITION Category**

|  |  |
| --- | --- |
| **Data Item type/ qualifier** | **Description** |
| **ACCELERATION** | Rate of Change of Velocity |
| **ACCUMULATED\_TIME** | The measurement of accumulated time associated with a Component |
| **ACTUATOR** | An actuator related condition. |
| **AMPERAGE** | A high or low condition for the electrical current. |
| **ANGLE** | The angular position of a Component. |
| **ANGULAR-ACCELERATION** | Rate of change of angular velocity. |
| **ANGULAR\_VELOCITY** | Rate of change of angular position |
| **COMMUNICATIONS** | A communications failure indicator. |
| **CONCENTRATION** | Percentage of one ingredient within a mixture of ingredients |
| **CONDUCTIVITY** | The ability of a material to conduct electricity |
| **DATA\_RANGE** | Information provided is outside of expected value range |
| **DIRECTION** | The direction of motion of a Component |
| **DISPLACEMENT** | The change in position of an object |
| **ELECTRICAL\_ENERGY** | The measurement of electrical energy consumption by a Component |
| **FILL\_LEVEL** | Represents the amount of a substance remaining compared to the planned maximum amount of that substance |
| **FLOW** | The rate of flow of a fluid |
| **FREQUENCY** | The number of occurrences of a repeating event per unit time |
| **HARDWARE** | The hardware subsystem of the Component’s operation condition. |
| **LEVEL** | **See FILL\_LEVEL** |
| **LINEAR\_FORCE** | The measure of the push or pull introduced by an actuator or exerted by an object |
| **LOAD** | The measure of the percentage of the standard rating of a device |
| **LOGIC\_PROGRAM** | An error occurred in the logic program or PLC (programmable logic controller). |
| **MASS** | The measurement of the mass of an object(s) or an amount of material |
| **MOTION\_PROGRAM** | An error occurred in the motion program. |
| **PATH\_FEEDRATE** | The federate of the tool path |
| **PATH\_POSITION** | The current control point of the path |
| **PH** | The measure of acidity or alkalinity |
| **POSITION** | The position of a Component. |
| **POWER\_FACTOR** | The ratio of real power flowing to a load to the apparent power in that AC circuit. |
| **PRESSURE** | The measurement of the force per unit area exerted by a gas or liquid. |
| **RESISTANCE** | The measurement of the degree to which an object opposes an electric current through it |
| **ROTARY\_VELOCITY** | The rotational speed of a rotary axis |
| **SOUND\_LEVEL** | The measurement of sound pressure level |
| **SPINDLE\_SPEED** | DEPRECATED in Rel 1.2. See ROTARY\_VELOCITY |
| **STRAIN** | Indicates the amount of deformation per unit length of an object when a load is applied |
| **SYSTEM** | A condition representing something that is not the operator, program, or hardware. This is often used for operating system issues. |
| **TEMPERATURE** | Indicates the temperature of a Component. |
| **TILT** | The measure of angular displacement |
| **TORQUE** | The measured of the turning force exerted on an object or by an object |
| **VOLT\_AMPERAGE** | The measure of the apparent power in an electrical circuit (commonly referred to as VA) |
| **VOLT\_AMPERAGE\_REACTIVE** | The measure of reactive power in an AC electrical power circuit (commonly referred to as var). |
| **VELOCITY** | Indicated the velocity of a component. |
| **VISCOSITY** | The measure of a fluid’s resistance to flow |
| **VOLTAGE** | The measurement of electrical potential between two points |
| **WATTAGE** | The measurement of power consumed or dissipated by an electrical circuit or device |

Source: http://www.mtconnect.org (2012)