MTConnect OPC UA

Companion Specification

Release Candidate

Version 1.02

November 20th, 2013

CONTENTS

Contents

1 Introduction

1.1 Background

In September 2010, the OPC Foundation and the MTConnect Institute signed a memorandum of understanding to provide a mechanism for OPC and MTConnect to collaborate to extend the reach of the existing manufacturing data exchange standards and implementation technologies in order to:

- Evolve the existing standards for each organization to provide complete manufacturing technology interoperability.
- Provide the mechanism for continuous improvement of standards and specifications overseen by each body.
- Work directly with the end users and suppliers of technology and manufacturing.
- Provide a coordinating function to exchange insights, identify overlaps, and harmonize work where appropriate.
- Facilitate clear communication and education for users and others concerning possible overlaps and the ways the standards and specifications can be used.
- Provide a solid foundation to develop and deliver specifications, technology and processes to facilitate adoption of the technology into real products.

The outcome of that agreement is this companion specification called MTConnect-OPC UA. MTConnect-OPC UA companion specification ensures interoperability and consistency between MTConnect specifications and the OPC Unified Architecture (UA) specifications, as well as the manufacturing technology equipment, devices, software or other products that implement those standards.

1.2 MTConnect-OPC UA Goals

MTConnect-OPC UA is designed with all of the following first-class goals in mind, in the interest of wide and rapid adoption by vendors of equipment and software:

- *Incremental adoption*—the technical barrier to MTConnect-OPC UA enablement will be greatly reduced with this companion specification and the source code and binaries available in the MTConnect-OPC UA reference port.
- *Evolution*—MTConnect-OPC UA can incrementally evolve without jeopardizing backwards compatibility of previous MTConnect-OPC UA versions.
- *Customizability*—MTConnect-OPC UA's extensibility makes it easy to create value-added software and tools that are machine-specific or installation-specific, without jeopardizing compatibility with other equipment or software.
- *Non-proprietary*—built on open standards, backed by both the OPC Foundation and the MTConnect Institute which represents hundreds of companies, individuals, government organizations and non-profits all working toward the goal of increased productivity in the manufacturing arena.

1.3 Who Will Find Benefit from this Specification?

To adopt the MTConnect-OPC UA one will need to have a clear understanding of both MTConnect and OPC UA. From the technical side, we will discuss MTConnect-OPC UA from:

- The backend or OPC UA Server and MTConnect agent/adapter architecture.
- The client or software application side, we will discuss how one develops an application that is MTConnect-OPC UA enabled.

From the business side, we will reference a companion business MTConnect-OPC UA white paper that addresses the concerns from the owners and top management of the business as well as the operations and engineering management. It is the objective of this white paper to provide information primarily to MTConnect and OPC UA software developers. We do not make assumptions about the level of programming expertise beyond what would be considered to be “reasonable” level of expertise. It is for this reason that we include enough details about both MTConnect and OPC UA to provide the ability to implement this companion specification without having references back to other documents. However, the OPC and MTConnect standards are critical and become much more meaningful with the appropriate overview from this document.

1.4 References

1.4.1 OPC Foundation

The following specifications from the OPC foundation are referenced by this specification.

[UA Part 1] OPC UA Specification: Part 1 – Concepts
http://www.opcfoundation.org/UA/Part1/

[UA Part 2] OPC UA Specification: Part 2 – Security Model
http://www.opcfoundation.org/UA/Part2/

[UA Part 3] OPC UA Specification: Part 3 – Address Space Model
http://www.opcfoundation.org/UA/Part3/

[UA Part 4] OPC UA Specification: Part 4 – Services
http://www.opcfoundation.org/UA/Part4/

[UA Part 5] OPC UA Specification: Part 5 – Information Model
http://www.opcfoundation.org/UA/Part5/

[UA Part 6] OPC UA Specification: Part 6 – Mappings
http://www.opcfoundation.org/UA/Part6/

[UA Part 7] OPC UA Specification: Part 7 – Profiles
http://www.opcfoundation.org/UA/Part7/

[UA Part 8] OPC UA Specification: Part 8 – Data Access
http://www.opcfoundation.org/UA/Part8/

[UA Part 9] OPC UA Specification: Part 9 – Alarms and Conditions
http://www.opcfoundation.org/UA/Part9/

[UA Part 10] OPC UA Specification: Part 10 – Programs
http://www.opcfoundation.org/UA/Part10/

[UA Part 11] OPC UA Specification: Part 11 – Historical Access
http://www.opcfoundation.org/UA/Part11/

[UA Part 13] OPC UA Specification: Part 13 – Aggregates
http://www.opcfoundation.org/UA/Part13/

1.4.2 MTConnect Institute

The following specifications from MTConnect are referenced by this specification.

[MT Part 1] MTConnect® Standard: Part 1 – Overview, Version 1.2
http://www.mtconnect.org/media/23617/mtc_part_1_overview_v1.2.pdf

[MT Part 2] MTConnect® Standard: Part 2 – Components and DataItems, Version 1.2
http://www.mtconnect.org/media/23620/mtc_part_2_components_v1.2.pdf

[MT Part 3] MTConnect® Standard: Part 3 – Streams, Version 1.2
http://www.mtconnect.org/media/23623/mtc_part_3_streams_v1.2.pdf

[MT Part 4] MTConnect® Standard: Part 4 – Assets, Version 1.2
http://www.mtconnect.org/media/23626/mtc_part_4_assets_v1.2.pdf

1.5 Abbreviations

The following abbreviation are used in this document

- ERP – Enterprise Resource Planning
- HMI – Human Machine Interface
- Http – Hyper Text Transport Protocol
- MES – Management Execution Systems
- PLC – Programmable Logic Controller
- PMS - Production Management Systems
- SCADA - Supervisory Control And Data Acquisition
- TCP/IP - Transmission Control Protocol/Internet Protocol
- XML - eXtensible Mark-up Language

2 Use Cases

2.1 Overview

Before delving into the details of the specification it is useful to identify some of the key use cases for the technology. The use cases defined here are not an exhaustive list; however, they should help demonstrate how this specification is expected to be used and to help illustrate the benefits of a common information model.

2.2 Device Maker

The use case, shown in Figure 1, centers on the manufacturer of a piece of equipment or device that needs to provide connectivity to other systems. In some cases, the device manufacturer will be targeting markets other than equipment (Machine Tool) and would benefit from a more generic specification like OPC UA. On the other hand, the standardized semantics of MTConnect are extremely important to interoperability within the Machine Tools space. The MTConnect-OPC UA specification and the resulting standard information model allows the device manufacturers to standardize on OPC UA as the network interface while making their information accessible to MTConnect aware applications. Figure 1 shows several clients developed for different purposes that can access information produced by the device via OPC UA.

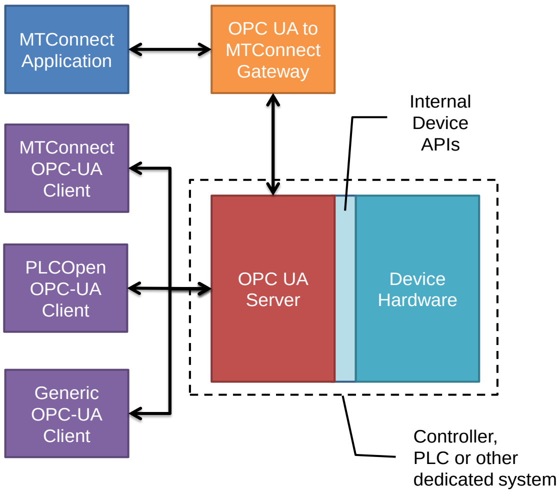


Figure 1 – The Device Manufacturer Use Case

The device manufacture may also have a native MTConnect device and make use of an MTConnect to OPC UA gateway to provide the information to OPC UA aware clients (see Figure 2). This standard allows for easy information flow between Client and Server that support either MTConnect or OPC UA.

The MTConnect or OPC UA interface may reside directly in the Machine, but it may also reside in some other device that communicates with the Machine. The actual location for the interface is up to the Machine vendor.

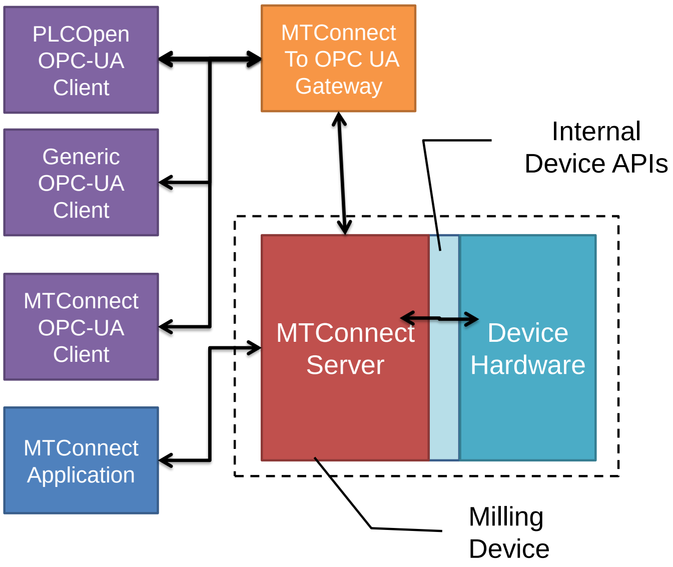


Figure 2 – Device Manufacturer Native MTConnect

2.3 Independent Software Vendor

The use case shown in Figure 3 centers on an Independent Software Vendor (ISV) that wishes to sell products to users of equipment such as Machine Tools. An ISV will typically want to provide gateways that convert information between MTConnect and OPC UA as well as adding numerous features that add value to the semantics defined in the MTConnect standards. The MTConnect-OPC UA specification allows the ISV to extend the MTConnect-OPC UA information model with application specific constructs which can be easily accessed via any standard OPC UA client product. These added features will exist in parallel to the standard MTConnect interfaces. Figure 3 shows an ISV product that consumes data from MTConnect and OPC UA enabled devices and then makes it available via MTConnect and OPC UA.

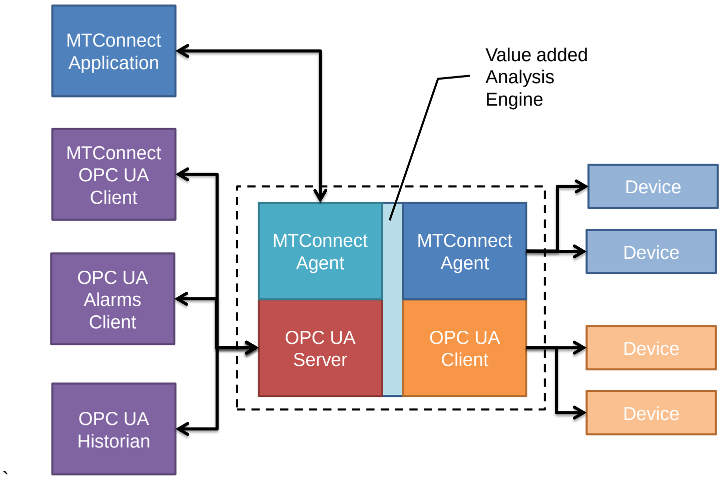


Figure 3 – The Independent Software Vendor (ISV) Use Case

2.4 End-User Engineer

This use case shown in Figure 4 centers on an Engineer or Systems Integrator responsible for setting up and configuring an MTConnect enabled system for a user of Machine Tools. The Engineer is typically familiar with the MTConnect specification but wishes to configure generic OPC UA client applications. The MTConnect-OPC UA specification allows the Engineer to understand how MTConnect concepts are represented in OPC UA and determine what they need to do to configure their OPC UA Applications. Without this specification, an Engineer interested in OPC based data would have had to rely on vendor documentation and a laborious process of manually mapping tags to MTConnect concepts. This specification eliminates the need for that by providing a standard mapping. Figure 4 shows how the common Information Model defined by this specification gives the End User Engineer choices when it comes to accessing device data.

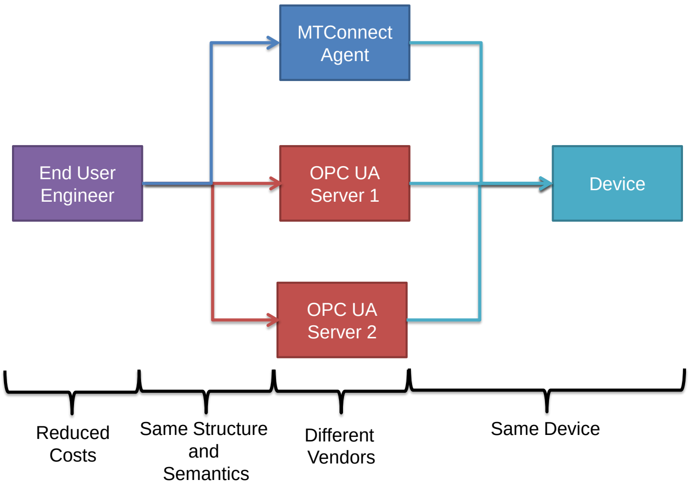


Figure 4 - The End User Engineering Use Case

3 MTConnect

3.1 What is MTConnect?

MTConnect is an open and royalty-free set of standards designed as a universal factory floor communications protocol. MTConnect is intended specifically for the shop floor environment. While there are numerous communication solutions available, MTConnect defines a “dictionary” for manufacturing data. This means that all data is provided with full context – name, definition, scaling, etc. With most communication networks, all data is defined at the point of use – the application. With MTConnect, the data is defined at the source – the device or Machine Tool. MTConnect devices process information locally and then provide that data in a consistent format to any client application requesting data - ERP, MES, Production Management Systems, Maintenance Systems or a standard Browser, for examples.

3.2 Basics of MTConnect

MTConnect is based on standard Internet technologies – HTTP, Ethernet, and XML (Extensible Mark-Up Language – the underlying language of most web sites).

As an Extensible Standard, MTConnect cannot address every conceivable data need on the shop floor. MTConnect provides a clearly defined method for adding additional data types which can be exchanged between equipment, devices, controllers and applications; providing the flexibility to meet the demands of varying environments.

MTConnect is made up of five fundamental components (see Figure 5 below):

Device – A type of equipment (Machine Tool) or data source.

Adapter – An optional piece of software (and sometimes hardware) that provides a link or conversion from the data source and data definition in the device to the MTConnect Data definition. This can be thought of as a translator. The Adapter is not needed for devices that use MTConnect as their native language.

Agent – A piece of software that collects, arranges, and stores data from the device. It receives requests for data from applications, processes those requests, and then transmits the required data.

Network - The physical connection between a data source (device) and the data consumer (application). Normally, this is an Ethernet network. The communication on the network normally uses standard internet communications methods – http:// protocol. It should be noted that the MTConnect Structure is adaptable and can be implemented in conjunction with other networking solutions other than Ethernet and Internet protocols.

Client – A Client initiates all requests for MTConnect data. A Client resides in an application or device. The Client is a software function in the application or device that actually requests data from the Agent.

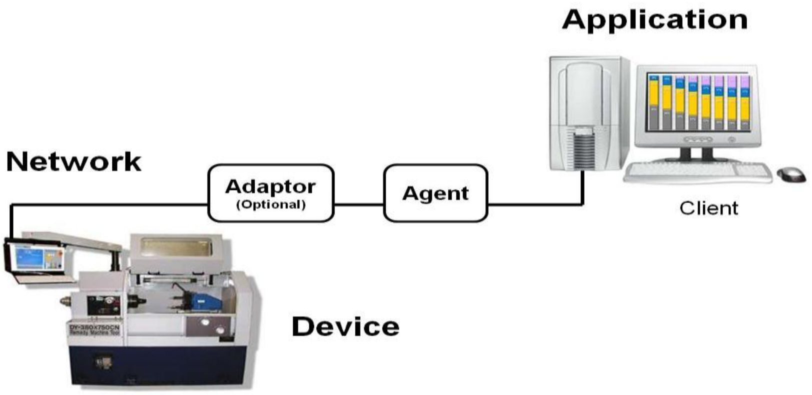


Figure 5 – MTConnect Overview

The MTConnect Standard does not restrict the physical implementation of how the MTConnect system is designed.

- The Network may be a physical implementation, like an Ethernet network. It can also be implemented using wireless or other technologies.
- The Internet Protocol (http) does not mean that your machine is automatically connected outside your plant to the Internet. This is a communications method only. Protection of your data is controlled by your networking standards.
- There is no specific requirement for where the Adaptor and Agent function is located. These can be located at the device. However, they can be placed anywhere in the networking architecture. Also, they do not need to be located together. It is totally MTConnect compliant to have the Adapter installed at the device and the Agent installed along with the Client. The location of these functions should be considered when implementing MTConnect since they will impact the level of data flow on different segments of your network.

4 OPC UA

4.1 What is OPC UA?

OPC UA is an open and royalty free set of standards designed as a universal factory floor communications protocol.

OPC UA is designed specifically for the factory and industrial environment. While there are numerous communication solutions available, OPC UA has several advantages:

- A state of art security model (see [UA Part 2]).
- A fault tolerant communication protocol.
- An information modeling framework that allows application developers to represent their data in a way that makes sense to them.

OPC UA has a broad scope which delivers for economies of scale for application developers. This means that a larger number of high quality applications at a reasonable cost are available to factory owners. When combined with powerful semantic models such as MTConnect, OPC UA makes it easier for factory owners to access data via generic commercial application.

The OPC UA model is scalable from small devices to ERP systems. OPC UA devices process information locally and then provide that data in a consistent format to any application requesting data - ERP, MES, PMS, Maintenance Systems, HMI, Smartphone or a standard Browser, for examples. For a more complete overview see [UA Part 1].

4.2 Basics of OPC UA

As an Open Standard, OPC UA is based on standard Internet technologies – TCP/IP, HTTP, Ethernet, and XML.

As an Extensible Standard, OPC UA provides a set of services (see [UA Part 4]) and a basic information model framework. This framework provides an easy manner for creating and exposing vendor defined information in a standard way. More importantly all OPC UA Clients are expected to be able to discover and use vendor defined information. This means OPC UA users can benefit from the economies of scale that come with generic visualization and historian applications. This specification is an example of an OPC UA Information Model designed to meet the needs of Machine Tool developers and users.

OPC UA Clients can be any consumer of factory data from another device on the network to browser base thin clients and ERP systems. The full scope of OPC UA applications are shown in Figure 6.

The services are described in the following service sets:

- Discovery Service Set – used by a Client to discover the Servers and connection information that are available in a system
- Secure Channel Service Set – used to establish secure communication over which all subsequent communication occurs. Secure channels specify communication protocols and encoding of data. They are used in conjunction with Session Services and provide consistent functionality for all service sets irrespective of the selected communication protocol (TCP, HTTP, HTTPs) data Encoding (OPC Binary, XML) or network architecture (firewalls, routers …).
- Session Service Set – is used to establish a session context that is used for subsequent communication, including user information. This session context is not lost if a communication error occurs , allowing secure channels to be recovered or rebuilt without data loss
- NodeManagement Service Set – allows clients to manage the AddressSpace available in a Server where the AddressSpace is all of the Nodes, both instance and type definition and all of the relationships between them. Management includes adding / deleting nodes and adding / deleting relationships between nodes. Not all Servers support node management functionality.

- View Service Set – used by a Client to discover the information model that is being exposed in the AddressSpace of the Server It include simple browsing of the address space, but also include services to cover browse information to concrete node references (NodeIds)
- Query Service Set – is an extension to view service allowing a client to query the AddressSpace of large servers. Usually on large servers support Query services
- Attribute Service Set – allows a client to read and write current and historical values to nodes in the address spaces.
- Method Service Set – allow Servers to extend the functionality provided by a system, without having to define new services. The methods are defined in the AddressSpace as part of an information model. Client can discover the available methods using the View Service Set and access them via this service set.
- MonitoredItem Service Set – used by client in conjunction with the Subscription Service Set to obtain a steady stream of values. This Service Set allows the client to define the individual data that is to be reported, filters and buffering for the items and the sampling rate.
- Subscription Service Set – used by the client in conjunction with the MonitoredItem Service Set to obtain a steady stream of values. This service set is used to define the grouping of items to be returned and the interval at which they are sent.

These standard services are available with any information model, allowing for generic Client access to any Server.

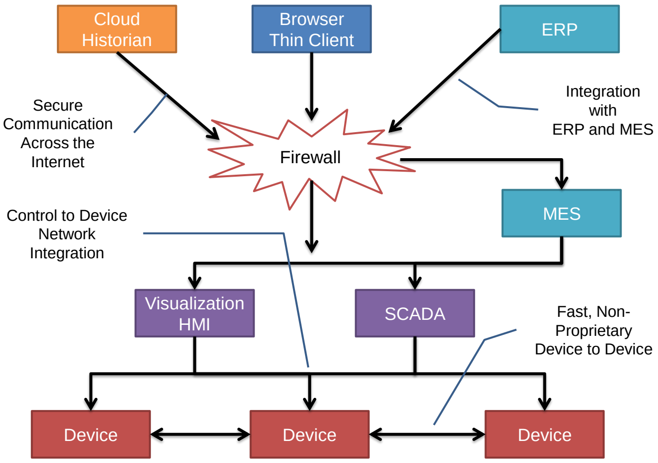

Cloud Historian
Browser Thin Client
ERP
Secure Communication Across the Internet
Integration with ERP and MES
Firewall
MES
Control to Device Network Integration
Visualization HMI
SCADA
Fast, Non-Proprietary Device to Device
Device
Device
Device

Figure 6 – The Scope of OPC UA within an Enterprise

4.3 Information Modeling in OPC UA

4.3.1 Concepts

OPC UA provides a framework that can be used to represent complex information as Objects in an address space which can be accessed with standard web services. These Objects consist of Nodes connected by References. Different classes of Nodes convey different semantics. For example a Variable Node represents a value that can be read or written. The Variable Node has an associated DataType that can define the actual value, such as a string, float, structure etc. It can also describe the variable value as a variant. A Method Node represents a function that can be called. Every Node has a number of Attributes including a unique identifier called a NodeId and non-localized name called as BrowseName. All of these concepts combined together create what is commonly reference to in OPC UA as the Address Space. An Object representing a 'Reservation' is shown in Figure 8 as an illustration of these concepts.

NOTE: The figures used to illustrate OPC UA information models use a notation that was developed for the OPC UA specification. The notation is summarized in Figure 7. UML representations can also be used; however, the OPC UA notation is less ambiguous because there is a direct mapping from the elements in the figures to Nodes in the address space of an OPC UA server. A complete description of the different types of Nodes and References can be found in [UA Part 3] and the base OPC UA Address space is described in [UA Part 5].

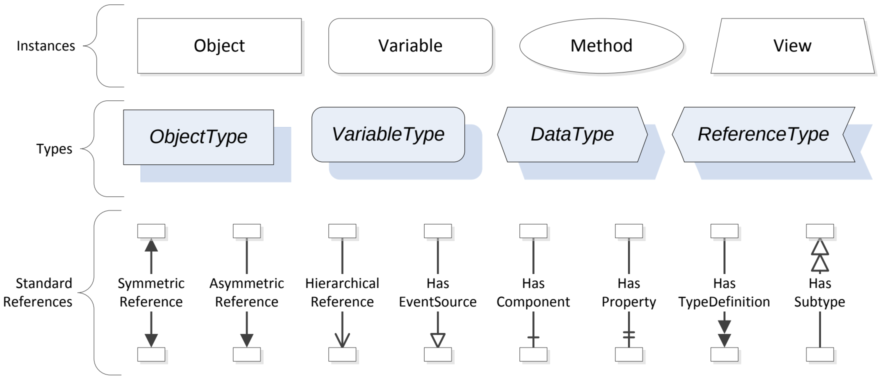


Figure 7 – The OPC UA Information Model Notation

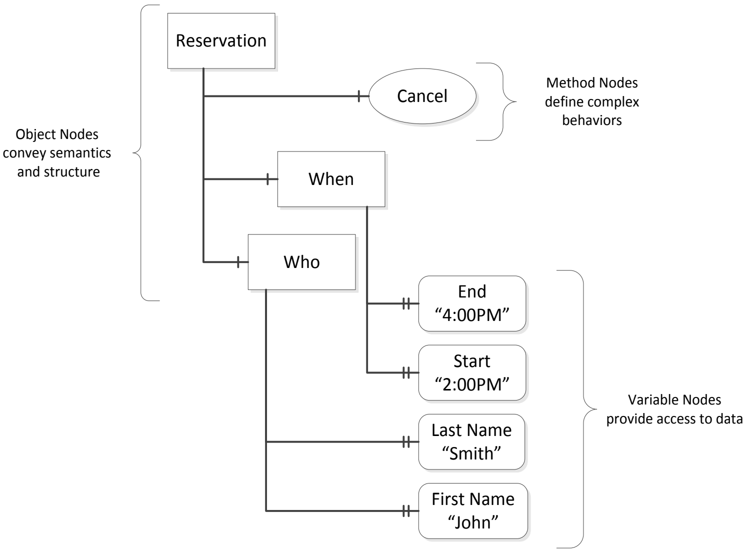

Reservation
Cancel
Method Nodes define complex behaviors
Object Nodes convey semantics and structure
When
Who
End "4:00PM"
Start "2:00PM"
Last Name "Smith"
First Name "John"
Variable Nodes provide access to data

Figure 8 – A Basic Object in an OPC UA Address Space

Object and Variable Nodes are called Instance Nodes and they always reference a Type Definition (ObjectType or VariableType) Node which describes their semantics and structure. Figure 9 illustrates the relationship between an Instance and its Type Definition.

The Type Nodes are templates that define all of the children that can be present in an Instance of the Type. In the example in Figure 9 the PersonType ObjectType defines two children: First Name and Last Name. All instances of PersonType are expected to have the same children with the same BrowseNames. Within a Type the BrowseNames uniquely identify the child. This means Client applications can be designed to search for children based on the BrowseNames from the Type instead of NodeIds. This eliminates the need for manual reconfiguration of systems if a Client uses Types that multiple devices implement.

OPC UA also supports the concept of sub typing. This allows a modeler to take an existing Type and extend it. There are rules regarding sub typing defined in [UA Part 3], but in general they allow the additions to a given type or the restriction of a DataType to a more specific data type. For example the modeler may decide that the existing ObjectType in some cases needs an additional variable. The modeler can create a subtype of the ObjectType and add the variable. A client that is expecting the parent type can treat the new ObjectType as if it was of the parent ObjectType and just ignore the additional variable. A client that understands the new subtype may display or otherwise process the additional variable. With regard to DataTypes, if a variable is defined to have a numeric value, a sub type could restrict the Value to a float.

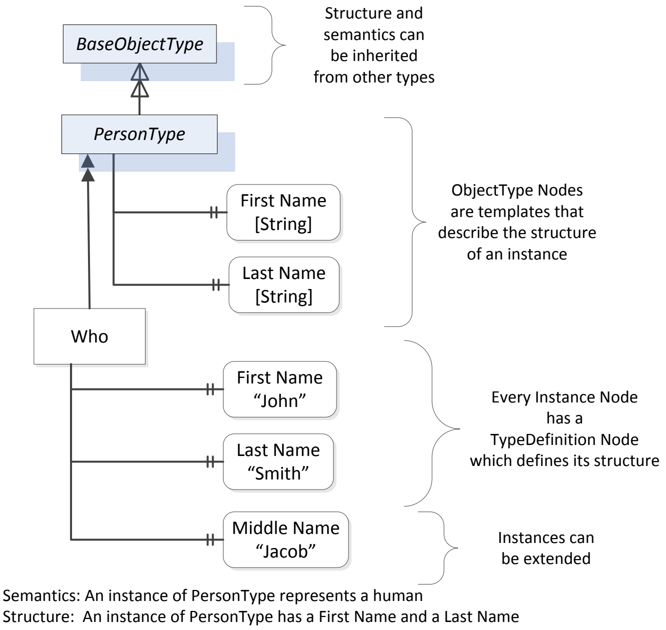

BaseObjectType
Structure and semantics can be inherited from other types
PersonType
First Name [String]
Last Name [String]
ObjectType Nodes are templates that describe the structure of an instance
Who
First Name “John”
Last Name “Smith”
Every Instance Node has a TypeDefinition Node which defines its structure
Middle Name “Jacob”
Instances can be extended
Semantics: An instance of PersonType represents a human
Structure: An instance of PersonType has a First Name and a Last Name

Figure 9 – The Relationship between Type Definitions and Instances

References allow Nodes to be connected together in ways that describe their relationships. All References have a ReferenceType that specifies the semantics of the relationship. References can be hierarchical or non-hierarchical. Hierarchical references are used to create the structure of Objects and Variables. Non-hierarchical are used to create arbitrary associations. Applications can define their own ReferenceType by creating Subtypes of the existing ReferenceType. Subtypes inherit the semantics of the parent but may add additional restrictions. Figure 10 depicts several references connecting different Objects.

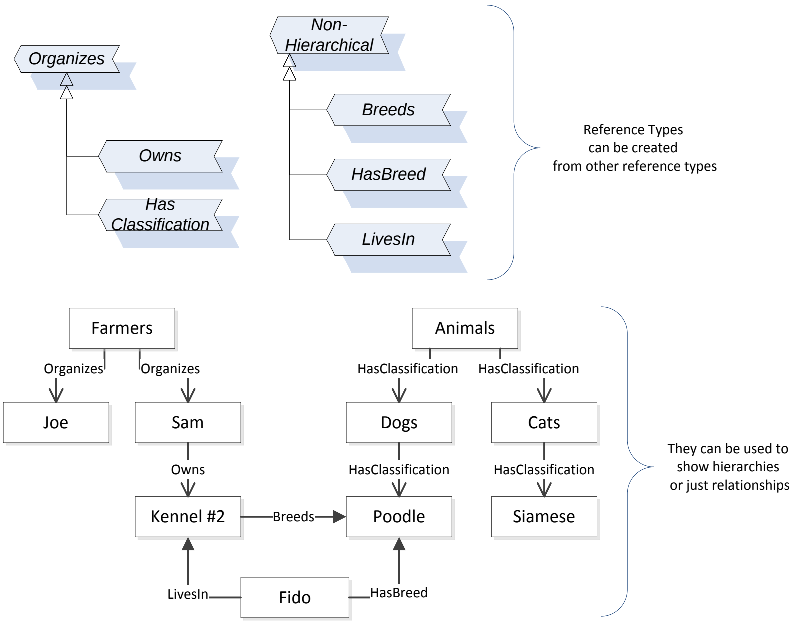


Figure 10 – Examples of References between Objects

OPC UA specification defines a very wide range of functionality in its basic information model. It is not expected that all clients or servers support all functionality in the OPC UA specifications. OPC UA includes the concept of profiles, which segment the functionality into testable certifiable units. This allows the development of companion specification (such as MTConnect-OPC UA) that can describe the subset of functionality that is expected to be implemented. The profiles do not restrict functionality, but generate requirements for a minimum set of functionality (see [UA Part 7])

The OPC foundation also defines a set of information models that provide a basic set of functionality. The Data Access specification (see [UA Part 8]) provides a basic information model for typical data. The Alarm and Condition specification (see [UA Part 9]) defines a standard information model for Alarms and Conditions. The Programs specification (see [UA Part 10]) defines a stand information model for extending the functionality available via method calls and state machines. The Historical Access specification (see [UA Part 11]) defines the information model associated with Historical Data and Historical Events. The aggregates specification (see [UA Part 13]) defines a series of standard aggregate function that allow a client to request summary data. Examples of aggregates include averages, minimums, time in state, Standard deviation, etc.

4.3.2 Namespaces

OPC UA allows information from many different sources to be combined into a single coherent address space. Namespaces are used to make this possible by eliminating naming and id conflicts between information from different sources. Namespaces in OPC UA have a globally unique string called a NamespaceUri and a locally unique integer called a NamespaceIndex. The NamespaceIndex is only unique within the context of a Session between an OPC UA Client and an OPC UA Server. All of the web services defined for OPC UA use the NamespaceIndex to specify the Namespace for qualified values.

There are two types of values in OPC UA that are qualified with Namespaces: NodeIds and QualifiedNames. NodeIds are globally unique identifiers for Nodes. This means the same Node with the same NodeId can appear in many Servers. This, in turn, means Clients can have built in knowledge of some Nodes. OPC UA Information Models generally define globally unique NodeIds for the TypeDefinitions defined by the Information Model.

QualifiedNames are non-localized names qualified with a Namespace. They are used for the BrowseNames of Nodes and allow the same Names to be used by different information models without conflict. The BrowseName is used to identify the children within a TypeDefinitions. Instances of a TypeDefinition are expected to have children with the same BrowseNames. TypeDefinitions are not allowed to have children with duplicate BrowseNames; however, Instances do not have that restriction.

4.3.3 Companion Specifications

An OPC UA companion specification for an industry specific vertical market describes an information model by defining ObjectTypes, VariableTypes, DataTypes and ReferenceTypes that represent the concepts used in the vertical market. Table 1 contains an example of an ObjectType definition.

Table 1 – Example *ObjectType* Definition

Attribute	**Value**				
BrowseName	WidgetType				
IsAbstract	True				
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the *BaseObjectType* from [UA Part 5].					
HasProperty	Variable	Color	String	PropertyType	Mandatory
HasProperty	Variable	Flavor	LocalizedText	PropertyType	Mandatory
HasProperty	Variable	Rank	Int32	PropertyType	Mandatory

The BrowseName is a non-localized name for an ObjectType.

IsAbstract is a flag indicating whether instances of the ObjectType can be created.

The bottom of the table lists the child nodes for the type. The Reference is the type of reference between the Object instance and the child Node. The NodeClass is the class of Node. The BrowseName is the non-localized name for the child. The DataType is the structure of the Value accessible via the Node (only used for Variable NodeClass Nodes) and the TypeDefinition is the ObjectType or VariableType for the child.

The ModellingRule indicates whether a child is Mandatory or Optional. It can also indicate cardinality. Note that the BrowseName is not defined if the cardinality is greater than 1. Figure 11 visually depicts the ObjectType defined in Table 1 along with two instances of the ObjectType.

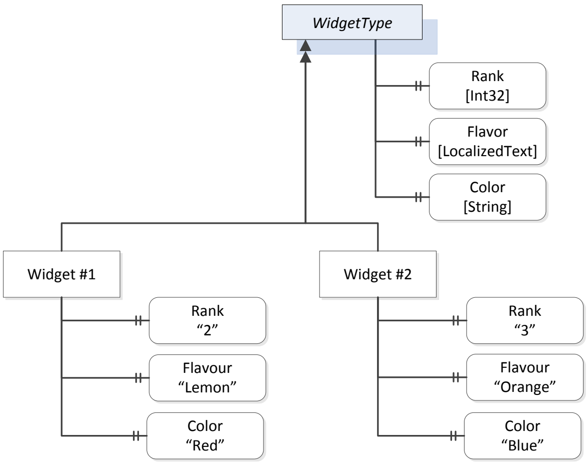


Figure 11 – A Visual Representation of the Sample ObjectType

5 MTConnect in OPC UA

5.1 Base Information

It is expected that MTConnect and OPC UA will continue to evolve their respective standards, in particular as the MTConnect defined information model is revised an OPC UA Client may desire to know the version of the MTConnect model implement by a given server. The information model will always be backwards compatible.

MTConnect defines a Probe request, which is just mapped to an OPC UA Browse. The Probe request returns the devices that are defined in an agent. In OPC UA, a client can at any time browse the address space, and return the detailed structure of the address space, this includes all relationships between devices, their defined types and data types. An OPC UA Client can also subscribe for a ModelChangeEvents changes and SemanticChangeEvents which allow the client to be notified if the address space is modified. ModelChangeEvent are generated by server whenever a new node or reference is added to a server. The Node maybe a new type definition, change to a type definition or a new instance node. SemanticChangeEvents are generated by a server when some aspect of an instance of a node type is change and the change would result in a change that a client should be aware of. An example of a SemanticChangeEvent would be for a change in the engineering Unit assigned to a node.

MTConnect defines a Sample Request command to obtain information from the device. This command can be mapped to an OPC UA read or subscription service. The read service will return the latest value of the requested item. This is appropriate for obtaining the most recent value of a variable. The Subscription service is described the next paragraph. In some cases Historical data access may be required. This allows a client to obtain older values for a device, based on time ranges or other criteria. This feature can be used by a server to allow addition data to be available and could be used to enhance MTConnect Sample Request for cases where more than just the current value is required.

MTConnect describes a Streaming command, which matches to the OPC UA subscription service. This service would allow a client to obtain a steady stream of values associated with a device. The subscription service supports filtering, based on data changes or even aggregates, where an aggregate is a minimum, maximum, average, time in state or any of a long list of aggregates. For a complete list of OPC defined aggregates see [UA Part 11]. It would also allow the return of any OPC Events associated with the device. An OPC Event in MTConnect terms would be a transition of a Condition. Subscription data includes message information that allows a client to determine if a packet was missed or is out of order. The client can request a replay of any missed information or re-sequence the packets. Also included is a keep-alive heart beat that allows for rapid determination of and recovery from communication failures. This functionality is used to ensure the MTConnect Streaming functionality is matched.

MTConnect defines Asset, which becomes a simple addition to the OPC UA information model that describes the MTConnect devices. MTConnect defines the structure of many standard devices, components or Asset. The remainder of this document will provide a mapping of the information model that is described by mapping these MTConnect items into an OPC UA information model.

MTConnect error codes will be mapped to OPC UA error codes where appropriate. The OPC UA server has a multilevel error reporting capability. A Service can include error information, as well as the individual data items that are being reported as part of the service.

As discussed in the overview of OPC UA, all companion specifications usually define a Namespace. For this companion specification all TypeDefinitions and BrowseNames defined by this specification are qualified by the MTConnect Namespace (“urn:mtconnect.com:MTConnectDevices:1.2”) unless stated otherwise.

The description associated with MTConnect items is always mapped to the Description attribute available in OPC UA.

The Name from MTConnect is always mapped to the BrowseName in OPC UA.

The NativeName from MTConnect will be the DisplayName, but if a NativeName is not provided then the BrowseName will be used as the default DisplayName.

The UUID from MTConnect can be mapped to the NodeId in OPC UA. The Namespace associated with MTConnect will assure that the NodeId is unique.

5.2 Additional Subscription information

A DataChange Subscription can be created with the following steps:

- Browse the Address Space and read the NodeIds for the DataItems of interest;
- Create a Subscription;
- Create a MonitoredItem for each DataItem of interest;

As an alternative to browsing the address space, a browse path from a starting node can be created using the type definitions and the TranslateBrowsePathsToNodeIds service can be called to obtain the actual NodeIds. Figure 12 provides an example of a data item that is for a know type. The illustrated path is the same for any instance of a boiler object.

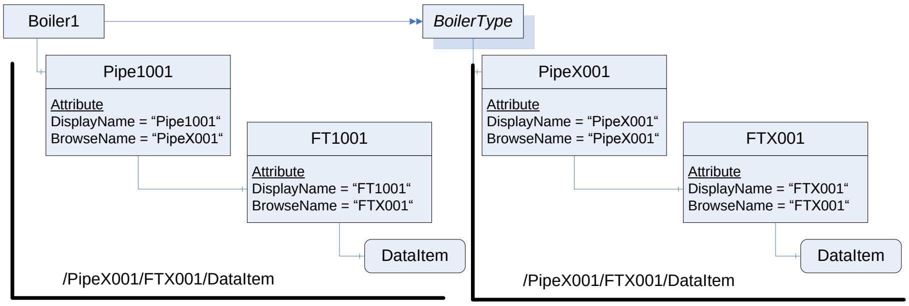


Figure 12 - Browse Path Example

Each Notification has the current value for a DataItem, the Timestamp and StatusCode.

The MonitoredItem allows the Client to specify a different SamplingInterval and QueueSize for each DataItem. The MinimumSamplingInterval attribute for each DataItem specifies the fastest supported SamplingInterval. The PublishingInterval for a Subscription controls how frequently Notifications are returned to the Client. If this value is longer than SamplingInterval then the Server will buffer changes and return multiple Notifications in one message.

An OPC Event Subscription can be created with the following steps:

- Browse the Address Space and read the NodeIds for the Devices or Components of interest;
- Create a Subscription;
- Create a MonitoredItem for each Device or Component of interest;
- Select the fields to return for each event, where the complete list of available fields is defined as part of the OPC UA EventType definition;
- Specify additional filter criteria for the events to return;

Events in OPC UA propagate up the HasNotifier hierarchy. This means subscribing to a parent Node in a hierarchy (i.e. an MTConnect Device) will request events for all Components under that Node. If a Client wants to receive all events for all devices it can subscribe to the Server node.

Each Condition Event is a snapshot of the *MTConditionType* Object. The Client must select the fields from the Condition that it wishes to receive in the update by specifying the BrowsePath (a sequence of BrowseNames).

Table 2 provides an example of an Event produced from the following Condition element:

<Fault type="MOTION_PROGRAM" dataitemid="cc2" sequence="25" qualifier="HIGH" nativeCode="PR1123" timestamp="...">Syntax error on line 107</Fault>

Table 2 – An Example Condition Event

BrowsePath	Value	Notes
<blank>	<server defined>	Identifies the Condition Node in the Address space.
EventId	<server defined>	Equivalent to the sequence number but an opaque value.
ConditionName	MotionProgram	From the name attribute,
ConditionClassName	MotionProgram	From the type attribute.
Message	Syntax error on line 107	The contents of the Condition element.
Time	…	The timestamp attribute.
Severity	700	Inferred from the current state.
CurrentState	Fault	From the Name.
ActiveState/Id	True	Inferred from the Name.
LimitState/CurrentState	High	From the qualifier attribute.
NativeCode	PR1123	From the NativeCode attribute.

A Client can request any or all of the fields defined by the children of the *MTConditionType ObjectType.* OPC UA Event Filters allow for SQL like filtering on events. For example, the following filter would select only those events for MOTION_PROGRAMS in the Fault state:

(ConditionClassName = ‘MotionProgram’) AND (CurrentState = ‘Fault’)

OPC UA does not define a display syntax for filters. Instead it provides a generic structure that stores a pre-parsed expression tree. Client applications are free to use whatever display syntax they feel is appropriate. Some clients may provide a GUI that builds filters; other may parse an SQL like syntax for filters.

5.3 MTDevices

Comment [PEH1]: Add a real world instance example of the type to help illustrate the possible nesting / overall structure

-Waiting for John

Companion specifications usually define a type system, but they can also define a basic structure for the instance space. This basic structure can help clients that wish to translate from an OPC UA address space to an MTConnect system.

The top level container for an MTConnect Agent is the MTDevices element. This element is an Object of MTDevicesType, where MTDevicesType is a subtype of FolderType. The Type is defined in Figure 13 illustrates the top of the address space for an MTConnect enabled OPC UA Server.

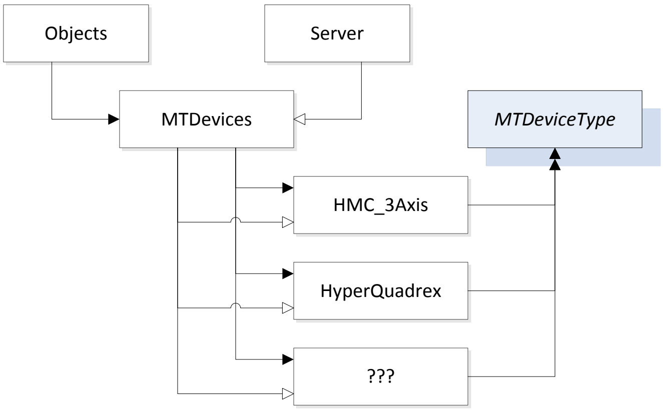


Figure 13 – MTConnect Devices in a UA Address Space

Figure 13 depicts two separate hierarchies. The Organizes hierarchy from Objects Node is used to discover the structure of the devices. The HasNotifier hierarchy from the Server Node is used when subscribing to events (see discussion later in document).

Table 3 – *MTDevicesType* Definition

Attribute	**Value3**				
BrowseName	MTDevicesType				
IsAbstract	False				
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Is a Subtype of UA: FolderType					
HasComponent	Variable	MTConnectVersion	String	BaseDataVariableType	Optional
HasComponent	Variable	OPCUAMappingDate	DateTime	BaseDataVariableType	Optional
HasComponent	Variable	OPCUAVersion	String	BaseDataVariableType	Optional

For an example of what a device may look like, see Annex A.

5.4 MTDeviceType

An MTConnect Device is the container for all of the components for a piece of equipment.

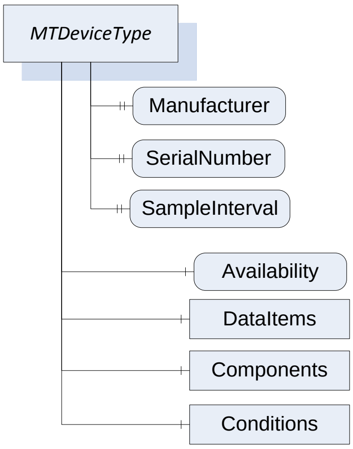


Figure 14 – MTDeviceType

Each Device is mapped to an instance of an Object Node with a TypeDefinition that is *MTDeviceType* or one of its Subtypes. The *MTDeviceType* ObjectType is defined in Table 4.

Table 4 – *MTDeviceType* Definition

Attribute	**Value3**				
BrowseName	MTDeviceType				
IsAbstract	False				
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Is a Subtype of UA: BaseObjectType					
HasComponent	Variable	Availability	String	DataItemType	Mandatory
HasProperty	Variable	Manufacturer	String	PropertyType	Mandatory
HasProperty	Variable	SerialNumber	String	PropertyType	Mandatory
HasProperty	Variable	Configuration	String	PropertyType	Optional
HasProperty	Variable	SampleInterval	Duration	PropertyType	Optional
HasComponent	Object	DataItems		FolderType	Mandatory
HasComponent	Object	Components		FolderType	Mandatory
HasComponent	Object	Conditions		FolderType	Optional

The DataItems Object is a container for the MTConnect Sample and Event DataItems supported by the Device. This folder may be empty if no DataItems exist for this device.

The Components Object is a container for the MTConnect components supported by the Device. This folder may be empty if no components exist for this device.

The Conditions Object is a container for the MTConnect Condition DataItems supported by the Device. If the folder does not exist the device does not support any Conditions.

The Availability Node is the target of a HasComponent reference from the Device and the DataItems folder. This allows clients to treat it as any other DataItem.

The Configuration Property can contain an additional description that is related to configuration information.

The SampleInterval defines the interval at which data is being provided from the device (if the device is being polled). If the OPC UA server is on the device then this parameter can be omitted.

The value of the Manufacturer property is the ‘manufacturer’ attribute;

The value of the SerialNumber property is the 'serialNumber' attribute;

Any Components of the Device are added as targets of HasComponent references.

Any Components of the Device that are Objects are added as targets of HasNotifier references

5.5 MTComponentType

5.5.1 Design MTComponentType

An MTConnect Component element is the container for Machine Tool information that can be accessed via MTConnect. Each Component is mapped to an instance of an Object Node with a TypeDefinition that is *MTComponentType* or one of its Subtypes.

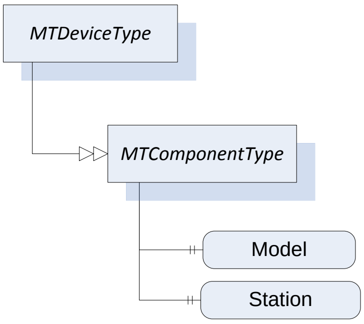


Figure 15 – MTDeviceType

The *MTComponentType* ObjectType is defined in Table 5.

Table 5 – *MTComponentType* Definition

Attribute	**Value**				
BrowseName	MTComponentType				
IsAbstract	False				
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Is a Subtype of MTDeviceType					
HasProperty	Variable	Model	String	PropertyType	Optional
HasProperty	Variable	Station	String	PropertyType	Optional

The MTConnect specification defines many standard Components, but it is expected that additional Subtype of component will be defined by vendors. The following design rules should be followed when creating an Object Node from an MTConnect Component element:

- The BrowseName is the 'name' attribute;
- The Description is the CDATA from the 'Description' element;
- The TypeDefinition is the NodeId of the *MTComponentType* identified by the Name;
- Any DataItems are added as children of the DataItems or Conditions folders;
- Any sub-Components are added as targets of HasComponent and HasNotifier references;

The Name of a Component element identifies the Subtype of *MTComponentType.* If this Name is not already recognized by the UA Server it can create a new ObjectType Node by following these design rules:

- The BrowseName is the Name with 'MTComponentType' appended;

- Add a HasSubtype reference from *MTComponentType;*

The NodeId of the dynamically created ObjectType is server dependent. This specification will define NodeIds for the Component types known at the time this document was written.

In the following section some standard subtype of components are defined.

5.5.2 MTAxesType

From MTConnect, Axes is the root of all device components that have linear or rotational motion. Currently there are only Linear and Rotary axes supported and when axes are defined the Axes component **MUST** contain at least one Linear or Rotary axis. The Linear axes **MUST** be named X, Y, Z with numbers appended for additional axes in the same plane, for example X2, Y2, and Z2 are the secondary axes to X, Y, and Z. Rotary axes **MUST** be named A, B, and C and rotate around the X, Y, and Z axes respectively. As with the linear axes, a number **MUST** be appended for additional axes in the same plane. The Axis is illustrated in Figure 16

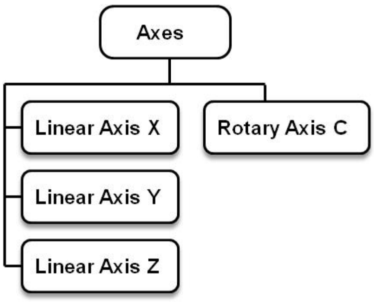


Figure 16 - Axes Illustration

The MTAxesType is defined in Table 6

Table 6 - MTAxesType

Attribute	Value				
BrowseName	MTAxesType				
IsAbstract	False				
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Is a Subtype of MTDeviceType					
HasProperty	Variable	LinearAxesX	String	PropertyType	Optional
HasProperty	Variable	LinearAxesY	String	PropertyType	Optional
HasProperty	Variable	LinearAxesZ	String	PropertyType	Optional
HasProperty	Variable	RotaryAxesA	String	PropertyType	Optional
HasProperty	Variable	RotaryAxesB	String	PropertyType	Optional
HasProperty	Variable	RotaryAxesC	String	PropertyType	Optional

A linear axis represents the movement of a physical device, or a portion of a device, in a straight line. Movement may be in either a positive or negative direction.

An axis whose function is to provide rotary motion may function as a continuous rotation (i.e. spindle mode), continuous-path contour cutting in a rotary motion (i.e. contouring), or repositioning (i.e. indexing) different faces of the part. As such, a rotary axis **MUST** operate in one of the three following modes: `SPINDLE`, `INDEX`, or `CONTOUR`.

5.5.3 MTControllerType

From MTConnect, The Controller component represents an intelligent device. Examples include a CNC (Computer Numerical Control) or PAC (Programmable Automation Control) which may be referred to as a Motion Control or General Purpose Motion Control. The Controller provides information regarding the execution of a control program and the execution state of the device. There are no required subcomponents of the Controller.

Table 7 - MTControllerType

Attribute	Value				
BrowseName	MTControllerType				
IsAbstract	False				
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Is a Subtype of MTDeviceType					

5.5.4 MTDoorType

From MTConnect, This component represents a door closure that can be opened or closed. It MUST have a DataItem called DoorState to indicate if it is opened, closed or unlatched. A device may contain multiple door components.

Table 8 - MTDoorType

Attribute	Value				
BrowseName	MTDoorType				
IsAbstract	False				
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Is a Subtype of MTDeviceType					
HasProperty	Variable	DoorState	String	PropertyType	Optional

5.5.5 MTActuatorType

From MTConnect, An Actuator is a device for moving or controlling a mechanism or system. It takes energy, usually transported by air, electric current, or liquid and converts it into some kind of motion. An Actuator may be a Component of a Device or it may be a subcomponent of a parent Component.

Table 9 - MTActuatorType

Attribute	Value				
BrowseName	MTActuatorType				
IsAbstract	False				
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Is a Subtype of MTDeviceType					
HasProperty	Variable	LinearAxesX	String	PropertyType	Optional

5.6 MT Data Items

An MTConnect DataItem describes a piece of information that can be collected from a component. Sample and Event DataItems are mapped to an instance of a Variable Node with a TypeDefinition that is a Subtype of either *MTSampleDataItemType* or *MTEventDataItemType.*

Each MTConnect DataItem defines a category, type and Subtype attributes. These three values uniquely identify a Subtype. The mappings for SAMPLE and EVENT categories are:

- SAMPLE: MTSampleDataItemType
- EVENT: MTEventDataItemType
- CONDITION MTConditionDataItemType

The mappings for CONDITION category is discussed in the section on Conditions below.

The type is mapped to a Subtype of one of the category types. The following design rules are followed when creating the VariableType Node from type attribute:

- The BrowseName is camel case form of type with ‘ItemType’ appended;
- Add a HasSubtype reference from VariableType for the category;

‘Camel Case’ refers to the convention where multiple words are combined to make a symbol by removing spaces and capitalizing the first letter of each word (e.g. ‘PathFeedrate’).

The Subtype is mapped to a Subtype of the VariableType for the type. The following design rules are followed when creating the VariableType Node from Subtype attribute:

- The BrowseName is camel case form of Subtype with the BrowseName of the parent type appended (e.g. ActualRotaryVelocityType);
- Add a HasSubtype reference from VariableType for the type;

The NodeId of the dynamically created VariableType are server dependent. This specification will define NodeIds for the DataItem types known at the time this document was written.

5.7 Sample Data Items

5.7.1 MTSampleDataItemType

The *MTSampleDataItemType* VariableType is defined in Table 10.

Table 10 – MTSampleDataItemType Definition

Attribute	**Value**				
BrowseName	MTSampleDataItemType				
IsAbstract	False				
DataType	Number				
ValueRank	-1 (-1 = Scalar)				
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Inherit the *Properties* of the *AnalogItemType* which is defined in [UA Part 8]					
HasProperty	Variable	CoordinateSystem	CoordinateSystem TypeEnum	PropertyType	Optional
HasProperty	Variable	NativeUnits	EUInformation	PropertyType	Optional
HasProperty	Variable	NativeScale	Float	PropertyType	Optional
HasProperty	Variable	SampleInterval	Duration	PropertyType	Optional

If the statistic information from MTConnect is provided, it is mapped as a ComputedBy reference to an instance of an AggregateFunctionType defined in [UA Part 13]. If one of the standard aggregate instance definition defined by OPC does not match the MTConnect aggregate then a custom aggregate instance shall be defined.

If the Source information from MTConnect is provided, it is mapped as a SourceOf reference to the instance that is the Source element. This reference identifies the Component, subcomponent or DataItem where the physical connection to the data source originates.

In OPC UA time series data is stored as historical values or can be queue as part of a subscription to a client, but all of the value stored in an OPC UA address space are the current value for an item.

The Units are mapped to the engineering units provided by the AnalogItemType

The MTConnect significantDigits attribute is mapped to the ValuePrecision property of the AnalogItemType.

The MTConnect Constraints are provided; the Minimum and Maximum are mapped to the AnalogItemType instrument range.

The CoordinateSystem properties correspond to the attribute definitions in [MT Part 2].

The NativeUnits are mapped to the OPC UA EUInformation dataType.

The NativeScale properties correspond to the attribute definitions in [MT Part 2].

The SampleInterval defines the interval at which data is being provided from the device (if the device is being polled). If the OPC UA server is on the device then this parameter can be omitted.

Subtypes of MTSampleDataItemType may be created to allow easier definition of a specific type, i.e. one with a fixed engineering unit, description. The following sections describe some predefined subtypes.

5.7.2 AccelerationType

The acceleration of the component, has engineering unit fixed to Millimeter/Second^2.

Table 11 - AccelerationType

Attribute		Value			
BrowseName		AccelerationType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.3 AccumulatedTimeType

The AccumulatedTimeType is the non-contiguous duration of an event in seconds.

Table 12 - AccumulatedTimeType

Attribute		Value			
BrowseName		AccumulatedTimeType			
IsAbstract		False			
DataType		Duration			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.4 AmperageType

The current a component is drawing measured in Ampere.

Table 13 - AmperageType

Attribute		Value			
BrowseName		AmperageType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.5 AngleType

The angular position of a component relative to a parent, measured in degree.

Table 14 - AngleType

Attribute		Value			
BrowseName		AngleType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.6 AngularAccelerationType

The angular acceleration of the component measured in Degree/Second^2.

Table 15 - AngularAccelerationType

Attribute		Value			
BrowseName		AngularAccelerationType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.7 AngularVelocityType

The angular velocity of the component measured in Degree/Second.

Table 16 - AngularVelocityType

Attribute		Value			
BrowseName		AngularVelocityType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.8 AxisFeedrateType

The feed rate of a linear axis measured in Millimeter/Second.

Table 17 - AxisFeedrateType

Attribute		Value			
BrowseName		AxisFeedrateType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.9 ConcentrationType

The Concentration is the percentage of one component within a mixture of components measured in percent.

Table 18 - ConcentrationType

Attribute		Value			
BrowseName		ConcentrationType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.10 ConductivityType

The Conductivity is the ability of a material to conduct electricity measured in Siemens/Meter.

Table 19 - ConductivityType

Attribute		Value			
BrowseName		ConductivityType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.11 DisplacementType

The linear displacement measured in millimeters.

Table 20 - DisplacementType

Attribute		Value			
BrowseName		DisplacementType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.12 ElectricalEnergyType

Electrical energy consumed by a component measured in Watt-Second.

Table 21 - ElectricalEnergyType

Attribute		Value			
BrowseName		ElectricalEnergyType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.13 FillLevelType

The measurement of the amount of a substance remaining compared to the planned maximum amount of that substance measure in percent.

Table 22 - FillLevelType

Attribute		Value			
BrowseName		FillLevelType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.14 FlowType

The rate of flow of a fluid measured in Liter/Second.

Table 23 - FlowType

Attribute		Value			
BrowseName		FlowType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.15 FrequencyType

The frequency measured in hertz.

Table 24 - FrequencyType

Attribute	Value
BrowseName	FrequencyType
IsAbstract	False

DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.16 LinearForceType

The magnitude of a push or pull introduced by an actuator or exerted on an object measured in Newton.

Table 25 - LinearForceType

Attribute		Value			
BrowseName		LinearForceType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.17 LoadType

The measurement of the percentage of the standard rating for a device measured in percent.

Table 26 - LoadType

Attribute		Value			
BrowseName		LoadType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.18 MassType

The measurement of the mass of an object or the amount of material measured in kilogram

Table 27 - MassType

Attribute		Value			
BrowseName		MassType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.19 PathFeedrateType

The feed rate of the tool path measured in Millimeter/Second.

Table 28 - PathFeedrateType

Attribute		Value			
BrowseName		PathFeedrateType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.20 PathPositionType

The 3 dimensional position for tool tip in the path (X Y Z) measured in Millimeter_3D.

Table 29 - PathPositionType

Attribute		Value			
BrowseName		PathPositionType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.21 PHType

The measure of acidity or alkalinity measured in PH.

Table 30 - PHType

Attribute		Value			
BrowseName		PHType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.22 PositionType

The position of the component measured in Millimeter.

Table 31 - PositionType

Attribute		Value			
BrowseName		PositionType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.23 PowerFactorType

The PowerFactorType is the measurement of the ration of real power flowing to a load to the apparent power in that AC circuit measured in percent.

Table 32 - PowerFactorType

Attribute		Value			
BrowseName		PowerFactorType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.24 PressureType

The pressure measured in Pascal

Table 33 - PressureType

Attribute		Value			
BrowseName		PressureType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.25 ResistanceType

The electrical Resistance measured in Ohm

Table 34 - ResistanceType

Attribute		Value			
BrowseName		ResistanceType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.26 RotationalVelocityType

The rotational velocity of the component in RPM

Table 35 - RotationalVelocityType

Attribute		Value			
BrowseName		RotationalVelocityType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.27 SoundPressureType

The sound pressure measured in Decibel

Table 36 - SoundPressureType

Attribute		Value			
BrowseName		SoundPressureType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.28 StrainType

The strain is the amount of deformation per unit length of an object when a load is applied measured in percent

Table 37 - StrainType

Attribute		**Value**			
BrowseName		StrainType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of MTSampleDataItemType					

5.7.29 TemperatureType

The temperature measured in Celsius.

Table 38 - TemperatureType

Attribute	Value
BrowseName	TemperatureType
IsAbstract	False

DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.30 TiltType

The tilt is the measurement of angular displacement measured in Micro-Radian.

Table 39 - TiltType

Attribute		Value			
BrowseName		TiltType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.31 TorqueType

The torque is the turning force exerted on an object by an object measured in Newton-Meter

Table 40 - TorqueType

Attribute		Value			
BrowseName		TorqueType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.32 VelocityType

The velocity of the component measured in Millimeter/Second

Table 41 - VelocityType

Attribute		Value			
BrowseName		VelocityType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.33 ViscosityType

The Viscosity is the measurement of a fluid’s resistance to flow measured in Pascal-Second

Table 42 - ViscosityType

Attribute		Value			
BrowseName		ViscosityType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.34 VoltageType

The voltage measured in Volts

Table 43 - VoltageType

Attribute	Value

BrowseName		VoltageType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.35 VoltAmpereType

The measurement of the percentage of the standard rating for a device measured in percent.

Table 44 - VoltAmpereType

Attribute		Value			
BrowseName		VoltAmpereType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.36 VoltAmpereReactiveType

The measurement of the percentage of the standard rating for a device measured in percent.

Table 45 - VoltAmpereReactiveType

Attribute		Value			
BrowseName		VoltAmpereReactiveType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.7.37 WattageType

The measurement of power consumed or dissipated by an electric circuit or device measured in Watts

Table 46 - WattageType

Attribute		Value			
BrowseName		WattageType			
IsAbstract		False			
DataType		Number			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItemType					

5.8 Event DataItems

5.8.1 MTEventDataItemType

The *MTEventDataItemType* VariableType is defined in Table 47.

Table 47 – ***MTEventDataItemType*** **Definition**

Attribute	**Value**				
BrowseName	MTEventDataItemType				
IsAbstract	False				
DataType	BaseDataType				
ValueRank	-1 (-1 = Scalar)				
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of DiscreteItemType defined in [UA Part 8].					
HasProperty	Variable	CoordinateSystem	CoordinateSystemType Enum	PropertyType	Optional
HasProperty	Variable	SampleInterval	Duration	PropertyType	Optional

If the Source information from MTConnect is provided, it is mapped as a SourceOf reference to the instance that is the Source element. This reference identifies the Component, subcomponent or DataItem where the physical connection to the data source originates.

In OPC UA time series data is stored as historical values or can be queue as part of a subscription to a client, but all of the value stored in an OPC UA address space are the current value for an item.

The Constraints defined in [MT Part 2] are mapped to OPC UA DiscreteItemType and subtypes of it, where each subtype defines a specific enumeration or dataType.

The CoordinateSystem properties correspond to the attribute definitions in [MT Part 2].

The SampleInterval defines the interval at which data is being provided from the device (if the device is being polled). If the OPC UA server is on the device then this parameter can be omitted.

This variable type is expected to be subtyped, where the sub types will further define the dataType of the item. In many case this dataType will be an enumeration DataType. In some case additional properties or variable will be provided.

5.8.2 ActiveAxesType

The list of axes in use (array). The set of axes associated with a path that the controller is controlling. If this data item is not provided, it will be assumed the controller is controlling all axes.

Table 48 - ActiveAxesType

Attribute	Value				
BrowseName	ActiveAxesType				
IsAbstract	False				
DataType	String				
ValueRank	1				
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.8.3 ActuatorStateType

The availability of the component

Table 49 - ActuatorStateType

Attribute	**Value**				
BrowseName	ActuatorStateType				
IsAbstract	False				
DataType	ActuatorStateTypeEnum				
ValueRank	-1 (-1 = Scalar)				
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the MTEventDataItemType					

5.8.4 AvailabilityType

The availability of the component

Table 50 – AvailabilityType

Attribute		Value			
BrowseName		AvailabilityType			
IsAbstract		False			
DataType		AvailabilityTypeEnum			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.8.5 AxesCouplingType

The way the axes are associated

Table 51 - AxesCouplingType

Attribute		Value			
BrowseName		AxesCouplingType			
IsAbstract		False			
DataType		AxesCouplingTypeEnum			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.8.6 BlockType

The program code

Table 52 - BlockType

Attribute		Value			
BrowseName		BlockType			
IsAbstract		False			
DataType		String			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.8.7 ClampStateType

The clamp status

Table 53 – ClampStateType

Attribute		Value			
BrowseName		ClampStateType			
IsAbstract		False			
DataType		ClampStateTypeEnum			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.8.8 ControllerModeType

CNC mode state

Table 54 -ControllerModeType

Attribute		Value			
BrowseName		ControllerModeType			
IsAbstract		False			
ValueRank		-1 (-1 = Scalar)			
DataType		ControllerModeTypeEnum			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule

Subtype of the MTEventDataItemType

5.8.9 CoupledAxesType

The list of associated axes

Table 55 - CoupledAxesType

Attribute		Value			
BrowseName		CoupledAxesType			
IsAbstract		False			
DataType		String			
ValueRank		1			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.8.10 DirectionType

The direction of rotation

Table 56 - DirectionType

Attribute		Value			
BrowseName		DirectionType			
IsAbstract		False			
ValueRank		-1 (-1 = Scalar)			
DataType		DirectionTypeEnum			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.8.11 DoorStateType

The status of the door

Table 57 - DoorStateType

Attribute		Value			
BrowseName		DoorStateType			
IsAbstract		False			
ValueRank		-1 (-1 = Scalar)			
DataType		DoorStateTypeEnum			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.8.12 EmergencyStopType

Emergency Stop status

Table 58 - EmergencyStopType

Attribute		Value			
BrowseName		EmergencyStopType			
IsAbstract		False			
DataType		EmergencyStopTypeEnum			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Is a type EmergencyStopType					

5.8.13 ExecutionType

Program execution events

Table 59 - ExecutionType

Attribute	Value
BrowseName	ExecutionType
IsAbstract	False

ValueRank		-1 (-1 = Scalar)			
DataType		ExecutionTypeEnum			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.8.14 LineType

The program's line of execution

Table 60 - LineType

Attribute		Value			
BrowseName		LineType			
IsAbstract		False			
DataType		UInt16			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.8.15 MessageType

A generic message

Table 61 - MessageType

Attribute		Value			
BrowseName		MessageType			
IsAbstract		False			
DataType		String			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.8.16 PalletIdType

The current pallet Identifier

Table 62 – PalletIdType

Attribute		Value			
BrowseName		PalletIdType			
IsAbstract		False			
DataType		String			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.8.17 PartCountType

An event that indicates the number of parts

Table 63 - PartCountType

Attribute		Value			
BrowseName		PartCountType			
IsAbstract		False			
DataType		UInt16			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.8.18 PartIdType

The current Tool Identifier

Table 64 - PartIdType

Attribute	Value

BrowseName		PartIdType			
IsAbstract		False			
DataType		String			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.8.19 PathModeType

The actuator state of the component

Table 65 - PathModeType

Attribute		Value			
BrowseName		PathModeType			
IsAbstract		False			
ValueRank		-1 (-1 = Scalar)			
DataType		PathModeTypeEnum			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.8.20 PowerStateType

The components power state

Table 66 - PowerStateType

Attribute		Value			
BrowseName		PowerStateType			
IsAbstract		False			
ValueRank		-1 (-1 = Scalar)			
DataType		PowerStateTypeEnum			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.8.21 ProgramType

The programs file name or identifier

Table 67 - ProgramType

Attribute		Value			
BrowseName		ProgramType			
IsAbstract		False			
DataType		String			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.8.22 RotaryModeType

The function of the rotary axis

Table 68 - RotaryModeType

Attribute		Value			
BrowseName		RotaryModeType			
IsAbstract		False			
DataType		RotaryModeTypeEnum			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.8.23 ToolAssetIdType

The unique tool Identifier as referenced in MTConnect specification on an assets

Table 69 - ToolAssetIdType

Attribute		Value			
BrowseName		ToolAssetIdType			
IsAbstract		False			
DataType		String			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.8.24 ToolNumberType

The identifier of a tool provided by a device controller

Table 70 - ToolNumberType

Attribute		Value			
BrowseName		ToolNumberType			
IsAbstract		False			
DataType		Uint16			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.8.25 WorkHoldingIdType

The current work holding Identifier

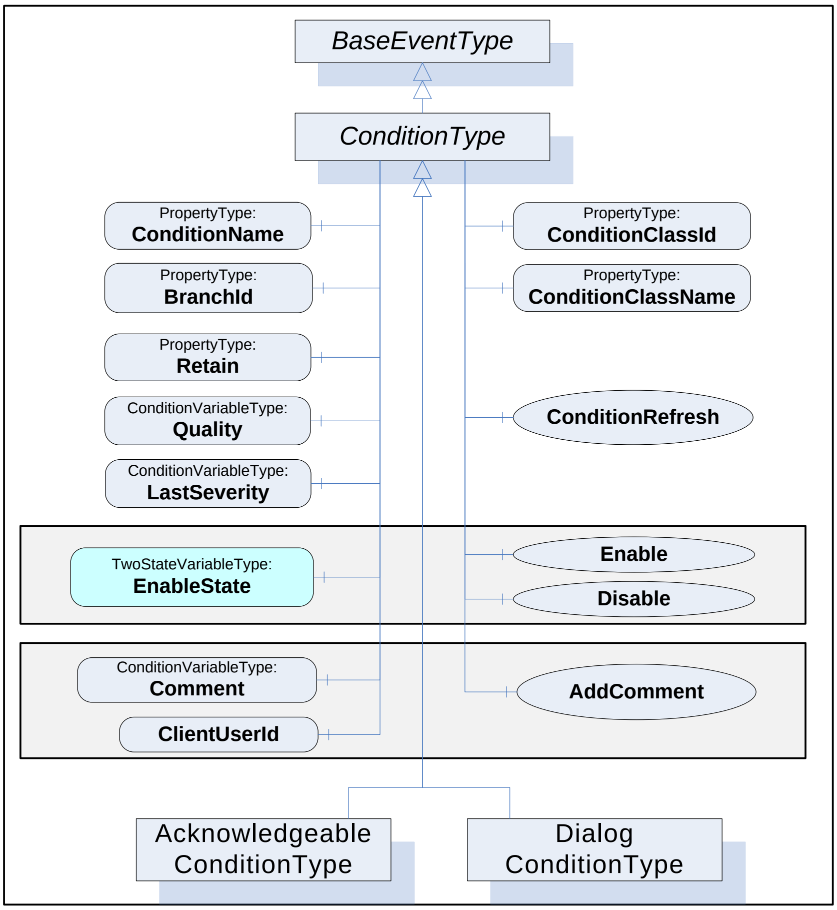
Table 71 - WorkHoldingIdType

Attribute		Value			
BrowseName		WorkHoldingIdType			
IsAbstract		False			
DataType		String			
ValueRank		-1 (-1 = Scalar)			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the MTEventDataItemType					

5.9 Conditions

5.9.1 MTConditionType

MTConnect Conditions are DataItems which report the state of health or functionality for a component. Semantically they are equivalent to Conditions in OPC UA which are Object Nodes that produce Events. OPC UA Conditions also appear in the address space and can be accessed like any other Variable.


BaseEventType
ConditionType
PropertyType: ConditionName
PropertyType: BranchId
PropertyType: Retain
ConditionVariableType: Quality
ConditionVariableType: LastSeverity
PropertyType: ConditionClassId
PropertyType: ConditionClassName
ConditionRefresh
TwoStateVariableType: EnableState
Enable
Disable
ConditionVariableType: Comment
ClientUserId
AddComment
Acknowledgeable ConditionType
Dialog ConditionType

Each MTConnect Condition is represented by an Object Node with a TypeDefinition *MTConditionType.* The *MTConditionType* ObjectType is defined in Table 72.

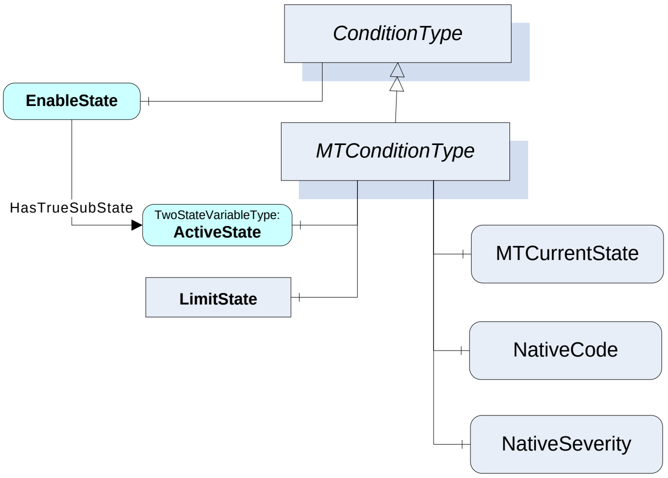


Table 72 – *MTConditionType* Definition

Attribute	**Value**				
BrowseName	MTConditionType				
IsAbstract	False				
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the *ConditionType* which is defined in [UA Part 9].					
HasComponent	Variable	MTCurrentState	String	BaseDataVariableType	Mandatory
HasComponent	Variable	ActiveState	LocalizedText	TwoStateVariableType	Mandatory
HasComponent	Object	LimitState		ExclusiveLimitStateMachineType (defined in [UA Part 9]	Optional
HasProperty	Variable	NativeCode	String	PropertyType	Optional
HasProperty	Variable	NativeSeverity	String	PropertyType	Optional

The NativeSeverity and NativeCode properties correspond to the attribute definitions in [MT Part 3].

The MTCurrentState is the MTConnect Condition state: Unavailable, Normal, Warning or Fault.

The Severity property (inherited from ConditionType) should set based on the value of this Variable according to these rules:

- Unavailable: 1
- Normal: <= 100
- Warning: > 100 && <=500
- Fault: > 500

The ActiveState Variable indicates whether the condition is in a state that requires attention. This Variable is ‘Active’ if the MTConnect Condition state is Warning or Fault. If the MTConnect Condition state is Normal or Unavailable this Variable is ‘Inactive’.

The LimitState is used to specify the MTConnect Condition qualifier attribute. If the qualifier is ‘HIGH’ the LimitState/CurrentState Variable is ‘High’. If the qualifier is ‘LOW’ then the LimitState/CurrentState Variable is ‘Low’.

The Message property (inherited from BaseEventType) contains contents of the MTConnect Condition element. If no contents were provided the Server shall choose a suitable default message.

The BrowseName of the Condition is value of the name attribute for the DataItem. If the name attribute is missing the camel case version of the type attribute is used instead.

Many MTConnect Conditions may be related to an instance of *MTSampleDataItemType* which provides access to the value that can trigger the Condition. If these relationships are known the Server should provide a HasCondition reference from the Variable for the DataItem to the corresponding Condition Objects. Figure 17 depicts an example of a component with a Position DataItem with an associated Condition.

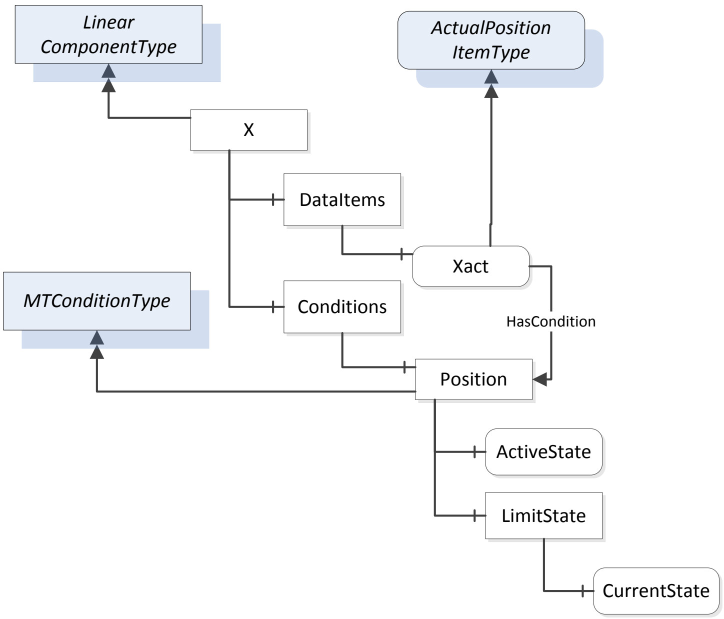


Figure 17 – MTConnect Conditions in the UA Address Space

A Condition communicates the device’s health and ability to function. It can be one of Unavailable, Normal, Warning, or Fault and there can be multiple active conditions at one time; whereas a sample or event can only have a single value at one point in time.

5.9.2 AccelerationConditionType

The AccelerationConditionType

Table 73 - AccelerationConditionType

Attribute		Value			
BrowseName		AccelerationConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.3 Accumulated_TimeConditionType

The Accumulated_TimeConditionType

Table 74 - Accumulated_TimeConditionType

Attribute		Value			
BrowseName		Accumulated_TimeConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.4 ActuatorConditionType

The ActuatorConditionType

Table 75 – ActuatorConditionType

Attribute		Value			
BrowseName		ActuatorConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.5 AmperageConditionType

The AmperageConditionType

Table 76 - AmperageConditionType

Attribute		Value			
BrowseName		AmperageConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.6 AngleConditionType

The AngleConditionType

Table 77 - AngleConditionType

Attribute		Value			
BrowseName		AngleConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.7 Angular-AccelerationConditionType

The Angular-AccelerationConditionType

Table 78 - Angular-AccelerationConditionType

Attribute		Value			
BrowseName		Angular-AccelerationConditionType			
IsAbstract		false			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.8 Angular_VelocityConditionType

The Angular_VelocityConditionType

Table 79 - Angular_VelocityConditionType

Attribute		Value			
BrowseName		Angular_VelocityConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.9 CommunicationsConditionType

The CommunicationsConditionType

Table 80 - CommunicationsConditionType

Attribute		Value			
BrowseName		CommunicationsConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.10 ConcentrationConditionType

The ConcentrationConditionType

Table 81 - ConcentrationConditionType

Attribute		Value			
BrowseName		ConcentrationConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.11 ConductivityConditionType

The ConductivityConditionType

Table 82 - ConductivityConditionType

Attribute		Value			
BrowseName		ConductivityType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.12 Data_RangeConditionType

The Data_RangeConditionType

Table 83 - Data_RangeConditionType

Attribute		Value			
BrowseName		Data_RangeConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.13 DirectionConditionType

The DirectionConditionType

Table 84 - DirectionConditionType

Attribute		Value			
BrowseName		DirectionConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.14 DisplacementConditionType

The DisplacementConditionType

Table 85 - DisplacementConditionType

Attribute	Value
BrowseName	DisplacementConditionType
IsAbstract	False

Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.15 Electrical_EnergyConditionType

The Electrical_EnergyConditionType

Table 86 - Electrical_EnergyConditionType

Attribute		Value			
BrowseName		Electrical_EnergyConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.16 Fill_LevelConditionType

The Fill_LevelConditionType

Table 87 - Fill_LevelConditionType

Attribute		Value			
BrowseName		Fill_LevelConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.17 FlowConditionType

The FlowConditionType

Table 88 - FlowConditionType

Attribute		Value			
BrowseName		FlowConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.18 FrequecncyConditionType

The FrequecncyConditionType

Table 89 - FrequecncyConditionType

Attribute		Value			
BrowseName		FrequecncyConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.19 HardwareConditionType

The HardwareConditionType

Table 90 - HardwareConditionType

Attribute		Value			
BrowseName		HardwareConditionType			
IsAbstract		false			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.20 Linear_ForceConditionType

The Linear_ForceConditionType

Table 91 - Linear_ForceConditionType

Attribute		Value			
BrowseName		Linear_ForceConditionType			
IsAbstract		false			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.21 LoadConditionType

The LoadConditionType

Table 92 - LoadConditionType

Attribute		Value			
BrowseName		LoadConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.22 Logic_ProgramConditionType

The Logic_ProgramConditionType

Table 93 - Logic_ProgramConditionType

Attribute		Value			
BrowseName		Logic_ProgramConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.23 MassConditionType

The MassConditionType

Table 94 - MassConditionType

Attribute		Value			
BrowseName		MassConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.24 Motion_ProgramConditionType

The Motion_ProgramConditionType

Table 95 - Motion_ProgramConditionType

Attribute		Value			
BrowseName		Motion_ProgramConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.25 Path_FeedrateConditionType

The Path_FeedrateConditionType

Table 96 - Path_FeedrateConditionType

Attribute		Value			
BrowseName		Path_FeedrateConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.26 Path_PositionConditionType

The Path_PositionConditionType

Table 97 – Path_PositionConditionType

Attribute		Value			
BrowseName		Path_PositionConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.27 PHConditionType

The PHConditionType

Table 98 - PHConditionType

Attribute		Value			
BrowseName		PHConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.28 PositionConditionType

The PositionConditionType

Table 99 - PositionConditionType

Attribute		Value			
BrowseName		PositionConditionType			
IsAbstract		false			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.29 Power_FactorConditionType

The Power_FactorConditionType

Table 100 - Power_FactorConditionType

Attribute		Value			
BrowseName		Power_FactorConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.30 PressureConditionType

The PressureConditionType

Table 101 – PressureConditionType

Attribute		Value			
BrowseName		PressureConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.31 ResistanceConditionType

The ResistanceConditionType

Table 102 - ResistanceConditionType

Attribute	Value
BrowseName	ResistanceConditionType
IsAbstract	False

Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.32 Rotary_VelocityConditionType

The Rotary_VelocityConditionType

Table 103 - Rotary_VelocityConditionType

Attribute	Value				
BrowseName	Rotary_VelocityConditionType				
IsAbstract	False				
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.33 Sound_LevelConditionType

The Sound_LevelConditionType

Table 104 – Sound_LevelConditionType

Attribute	Value				
BrowseName	Sound_LevelConditionType				
IsAbstract	False				
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.34 StrainConditionType

The StrainConditionType

Table 105 - StrainConditionType

Attribute	Value				
BrowseName	StrainConditionType				
IsAbstract	false				
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.35 SystemConditionType

The SystemConditionType

Table 106 - SystemConditionType

Attribute	Value				
BrowseName	SystemConditionType				
IsAbstract	false				
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.36 TemperatureConditionType

The TemperatureConditionType

Table 107 - TemperatureConditionType

Attribute	Value				
BrowseName	TemperatureConditionType				
IsAbstract	False				
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.37 TiltConditionType

The TiltConditionType

Table 108 - TiltConditionType

Attribute		Value			
BrowseName		TiltConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.38 TorqueConditionType

The TorqueConditionType

Table 109 - TorqueConditionType

Attribute		Value			
BrowseName		TorqueConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.39 VelocityConditionType

The VelocityConditionType

Table 110 - VelocityConditionType

Attribute		Value			
BrowseName		VelocityConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.40 ViscosityConditionType

The ViscosityConditionType

Table 111 - ViscosityConditionType

Attribute		Value			
BrowseName		ViscosityConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.41 VoltageConditionType

The VoltageConditionType

Table 112 - VoltageConditionType

Attribute		Value			
BrowseName		VoltageConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.42 Volt_AmperageConditionType

The Volt_AmperageConditionType

Table 113 – Volt_AmperageConditionType

Attribute		Value			
BrowseName		Volt_AmperageConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.43 VoltAmperageReactiveConditionType

The VoltAmperageReactiveConditionType

Table 114 - VoltAmperageReactiveConditionType

Attribute		Value			
BrowseName		VoltAmperageReactiveConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.9.44 WattageConditionType

The WattageConditionType

Table 115 - WattageConditionType

Attribute		Value			
BrowseName		WattageConditionType			
IsAbstract		False			
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTConditionType					

5.10 Sensor

5.10.1.1 Overview

Sensor is an abstract type component that provides measurement data related to a device or component. A sensor may or may not be integral to a parent device or component – it can function as a device. A sensor can be external to the parent device and can be moved from one device to another. Sensors MAY have their own UUID so they can be tracked throughout their lifetime. Sensors that are not integral to a parent component or device SHALL have a UUID. Figure 18 illustrates the possible layout of sensors

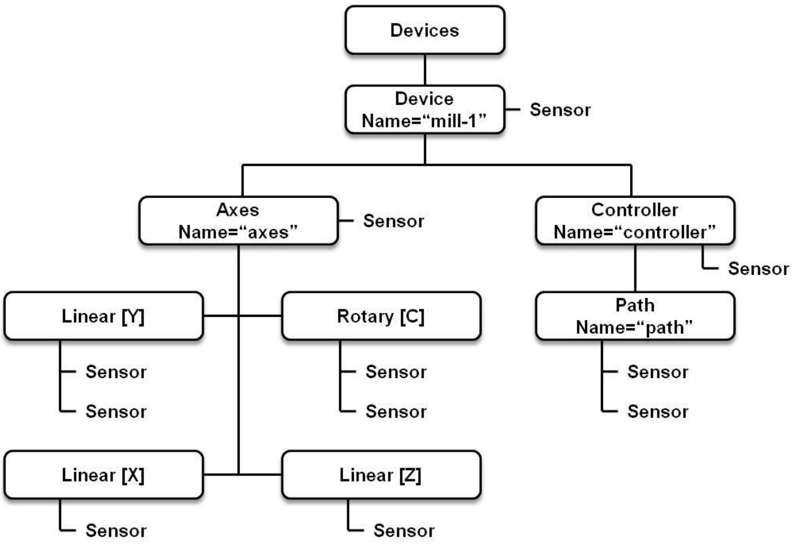


Figure 18: Example Sensors Structure

Sensors are typically comprised of two major components – the sensing element (provides a signal or measured value) and the sensor interface (signal processing, conversion, and communications). In MTConnect, the sensor interface is modeled as a component called Sensor. The sensing element or measured value is modeled as a DataItem. Example: A pressure transducer could be modeled as a Sensor component with a name = Pressure Transducer B and its measured value could be modeled as a DataItem type PRESSURE.

Sensors and sensor DataItems SHOULD be modeled into the MTConnect schema in such a manner that the sensor DataItem is coupled with the component or sub-component to which it is directly associated. For example: If a temperature sensor is monitoring the temperature of the machine’s X axis, it should be modeled as a DataItem TEMPERATURE for the component X AXIS.

The Sensor may provide the signal processing for multiple sensing elements. In this case, the Sensor MAY be modeled as an independent component or sub-component within a device. The sensor may provide additional configuration data and diagnostic information specific to the sensor interface itself or the individual sensing elements. Configuration data is provided in the Component Element Configuration. The sensor configuration data provides information, required maintenance and support of the sensor devices. It also provides configuration data for each sensing element, as follows:

- FirmwareVersion-Version number for the Sensor as specified by the manufacturer
- CalibrationDate-Date upon which the Sensor was last calibrated.
- NextCalibrationDate-Date upon which the Sensor is next scheduled to be calibrated.
- CalibrationInitials-The initials of the person verifying the validity of the calibration data
- Channels-A sensor can be comprised of multiple sensing elements. Each sensing element represents a Channel for the Sensor.

5.10.1.2 ChannelType

The calibration channel

Table 116 - ChannelType

Attribute	Value				
BrowseName	ChannelType				
IsAbstract	False				
Value rank					
Data Type	ChannelNumberDataType				
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
HasComponent	Variable	CalibrationDate	CalibrationDateType	PropertyType	Optional
HasComponent	Variable	NextCalibrationDate	CalibrationDateType	PropertyType	Optional
HasComponent	Variable	CalibrationInitials	CalibrationInitialsType	PropertyType	Optional

In MTConnect the “Description” XML element is mapped to the OPC UA description that is part of the base OPC Node type.

The value of the ChannelType variable if the unique number assigned to a channel

CalibrationDate –the last time the channel was calibrated

NextCalibrationDate – the next schedule time for a calibration

CalibrationInitials – The initials of the person that did the calibration

5.10.1.3 SensorConfigurationType

Calibration data for a sensor

Table 117 - SensorConfigurationType

Attribute		Value			
BrowseName		SensorConfigurationType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the ConfigurationType					

HasComponent	Variable	FirmwareVersion	FirmwareVersionDataType	PropertyType	Optional
HasComponent	Variable	CalibrationDate	CalibrationDateType	PropertyType	Optional
HasComponent	Variable	NextCalibrationDate	CalibrationDateType	PropertyType	Optional
HasComponent	Variable	CalibrationInitials	CalibrationInitialsType	PropertyType	Optional
HasComponent	Variable	Channels	ChannelType	PropertyType	Optional

5.10.1.4 SensorConfigurationType

Calibration data for a sensor

Table 118 - SensorConfigurationType

Attribute		**Value**			
BrowseName		SensorConfigurationType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the ConfigurationType					

5.10.1.5 ConfigurationType

Abstract configuration

Table 119 - ConfigurationType

Attribute		**Value**			
BrowseName		ConfigurationType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Is a type ConfigurationType					

5.10.1.6 SensorType

A sensor, this is not abstract to allow for easy extensibility.

Table 120 - SensorType

Attribute		**Value**			
BrowseName		SensorType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the CommonComponentType					

5.10.1.7 SensorType

A sensor, this is not abstract to allow for easy extensibility.

Table 121 - SensorType

Attribute		**Value**			
BrowseName		SensorType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Is a substitute CommonComponent					

5.10.1.8 SourceType

A native data source

Table 122 - SourceType

Attribute		**Value**			
BrowseName		SourceType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the ItemSourceType					
HasProperty	Variable	componentId	SourceComponentIdDataType	PropertyType	Optional

6 Assets

6.1 Overview

From the MTConnect specification version 1.2 Part 4 "Assets", an Asset is something that is associated with the manufacturing process that is not a component of a device, can be removed without detriment to the function of the device, and can be associated with other devices during their lifecycle. An asset does not have computational capabilities, but may carry information in some media physically attached to the asset. Concrete examples of Assets are things like Cutting Tools, Workholding Systems, and Fixtures. The Cutting Tool is the only asset type covered by the MTConnect Asset standard as of version 1.2.

Asset support the concept of notification, via events for assets that have changed. OPC UA supports two basic event types that are used to reflect changes to the address space:

- ModelChangeEvent (BaseModelChangeEventType and GeneralModelChangeEventType) used to indicate if a node was added to the address space or changed in the address space or deleted from the address space.

- SemanticChangeEventType – the value of a property in the address space was changed such that the meaning of the system has changed. An example of this event is when the engineering units associated with a value are modified.

Together these events allow all changes in the address space to be monitored. The MTConnect events that are generated for Asset_Changed shall be mapped to one of these events.

6.2 Generic

6.2.1 Overview

The following item are generic asset items. Currently only one specific asset type is defined - Cutting Tool Asset. In the future additional Asset type will be defined.

6.2.2 AssetType

An abstract mobile asset

Table 123 - AssetType

Attribute	**Value**				
BrowseName	AssetType				
IsAbstract	True				
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the BaseObjectType					
HasProperty	Variable	SerialNumber	SerialNumberDataType	PropertyType	Mandatory
HasProperty	Variable	Manufacturers	ManufacturersDataType	PropertyType	Optional
HasProperty	Variable	LastChangeTimestamp	DateTime	PropertyType	Mandatory
HasComponent	Variable	AssetDescription	BaseDataType	BaseDataVariableType	Optional

The DeviceUUID is mapped to a reference to the object that supplied the data.

The ToolId is a definition of the type of Asset and is mapped to the TypeDefinition. It is assumed that individual type will exist for each class of Asset.

The AssetId is a unique identification of the Asset, this is mapped to the NodeId of the Asset

The SerialNumber is the unique identifier for this assembly.

Manufacturers – The manufacturers of the cutting tool.

The LastChangeTimestamp is the time associated with the last change to any aspect of this asset.

AssetDescription - The assets general description should be mapped to the description that is available as the part of all OPC objects and variables. This field should only be used if addition description space is required, or if data other than a string is required to describe the Asset.

6.2.3 MeasurementType

An abstract type for edge measurements

Table 124 - MeasurementType

Attribute		**Value**			
BrowseName		MeasurementType			
DataType		MeasurementValueDataType			
IsAbstract		True			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the AnalogItemType					
HasProperty	Variable	SignificantDigits	SignificantDigitsValueDataType	PropertyType	Optional
HasComponent	Variable	Units	EUInformation	BaseDataType	Optional
HasComponent	Variable	NativeUnits	EUInformation	BaseDataType	Optional
HasProperty	Variable	Code	CodeType	PropertyType	Optional
HasProperty	Variable	Maximum	MeasurementValueType	PropertyType	Optional
HasProperty	Variable	Minimum	MeasurementValueType	PropertyType	Optional
HasProperty	Variable	Nominal	MeasurementValueType	PropertyType	Optional

significantDigits – the number of digits that this value has.

Units – The engineering units associated with this value.

nativeUnits – The engineering units associated with the base value.

Code – a code type that is associated with the value.

Maximum – The maximum value associated with this value.

Minimum – The Minimum value associated with this value.

Nominal – The Normal value associated with this value.

6.3 Cutting Tool Asset

6.3.1 Overview of Cutting Tool

A Cutting Tool is an assembly of items for removing material from a work-piece through a shearing action at the defined cutting edge or edges of the Cutting Item. A Cutting Tool can be a single item or an assembly of one or more Adaptive Items, a Tool Item and several Cutting Items on a Tool Item. MTConnect will adopt the ISO 13399 structure when formulating the vocabulary for cutting tools. MTConnect will focus on the application of the cutting tool and cutting items. At this time we are only concerned with two aspects of the cutting tool, the Cutting Tool and the Cutting Item. We will not be including the Tool Item, Adaptive Item, or the Assembly Items, as they are component parts of the cutting tool do not have a large impact on the use phase of the tool and will be sufficiently defined in the ISO 13399 portion of the document[1].

[1] International Organization for Standardization. ISO 13399: Cutting tool data representation and exchange. Geneva, Switzerland, 2000.

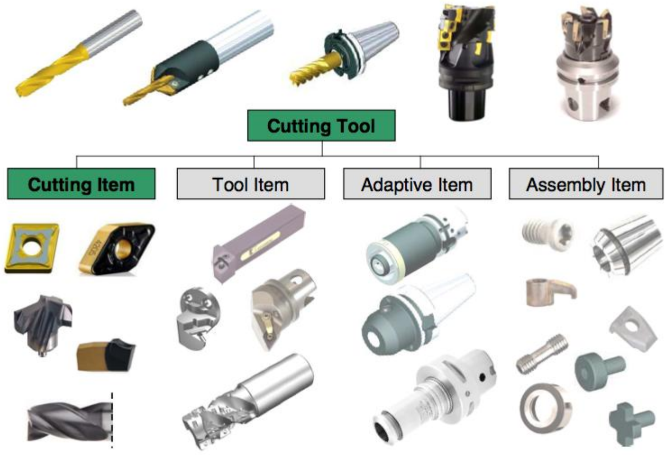


Figure 19 Cutting Tool Parts

The Figure 19 illustrates the parts of a cutting tool. The cutting tool is the aggregate of all the components and the cutting item is the part of the tool that removes the material from the work piece. These are the primary focus of MTConnect.

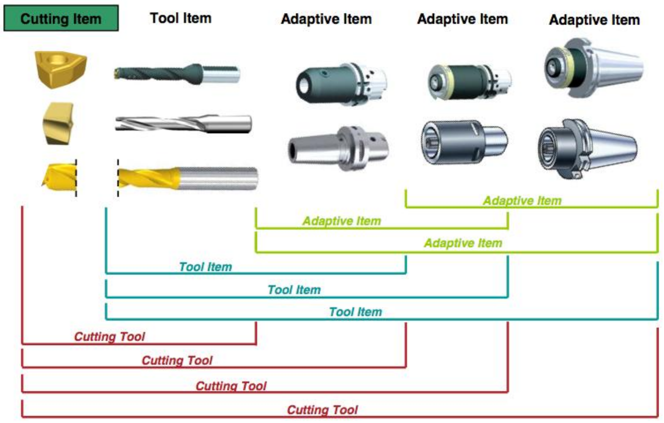


Figure 20: Cutting Tool Composition

Figure 20 provides another view of the cutting tool composition model. The adaptive items and tool items will be used for measurements, but will not be modeled as separate entities. The definitions will assume when referencing the cutting tool we are referring to the entirety of the asset and when we provide data regarding the cutting item we are referencing each individual component as illustrated on the left of the previous diagram.

Figure 21 and Figure 22 further illustrates the components of the cutting tool. As we compose the Tool Item, Cutting Item, Adaptive Item, we get a Cutting Tool. The Tool Item and Assembly Item will only be in the CuttingToolDefinitionType section that will contain the full ISO 13399 model.

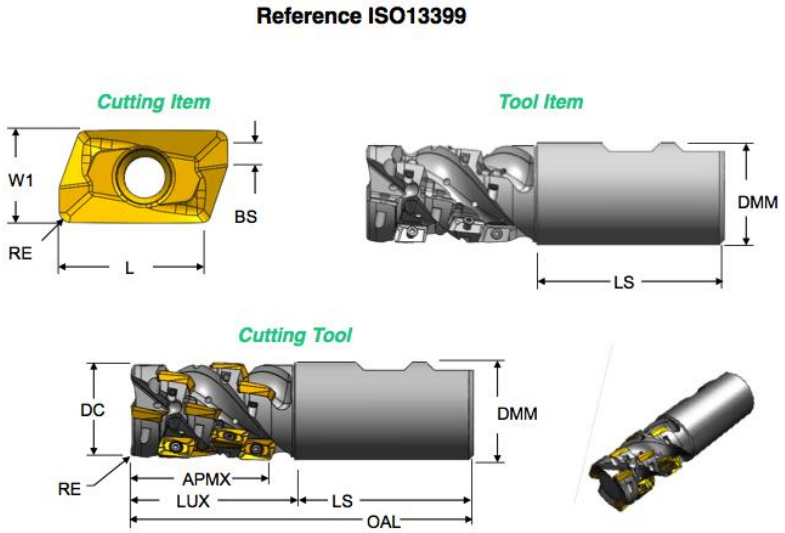


Figure 21: Cutting Tool, Tool Item and Cutting Item

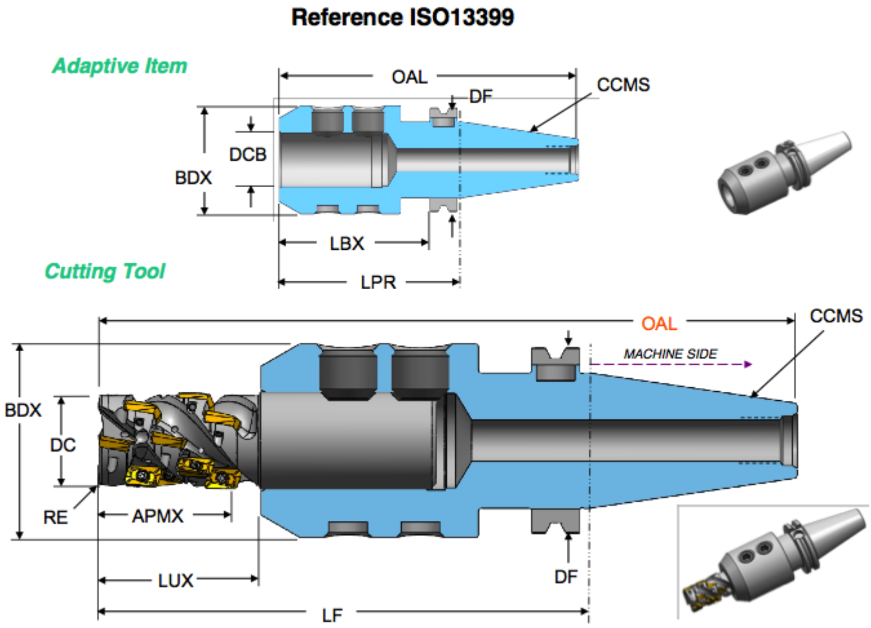


Figure 22: Cutting Tool, Tool Item and Cutting Item

Figure 22 use the ISO 13399 codes for each of the measurements. These codes will be translated into the MTConnect vocabulary as Figure 23 The measurements will have a maximum, minimum, and nominal value representing the tolerance of allowable values for this dimension.

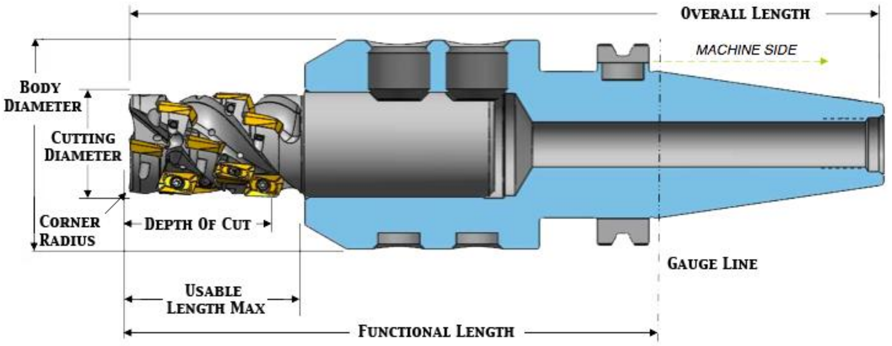


Figure 23: Cutting Tool Measurements

The MTConnect standard will not define the entire geometry of the cutting tool, but will provide the information necessary to use the tool in the manufacturing process. Additional information in MTConnect can be added to the definition of the cutting tool by means of schema extensions.

When mapping a cutting tool asset model to OPC UA, the model can be broken down into multiple sub models that are then used to build the final cutting tool model. The following figure illustrates the relationships between the types and general hierarchy of the object and variables. The exact mapping of MTConnect items to OPC Items is described later in this section.

6.3.2 CuttingToolType

A cutting tool

Table 125 - CuttingToolType

Attribute		Value			
BrowseName		CuttingToolType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the AssetType					
HasProperty	Variable	toolId	ToolIdDataType	PropertyType	Mandatory
HasProperty	Variable	toolGroup	ToolGroupDataType	PropertyType	Optional

6.3.3 CuttingToolLifeCycleType

A definition of a cutting tool application and life cycle

Table 126 - CuttingToolLifeCycleType

Attribute	Value				
BrowseName	CuttingToolLifeCycleType				
IsAbstract	False				
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**Modeling Rule**
HasComponent	Variable	CutterStatus	CutterStatusValueTypeEnum	CutterStatusType	Mandatory
HasComponent	Variable	ReconditionCount	ReconditionCountValueDataType	ReconditionCountType	Optional
HasComponent	Variable	ToolLife	BaseDataValueType	LifeType	Optional
HasComponent	Variable	ProgramToolNumber	ProgramToolNumberDataType	PropertyType	Optional
HasComponent	Variable	Location	LocationValueDataType	LocationType	Optional
HasComponent	Variable	ProgramSpindleSpeed	BaseDataValueType	ProgramSpindleSpeedType	Optional
HasComponent	Variable	ProgramFeedRate	FeedrateValueDataType	PropertyType	Optional
HasComponent	Variable	ConnectionCodeMachineSide	ConnectionCodeMachineSideDataType	PropertyType	Optional

Comment [PEH2]: Fix data type vs Typedefinitions (chek for each and correct both columns as needed

HasComponent	Variable	Measurements	MeasurementValueDataType	PropertyType	Optional
HasComponent	Variable	CuttingItems	BaseDataValueType	CuttingItemsType	Optional

6.3.4 CutterStatusType

The set of appropriate status for this cutting tool

Table 127 - CutterStatusType

Attribute		**Value**			
BrowseName		CutterStatusType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
HasComponent	Object	Status	CutterStatusValueType	FolderType	Mandatory

6.3.5 ReconditionCountType

The number of times this tool has been reconditioned

Table 128 - ReconditionCountType

Attribute		**Value**			
BrowseName		ReconditionCountType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the ReconditionCountValueType					
HasProperty	Variable	maximumCount	MaximumCountType	PropertyType	Optional

6.3.6 LifeType

Abstract cutter life

Table 129 - LifeType

Attribute		**Value**			
BrowseName		LifeType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the ToolLifeValueType					
HasProperty	Variable	Type	ToolLifeTypeEnum	PropertyType	Mandatory
HasProperty	Variable	countDirection	ToolLifeDirectionTypeEnum	PropertyType	Mandatory
HasProperty	Variable	warningLevel	ToolLifeValueDataType	PropertyType	Optional
HasProperty	Variable	Maximum	ToolLifeValueDataType	PropertyType	Mandatory

6.3.7 LocationType

The location of the tool in the tool changer (pot) or the station of the tool

Table 130 - LocationType

Attribute		**Value**			
BrowseName		LocationType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the LocationValueType					
HasProperty	Variable	Type	LocationsTypeEnum	PropertyType	Mandatory
HasProperty	Variable	negativeOverlap	OverlapDataType	PropertyType	Mandatory
HasProperty	Variable	positiveOverlap	OverlapDataType	PropertyType	Mandatory

6.3.8 ProgramSpindleSpeedType

The spindle speed properties of this tool

Table 131 - ProgramSpindleSpeedType

Attribute		Value			
BrowseName		ProgramSpindleSpeedType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the SpeedType					
HasProperty	Variable	maximum	MaximumDataType	PropertyType	Optional
HasProperty	Variable	minimum	MinimumDataType	PropertyType	Optional
HasProperty	Variable	nominal	NominalDataType	PropertyType	Optional

6.3.9 ProgramFeedRateType

The feed rate properties of this tool in MILLIMETERS/SECOND

Table 132 - ProgramFeedRateType

Attribute		Value			
BrowseName		ProgramFeedRateType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the SpeedType					
HasProperty	Variable	Maximum	MaximumDataType	PropertyType	Optional
HasProperty	Variable	Minimum	MinimumDataType	PropertyType	Optional
HasProperty	Variable	Nominal	NominalDataType	PropertyType	Optional

6.3.10 CuttingItemType

An edge into a tool assembly

Table 133 - CuttingItemType

Attribute		Value			
BrowseName		CuttingItemType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
HasProperty	Variable	Indices	IndexRangeDataType	PropertyType	Mandatory
HasProperty	Variable	itemId	ItemIdDataType	PropertyType	Optional
HasProperty	Variable	Grade	GradeDataType	PropertyType	Optional
HasProperty	Variable	manufacturers	ManufacturersDataType	PropertyType	Optional
HasComponent	Object	Description	AssetDescriptionType	FolderType	Optional
HasComponent	Object	Locus	LocusDataType	FolderType	Optional
HasComponent	Object	ItemLife	LifeType	FolderType	Optional
HasComponent	Object	Measurements	CuttingItemMeasurementsType	FolderType	Optional

6.3.11 CuttingItemsType

A list of edge

Table 134 - CuttingItemsType

Attribute		Value			
BrowseName		CuttingItemsType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
HasProperty	Variable	Count	EdgeCountDataType	PropertyType	Mandatory
HasComponent	Object	CuttingItem	CuttingItemType	FolderType	Mandatory(1..unbounded)

6.3.12 AssemblyMeasurementType

Measurements for the assembly

Table 135 - AssemblyMeasurementType

Attribute		Value			
BrowseName		AssemblyMeasurementType			
IsAbstract		True			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the MeasurementType					

6.3.13 BodyDiameterMaxType

BDX: The largest diameter of the body of a tool item

Table 136 - BodyDiameterMaxType

Attribute		Value			
BrowseName		BodyDiameterMaxType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the AssemblyMeasurementType					

6.3.14 BodyLengthMaxType

LBX: The distance measured along the X axis from that point of the item closest to the workpiece, including the cutting item for a tool item but excluding a protruding locking mechanism for an adaptive item, to either the front of the flange on a flanged body or the beginning of the connection interface feature on the machine side for cylindrical or prismatic shanks

Table 137 - BodyLengthMaxType

Attribute		Value			
BrowseName		BodyLengthMaxType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the AssemblyMeasurementType					

6.3.15 CommonMeasurementType

Measurements for both the assembly and the cutting item

Table 138 - CommonMeasurementType

Attribute		Value			
BrowseName		CommonMeasurementType			
IsAbstract		True			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the MeasurementType					

6.3.16 CornerRadiusType

nominal radius of a rounded corner measured in the XY-plane

Table 139 - CornerRadiusType

Attribute		Value			
BrowseName		CornerRadiusType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the CuttingItemMeasurementType					

6.3.17 CuttingDiameterMaxType

DC: The maximum diameter of a circle on which the defined point Pk of each of the master inserts is located on a tool item. The normal of the machined peripheral surface points towards the axis of the cutting tool.

Table 140 - CuttingDiameterMaxType

Attribute		**Value**			
BrowseName		CuttingDiameterMaxType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the AssemblyMeasurementType					

6.3.18 CuttingDiameterType

Diameter of a circle on which the defined point Pk of each of the master inserts is located on a tool item. The normal of the machined peripheral surface points towards the axis of the cutting tool.

Table 141 - CuttingDiameterType

Attribute		**Value**			
BrowseName		CuttingDiameterType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the CuttingItemMeasurementType					

6.3.19 CuttingEdgeLengthType

Theoretical length of the cutting edge of a cutting item over sharp corners

Table 142 - CuttingEdgeLengthType

Attribute		**Value**			
BrowseName		CuttingEdgeLengthType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the CuttingItemMeasurementType					

6.3.20 CuttingHeightType

Theoretical length of the cutting edge of a cutting item over sharp corners

Table 143 - CuttingHeightType

Attribute		**Value**			
BrowseName		CuttingHeightType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the CuttingItemMeasurementType					

6.3.21 CuttingItemMeasurementType

Measurements for the cutting item

Table 144 - CuttingItemMeasurementType

Attribute		**Value**			
BrowseName		CuttingItemMeasurementType			
IsAbstract		True			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the MeasurementType					

6.3.22 CuttingItemMeasurementsType

A collection of assembly measurements

Table 145 - CuttingItemMeasurementsType

Attribute		**Value**			
BrowseName		CuttingItemMeasurementsType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**

6.3.23 CuttingReferencePointType

The theoretical sharp point of the cutting tool from which the major functional dimensions are taken

Table 146 - CuttingReferencePointType

Attribute		**Value**			
BrowseName		CuttingReferencePointType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the CuttingItemMeasurementType					

6.3.24 CuttingToolDefinitionType

The description of an asset, can be freeform text or elements

Table 147 - CuttingToolDefinitionType

Attribute		**Value**			
BrowseName		CuttingToolDefinitionType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
HasProperty	Variable	Format	DefinitionFormatType	PropertyType	Optional

6.3.25 DepthOfCutMaxType

APMX: The maximum engagement of the cutting edge or edges with the workpiece measured perpendicular to the feed motion.

Table 148 - DepthOfCutMaxType

Attribute		**Value**			
BrowseName		DepthOfCutMaxType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the AssemblyMeasurementType					

6.3.26 DriveAngleType

The DriveAngleType is the angle between the driving mechanism locator on a tool item and the main cutting edge.

Table 149 - DriveAngleType

Attribute		**Value**			
BrowseName		DriveAngleType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the AssemblyMeasurementType					

6.3.27 FlangeDiameterMaxType

The maximum dimension between two parallel tangents on the outside edge of a flange

Table 150 - FlangeDiameterMaxType

Attribute		Value			
BrowseName		FlangeDiameterMaxType			
IsAbstract		false			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the AssemblyMeasurementType					

6.3.28 FlangeDiameterType

dimension between two parallel tangents on the outside edge of a flange

Table 151 - FlangeDiameterType

Attribute		Value			
BrowseName		FlangeDiameterType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the CuttingItemMeasurementType					

6.3.29 FunctionalLengthType

LF: The furthest distance from the gauge plane.

Table 152 - FunctionalLengthType

Attribute		Value			
BrowseName		FunctionalLengthType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the CommonMeasurementType					

6.3.30 FunctionalWidthType

Distance between the cutting reference point and the rear backing surface of a turning tool or the axis of a boring bar

Table 153 - FunctionalWidthType

Attribute		Value			
BrowseName		FunctionalWidthType			
IsAbstract		false			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the CuttingItemMeasurementType					

6.3.31 InclinationAngleType

angle between the tool rake plane and a plane parallel to the xy-plane measured in the tool cutting edge plane

Table 154 - InclinationAngleType

Attribute		Value			
BrowseName		InclinationAngleType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the CuttingItemMeasurementType					

6.3.32 IncribedCircleDiameterType

Diameter of a circle to which all edges of a equilateral and round regular insert are tangential

Table 155 - IncribedCircleDiameterType

Attribute	Value

BrowseName		IncribedCircleDiameterType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the CuttingItemMeasurementType					

6.3.33 OverallToolLengthType

OAL: largest length dimension of the tool item including the master insert where applicable

Table 156 - OverallToolLengthType

Attribute		**Value**			
BrowseName		OverallToolLengthType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the AssemblyMeasurementType					

6.3.34 PointAngleType

angle between the major cutting edge and the same cutting edge rotated by 180 degrees about the tool axis

Table 157 - PointAngleType

Attribute		**Value**			
BrowseName		PointAngleType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the CuttingItemMeasurementType					

6.3.35 ProtrudingLengthType

Dimension from the yz-plane to the furthest point of the tool item or adaptive item measured in the -X direction

Table 158 - ProtrudingLengthType

Attribute		**Value**			
BrowseName		ProtrudingLengthType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the CommonMeasurementType					

6.3.36 ShankDiameterType

DMM: dimension of the diameter of a cylindrical portion of a tool item or an adaptive item that can participate in a connection

Table 159 - ShankDiameterType

Attribute		**Value**			
BrowseName		ShankDiameterType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the AssemblyMeasurementType					

6.3.37 ShankHeightType

H: dimension of the height of a shank

Table 160 - ShankHeightType

Attribute	**Value**
BrowseName	ShankHeightType

IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the AssemblyMeasurementType					

6.3.38 ShankLengthType

LS: dimension of the length of a shank

Table 161 - ShankLengthType

Attribute		**Value**			
BrowseName		ShankLengthType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the AssemblyMeasurementType					

6.3.39 StepDiameterLengthType

length of a portion of a cutting tool that is related to the corresponding cutting diameter. The length is measured from the point "PK" of the corresponding diameter to the next projected point where the diameter starts to change

Table 162 - StepDiameterLengthType

Attribute		**Value**			
BrowseName		StepDiameterLengthType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the CuttingItemMeasurementType					

6.3.40 StepIncludedAngleType

angle between a major edge on a step of a stepped tool and the same cutting edge rotated 180 degrees about ist tool axis

Table 163 - StepIncludedAngleType

Attribute		**Value**			
BrowseName		StepIncludedAngleType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the CuttingItemMeasurementType					

6.3.41 ToolCuttingEdgeAngleType

angle between the tool cutting edge plane and the tool feed plane measured in a plane parallel the xy-plane

Table 164 - ToolCuttingEdgeAngleType

Attribute		**Value**			
BrowseName		ToolCuttingEdgeAngleType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the CuttingItemMeasurementType					

6.3.42 ToolLeadAngleType

angle between the tool cutting edge plane and a plane perpendicular to the tool feed plane measured in a plane parallel the xy-plane

Table 165 - ToolLeadAngleType

Attribute	**Value**

BrowseName		ToolLeadAngleType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the CuttingItemMeasurementType					

6.3.43 ToolOrientationType

The orientation as expressed in degrees of the cutting item to the work piece for this process.

Table 166 - ToolOrientationType

Attribute		**Value**			
BrowseName		ToolOrientationType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the BaseDataVariableType					

6.3.44 UsableLengthMaxType

The UsableLengthMaxType is the maximum length of a cutting tool that can be used in a particular cutting operation including the non-cutting portions of the tool.

Table 167 - UsableLengthMaxType

Attribute		**Value**			
BrowseName		UsableLengthMaxType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the AssemblyMeasurementType					

6.3.45 WeightType

WT: The weight measured in grams

Table 168 - WeightType

Attribute		**Value**			
BrowseName		WeightType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the CommonMeasurementType					

6.3.46 WiperEdgeLengthType

measure of the length of a wiper edge of a cutting item

Table 169 - WiperEdgeLengthType

Attribute		**Value**			
BrowseName		WiperEdgeLengthType			
IsAbstract		False			
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the CuttingItemMeasurementType					

7 DataType Mapping

7.1 Overview

This section defines a large number of DataTypes that exist in MTConnect and generate a mapping for them to OPC UA DataTypes. In many cases the MTConnect DataType is created as a Subtype of an OPC UA DataType. Where OPC UA has an existing complex DataType, the OPC UA DataType was used. For example Engineering units are mapped to OPC UA Engineering Units. The following table provides a listing of these types

MTConnect data type	**OPC UA DataType**	**Comment**
NativeUnitsType	EUInformation	The NativeUnits data type is nothing more then engineering units for the an item.
SampleIntervalType	Duration	The definition of the OPC UA type matches the definition of SampleIntervalType DataType

For all DataTypes that describe an enumeration, the data type only list the defined setting for the enumeration. These DataTypes are a Subtype of the Enumeration data type defined in [UA Part 3].

> From Part 3 – "*Enumeration DataTypes* are *DataTypes* that represent discrete sets of named values. Enumerations are always encoded as Int32 on the wire. Enumeration *DataTypes* inherit directly or indirectly from the *DataType Enumeration*. Enumerations have no encodings exposed in the *AddressSpace*. To expose the human-readable representation of an enumerated value the *DataType Node* may have the *EnumStrings Property* that contains an array of *LocalizedText*. The Integer representation of the enumeration value points to a position within that array. *EnumValues Property* can be used instead of the *EnumStrings* to support integer representation of enumerations that are not zero-based or have gaps. The *EnumValues Property* contains an array of a *Structured DataType* containing the integer representation as well as the human-readable representation."

For all MTConnect defined DataType that are enumerations the *EnumStrings Property* will be provided, unless a specific table is provide and includes the *EnumValues Property.*

The following items are MTConnect items that have been mapped to either existing OPC UA types or can be mapped using existing concepts, in particular the creation of a variable type that only represents a collection of another Variable type is mapped to the original variable type, since in OPU UA a variable value can be an array or the value.

Table 170 - MTConnect to OPC UA Variable Mapping

MTConnect Type	**OPC UA Type or Other**	**Comment**
ChannelsType	An array of ChannelType	Additional type not needed, just an array of ChannelType
ComponentsType	An array of ComponentType	Additional type not needed, just a collection of ComponentType
DataItemsType	An array of DataItemType	Additional type not needed, just a collection of DataItemType
AssemblyMeasurementsType	An array of AssemblyMeasurementType	Additional type not needed, just an array of AssemblyMeasurementType
AssetsType	An array of AssetType	Additional type not needed, just an array of AssetType
AssetDescriptionType	String	This type is not listed as a type, but is just a variable in the AssetType that provides an additional description.

When mapping MTConnect data types to OPC UA data types the following general concepts were used

- If OPC UA has a base data type that matches the OPC UA data type was used
- All structure type data was created as sub-types of the OPC UA structure data type
- All Enumerations where created as a sub=type of OPC UA Enumeration type.

Also a number of MTConnect data types can be mapped to a floating point datatype with associated engineering units

The following table lists the MTConnect Complex types that are mapped into OPC UA float type with engineering units. The following table provides a summary of these types:

Table 171 – EngineeringUnits Mapping

Sample	MTConnect Units	OPC UA Engineering Units
AbsTimeSeriesType		
AccelerationType	MILLIMETER/SECOND^2	
AccumulatedTimeType	SECOND	
AmperageType	AMPERE	
AngleType	DEGREE	
AngularAccelerationType	DEGREE/SECOND^2	
AngularVelocityType	DEGREE/SECOND	
AxisFeedrateType	MILLIMETER/SECOND or PERCENT for override	
ConcentrationType	PERCENT	
ConductivityType	SIEMENS/METER	
DisplacementType	MILLIMETER	
ElectricalEnergyType	WATT_SECOND	
FlowType	LITER/SECOND	
FrequencyType	HERTZ	
LinearForceType	NEWTON	
LoadType	PERCENT	
PathFeedrateType	MILLIMETER/SECOND or PERCENT for Override	
PathPositionType	default to WORK coordinates in MILLIMETER_3D	
PositionType	MILLIMETER	
PowerFactorType	PERCENT	
PressureType	PASCAL	
ResistanceType	OHM	
RotationalVelocityType	REVOLUTION/MINUTE or PERCENT for Override	
SoundPressureType	DECIBEL	
StrainType	PERCENT	
TemperatureType	degrees CELSIUS	
TiltType	MICRO_RADIAN	
TorqueType	NEWTON_METER	
VelocityType	MILLIMETER/SECOND	
ViscosityType	PASCAL_SECOND	
VoltageType	VOLT	
WattageType	WATT	

Comment [PEH3]: This is actually

“**Codes for Units of Measurement (Recommendation N°. 20)”** (see UN/CEFACT).

7.2 ActuatorStateTypeEnum

Enumeration: ActuatorStateTypeEnum

Table 172 - ActuatorStateTypeEnum

Enum	Description
ACTIVE_1	
INACTIVE_2	

The actuator state of the component

7.3 AlarmStateTypeEnum

Enumeration: AlarmStateTypeEnum

Table 173 –AlarmStateTypeEnum

Enum	Description
ACTIVE_1	
CLEARED_2	

The active or cleared state of the notification

7.4 AssetAttrDataType

An asset attribute type

Table 174 – AssetAttrDataType

Attribute	Value
BrowseName	AssetAttrDataType
IsAbstract	False
Subtype of string	

7.5 AssetBufferSizeDataType

The maximum number of assets

Table 175 - AssetBufferSizeDataType

Attribute	**Value**
BrowseName	AssetBufferSizeDataType
IsAbstract	False
Subtype of integer	

7.6 AssetCountAttrDataType

The number of assets

Table 176 - AssetCountAttrDataType

Attribute	**Value**
BrowseName	AssetCountAttrDataType
IsAbstract	False
Subtype of integer	

7.7 AssetIdDataType

The unique id of the asset

Table 177 - AssetIdDataType

Attribute	**Value**
BrowseName	AssetIdDataType
IsAbstract	False
Subtype of string	

7.8 AvailabilityTypeEnum

Enumeration: AvailabilityTypeEnum

Table 178 - AvailabilityTypeEnum

Enum	Description
AVAILABLE_1	
UNAVAILABLE_2	No data item reading for this device is available.

7.9 AxesCouplingTypeEnum

Enumeration: AxesCouplingTypeEnum

Table 179 - AxesCouplingTypeEnum

Enum	Description
MASTER_1	In a master-slave motion system, one controlling axis and one slave axis as a function of the position of the master axis
SLAVE_2	Controlled position of slave axis as a function of the position of the master axis
SYCHRONOUS_3	
TANDEM_4	
UNAVAILABLE_5	No data item reading for this device is avaiable.

7.10 AxesListValueDataType

A space delimited list of values

Table 180 - AxesListValueDataType

Attribute	Value
BrowseName	AxesListValueDataType
IsAbstract	False
Subtype of string	

7.11 BufferSizeDataType

The size of the agents buffer

Table 181 - BufferSizeDataType

Attribute	**Value**
BrowseName	BufferSizeDataType
IsAbstract	False
Subtype of integer	

7.12 CalibrationDateDataType

The time the calibration was preformed

Table 182 CalibrationDateDataType

Attribute	**Value**
BrowseName	CalibrationDateDataType
IsAbstract	False
Subtype of DateTime (defined in [UA Part 5])	

7.13 CalibrationInitialsDataType

The initials of the person doing the calibration

Table 183 - CalibrationInitialsDataType

Attribute	**Value**
BrowseName	CalibrationInitialsDataType
IsAbstract	False
Subtype of string	

7.14 CategoryTypeEnum

CategoryTypeEnum enumeration values are defined in

Table 184 - CategoryTypeEnum Description

Enum	**Description**
CONDITION_1	
EVENT_2	

SAMPLE_3	

The measurement sampling type

7.15 ChannelNumberDataType

The channel number

Table 185 - ChannelNumberDataType

Attribute	Value
BrowseName	ChannelNumberDataType
IsAbstract	False
Subtype of Int16 (defined in [UA Part 5])	

7.16 ClampStateTypeEnum

Enumeration: ClampStateTypeEnum

Table 186 - ClampStateTypeEnum

Enum	Description
CLOSED_1	State of being closed.
INDETERMINATE_2	Indeterminate state.
OPEN_3	State of being open.
UNAVAILABLE_4	No data item reading for this device is available.

7.17 ConditionDescriptionDataType

The description of the Condition

Table 187 - ConditionDescriptionDataType

Attribute	Value
BrowseName	ConditionDescriptionDataType
IsAbstract	False
Subtype of string	

7.18 ControllerModeTypeEnum

Enumeration: ControllerModeTypeEnum

Table 188 - ControllerModeTypeEnum

Enum	Description
AUTOMATIC_1	Machine mode that allows an operator to execute a part program on the CNC machine.
MANUAL_2	Machine mode where CNC machine behaves like a standard or conventional machine.
MANUAL_DATA_INPUT_3	Machine mode whereby the operator can enter data through the keyboard on the control panel and display screen. In MDI mode, CNC commands are entered through the keyboard and display screen manually and can be executed once.
SEMI_AUTOMATIC_4	
UNAVAILABLE_5	No data item reading for this device is available.

CNC mode state

7.19 CoordinateSystemTypeEnum

Enumeration: CoordinateSystemTypeEnum

Table 189 - CoordinateSystemTypeEnum

Enum	Description
MACHINE_1	Unchangeable coordinate system that has machine zero as its origin.

WORK_2	The coordinate system that represents the working area for a particular work piece whose origin is shifted within the MACHINE coordinate system. If the WORK coordinates are not currently defined in the device, the MACHINE coordinates will be used.

7.20 CodeDataType

An application specific code

Table 190 - CodeDataType

Attribute	**Value**
BrowseName	CodeDataType
IsAbstract	False
Subtype of string	

7.21 ComponentIdDataType

The id of the component (maps to the id from probe)

Table 191 - ComponentIdDataType

Attribute	**Value**
BrowseName	ComponentIdDataType
IsAbstract	False
Subtype of ID	

7.22 ConnectionCodeMachineSideDataType

The code for the connection to the machine

Table 192 - ConnectionCodeMachineSideDataType

Attribute	**Value**
BrowseName	ConnectionCodeMachineSideDataType
IsAbstract	False
Subtype of string	

7.23 CoordinateSystemTypeEnum

Enumeration: CoordinateSystemTypeEnum

Table 193 - CoordinateSystemTypeEnum

Enum	**Description**
MACHINE_1	Unchangeable coordinate system that has machine zero as its origin.
WORK_2	The coordinate system that represents the working area for a particular work piece whose origin is shifted within the MACHINE coordinate system. If the WORK coordinates are not currently defined in the device, the MACHINE coordinates will be used.

The coordinate system to be used for the position

7.24 CountValueDataType

The number of values

Table 194 - CountValueDataType

Attribute	Value
BrowseName	CountValueDataType
IsAbstract	False
Subtype of string	

7.25 CreationTimeDataType

The date and time the document was created

Table 195 - CreationTimeDataType

Attribute	Value
BrowseName	CreationTimeDataType
IsAbstract	False
Subtype of DateTime	

7.26 CutterStatusValueTypeEnum

Enumeration: CutterStatusValueTypeEnum

Table 196 - CutterStatusValueTypeEnum

Enum	Description
ALLOCATED_1	
AVAILABLE_2	
BROKEN_3	
EXPIRED_4	
MEASURED_5	
NEW_6	
NOT_REGISTERED_7	
RECONDITIONED_8	
TAGGED_OUT_9	
UNALLOCATED_10	
UNAVAILABLE_11	No data item reading for this device is available.
UNKNOWN_12	
USED_13	

The state of the tool. These can be combined to define the complete cutting tool state

7.27 DataItemEnumDataType

The types of measurements available

Table 197 - DataItemEnumDataType

Name	Type	Description
DataItemEnumDataType	Structure	Information that describes the build of the software.
DataItemEnum	DataItemEnumTypeEnum	The enumeration of the data Item type, if VendorDefined_73 then the string defined in following field is used, otherwise the string is ignored and shall be null.
DataItemEnumString	String	The vendor custom string.

7.28 DataItemEnumTypeEnum

Enumeration: DataItemEnumTypeEnum

Table 198 - DataItemEnumTypeEnum

Enum	Description
ACCELERATION_1	Rate of change of velocity
ACCUMULATED_TIME_2	The measurement of accumulated time associated with a component
ACTIVE_AXES_3	The set of axes associated with a path that the controller is controlling. If this data item is not provided, it will be assumed the controller is controlling all axes.
ACTUATOR_4	An actuator related condition.
ACTUATOR_STATE_5	The state of the actuator. ACTIVE or INACTIVE.
ALARM_6	DEPRECATED: Replaced with CONDITION category. Rel. 1.1.

Enum	Description
AMPERAGE_7	The measurement of AC Current or a DC current. Subtypes: ALTERNATING The measurement of alternating current. If not specified further in Statistic, defaults to RMS Current units in AMPERE. DIRECT The measurement of DC current units in AMPERE.
ANGLE_8	The angular position of a component relative to the parent. ACTUAL The angular position as read from the physical component in units DEGREE. COMMANDED - The angular position computed by the controller in units DEGREE.
ANGULAR_ACCELERATION_9	Rate of change of angular velocity.
ANGULAR_VELOCITY_10	Rate of change of angular position.
ASSET_CHANGED_11	
AVAILABILITY_12	Represents the components ability to communicate its availability. This SHALL be provided for the device and MAY be provided for all other components
AXIS_COUPLING_13	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 SHALL be viewed from the perspective of the axis, therefore a MASTER coupling indicates that this axis is the master of the COUPLED_AXES.
AXIS_FEEDRATE_14	The feedrate of a linear axis. Subtypes: ACTUAL - The actual federate of a linear axis in units MILLIMETER/SECOND. OMMANDED - The feedrate as specified in the program in units MILLIMETER/SECOND. OVERRIDE - The operator’s overridden value. Percent of commanded in units PERCENT.
BLOCK_15	The block of code being executed. The block contains the entire expression of the step in the program.
CODE_16	DEPRECATED. Rel 1.1.0
COMMUNICATIONS_17	A communications failure indicator.
CONCENTRATION_18	Percentage of one component within a mixture of components
CONDUCTIVITY_19	The ability of a material to conduct electricity
CONTROLLER_MODE_20	The current controller’s mode. AUTOMATIC, MANUAL, MANUAL_DATA_INPUT, FEED_HOLD, or SEMI_AUTOMATIC.
COUPLED_AXES_21	Refers to the set of associated axes. The value will be a space delimited set of axes names.
DIRECTION_22	The direction of motion. CLOCKWISE or COUNTER_CLOCKWISE
DISPLACEMENT_23	The displacement as the change in position of an object
DOOR_STATE_24	The opened or closed state of the door. OPEN, UNLATCHED, or CLOSED.
ELECTRICAL_POWER_25	
EMERGENCY_STOP_26	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 SHALL cease operation).
EXECUTION_27	The execution status of the Controller. READY, ACTIVE, INTERRUPTED, or STOPPED
FILL_LEVEL_28	The measurement of the amount of a substance remaining compared to the planned maximum amount of that substance
FLOW_29	The rate of flow of a fluid
FREQUENCY_30	The measurement of the number of occurrences of a repeating event per unit time
HARDWARE_31	The hardware subsystem of the component operation condition.
LEVEL_32	Deprecated in Rel. 1.2 See Fill_Level
LINE_33	The state of the high voltage line.
LINEAR_FORCE_34	The measure of the push or pull introduced by an actuator or exerted on an object
LOAD_35	The measurement of the percentage of the standard rating of a device
LOGIC_PROGRAM_36	An error occurred in the logic program or PLC (programmable logic controller).
MASS_37	The measurement of the mass of an object(s) or an amount of material
MESSAGE_38	An uninterrupted textual notification.
MOTION_PROGRAM_39	An error occurred in the motion program.
PART_COUNT_40	The current count of parts produced as represented by the controller. SHALL 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_41	An identifier of the current part in the device

Enum	Description
PATH_FEEDRATE_42	The feedrate of the tool path. Subtypes: ACTUAL - The three-dimensional feedrate derived from all components in units MILLIMETER/SECOND. COMMANDED - The feedrate as specified in the program in units MILLIMETER/SECOND. OVERRIDE - The operator's overridden value. Percent of commanded in units PERCENT.
PATH_MODE_43	The operational mode for this Path. SYNCHRONOUS, MIRROR, or INDEPENDENT. Default value is INDEPENDENT if not specified.
PATH_POSITION_44	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. ACTUAL The position of the component as read from the device in units MILLIMETER_3D. COMMANDED The position computed by the controller in units MILLIMETER_3D. TARGET The target position for the movement in units MILLIMETER_3D. PROBE The position provided by a probe in units MILLIMETER_3D.
PH_45	A measure of the acidity or alkalinity of a solution
POSITION_46	The position of the component. Defaults to MACHINE coordinates. Subtypes: ACTUAL-The position of the component as read from the device in units MILLIMETER. COMMANDED- The position as given by the Controller in units MILLIMETER. TARGET- The target position for the movement in units - MILLIMETER.
POWER_FACTOR_47	The measurement of the ratio of real power flowing to a load to the apparent power in that AC circuit.
POWER_STATE_48	The ON or OFF status of the component. DEPRECATION WARNING: MAY be deprecated in the future.
POWER_STATUS_49	DEPRECATED. Rel. 1.1.
PRESSURE_50	The force per unit area exerted by a gas or liquid
PROGRAM_51	The name of the program being executed
RESET_52	
RESISTANCE_53	The measurement of the degree to which an object opposes an electric current through it
ROTARY_MODE_54	The mode for the Rotary axis. SPINDLE, INDEX, or CONTOUR.
ROTARY_VELOCITY_55	The rotational speed of a rotary axis. ACTUAL The rotational speed the rotary axis is spinning at. ROTARY_MODE SHALL be SPINDLE in units REVOLUTION/MINUTE COMMANDED The rotational speed as specified in the program in units REVOLUTION/MINUTE. OVERRIDE The operator's overridden value. Percent of commanded in units PERCENT.
SOUND_PRESSURE_56	
SPINDLE_SPEED_57	DEPRECATED in REL 1.2. Replaced by ROTARY_VELOCITY
STRAIN_58	Strain is the amount of deformation per unit length of an object when a load is applied.
SYSTEM_59	A condition representing something that is not the operator, program, or hardware. This is often used for operating system issues.
TEMPERATURE_60	The measurement of temperature
TILT_61	A measurement of angular displacement
TOOL_ASSET_ID_62	The identifier of the tool currently in use for a given Path
TORQUE_63	The turning force exerted on an object or by an object
VELOCITY_64	The rate of change of position.
VIBRATION_65	
VISCOSITY_66	A measurement of a fluid's resistance to flow
VOLTAGE_67	The measurement of electrical potential between two points
VOLT_AMPERE_68	Volt-Ampere (VA)
VOLT_AMPERE_REACTIVE_69	Volt-Ampere Reactive (var)
WATTAGE_70	The measurement of power consumed or dissipated by an electrical circuit or device
WATT_SECOND_71	Measurement of electrical energy, equal to one Joule
WORKHOLDING_ID_72	The identifier for the workholding currently in use for a given Path
VendorDefined_73	The actual string is vendor defined

The types of measurements available

7.29 DataItemIdDataType

The item's unique ID that references the data item id from probe

Table 199 - DataItemIdDataType

Attribute	Value
BrowseName	DataItemIdDataType
IsAbstract	False
Subtype of NMTOKEN	

7.30 DataItemStatisticsDataType

Statistical operations on data

Table 200 - DataItemStatisticsDataType

Name	Type	Description
DataItemStatisticsDataType	Structure	Information that describes the build of the software.
DataItemStatisticsEnum	DataItemStatisticsTypeEnum	The enumeration of the data Item type, if VendorDefined_73 then the string defined in following field is used, otherwise the string is ignored and shall be null.
DataItemStatisticsString	String	The vendor custom string.

7.31 DataItemStatisticsTypeEnum

Enumeration: DataItemStatisticsTypeEnum

Table 201 - DataItemStatisticsTypeEnum

Enum	Description
AVERAGE_1	Mathematical Average value calculated for the DataItem during the calculation period
KURTOSIS_2	A measure of the “peakedness” of a probability distribution; i.e., the shape of the distribution curve
MAXIMUM_3	Maximum or peak value recorded for the DataItem during the calculation period
MEAN_4	
MINIMUM_5	Minimum value recorded for the DataItem during the calculation period
MODE_6	The number in a series of numbers that occurs most often
RANGE_7	Difference between the Maximum value and Minimum value of a DataItem during the calculation period. Also represents Peak-to-Peak measurement in an waveform.
ROOT_MEAN_SQUARE_8	Mathematical Root Mean Value (RMS) value calculated for the DataItem during the calculation period
STANDARD_DEVIATION_9	Statistical Standard Deviation value calculated for the DataItem during the calculation period

Statistical operations on data

7.32 DataItemSubEnumTypeEnum

Enumeration: DataItemSubEnumTypeEnum

Table 202 - DataItemSubEnumTypeEnum

Enum	Description
ACTUAL_1	The value of the component as read from the device.
ALL_2	
ALTERNATING_3	The measurement of alternating voltage. If not specified further in Statistic, defaults to RMS voltage
A_SCALE_4	A Scale weighting factor. This is the default weighting factor if no factor is specified
BAD_5	
B_SCALE_6	B Scale weighting factor
COMMANDED_7	The value computed by the controller.
CONTROL_8	The state of the low power line.
C_SCALE_9	C Scale weighting factor
DIRECT_10	The measurement of DC voltage

DYNAMIC_11	
D_SCALE_12	D Scale weighting factor
GOOD_13	
KINETIC_14	
LINE_15	The state of the high voltage line.
MAXIMUM_16	Maximum or peak value recorded for the DataItem during the calculation period
MINIMUM_17	Minimum value recorded for the DataItem during the calculation period
MOLE_18	
NO_SCALE_19	No weighting factor on the frequency scale
OTHER_20	Unsupported units
OVERRIDE_21	The operator’s overridden value. Percent of commanded.
PROBE_22	The value provided by a probe
TARGET_23	The target value for the movement.
VOLUME_24	
WEIGHT_25	

The sub-types for a measurement

7.33 DefinitionFormatTypeEnum

The sub-types for a measurement Enumeration: DefinitionFormatTypeEnum

Table 203 - DefinitionFormatTypeEnum

Enum	Description
EXPRESS_1	
TEXT_2	
UNDEFINED_3	
XML_4	

The format of the definition

7.34 DecibelValueDataType

The sound pressure

Table 204 -DecibelValueDataType

Attribute	Value
BrowseName	DecibelValueDataType
IsAbstract	False
Subtype of string	

7.35 DescriptionTextDataType

A description

Table 205 - DescriptionTextDataType

Attribute	Value
BrowseName	DescriptionTextDataType
IsAbstract	False
Subtype of string	

7.36 DirectionTypeEnum

Enumeration: DirectionTypeEnum

Table 206 - DirectionTypeEnum

Enum	Description
CLOCKWISE_1	
COUNTER_CLOCKWISE_2	

NEGATIVE_3	
POSITIVE_4	
UNAVAILABLE_5	No data item reading for this device is available.

The direction of rotation

7.37 DoorStateTypeEnum

Enumeration: DoorStateTypeEnum

Table 207 - DoorStateTypeEnum

Enum	Description
CLOSED_1	State of being closed.
INDETERMINATE_2	Indeterminate state.
OPEN_3	State of being open.
UNAVAILABLE_4	No data item reading for this device is available.

The status of the door

7.38 DurationTimeDataType

A length of time in seconds

Table 208 - DurationTimeDataType

Attribute	Value
BrowseName	DurationTimeDataType
IsAbstract	False
Subtype of float	

7.39 DurationValueDataType

The duration of an event in seconds

Table 209 - DurationValueDataType

Attribute	Value
BrowseName	DurationValueDataType
IsAbstract	False
Subtype of string	

7.40 EdgeCountDataType

The number of cutting edges

Table 210 -EdgeCountDataType

Attribute	**Value**
BrowseName	EdgeCountDataType
IsAbstract	False
Subtype of integer	

7.41 EmergencyStopTypeEnum

Enumeration: Emergency Stop status

Table 211 - EmergencyStopTypeEnum

Enum	Description
ARMED_1	The circuit is complete and the device is operating)
TRIGGERED_2	The circuit is open and the device SHALL cease operation).
UNAVAILABLE_3	No data item reading for this device is available.

7.42 EnergyValueDataType

The value of energy

Table 212 - EnergyValueDataType

Attribute	Value
BrowseName	EnergyValueDataType
IsAbstract	False
Subtype of string	

7.43 ExecutionTypeEnum

Enumeration: ExecutionTypeEnum

Table 213 - ExecutionTypeEnum

Enum	Description
ACTIVE_1	
INTERRUPTED_2	
READY_3	
STOPPED_4	
UNAVAILABLE_5	No data item reading for this device is available.

Program execution events

7.44 FeedrateValueDataType

The federate

Table 214 - FeedrateValueDataType

Attribute	Value
BrowseName	FeedrateValueDataType
IsAbstract	False
Subtype of string	

7.45 FirmwareVersionDataType

The firmware version of this sensor

Table 215 - FirmwareVersionDataType

Attribute	**Value**
BrowseName	FirmwareVersionDataType
IsAbstract	False
Subtype of string	

7.46 ForceValueDataType

The magnitude of push or pull

Table 216 - ForceValueDataType

Attribute	Value
BrowseName	ForceValueDataType
IsAbstract	false
Subtype of string	

7.47 GradeDataType

The material for a cutting item

Table 217 -GradeDataType

Attribute	**Value**
BrowseName	GradeDataType
IsAbstract	false
Subtype of string	

7.48 IDDataType

An identifier

Table 218 - IDDataType

Attribute	Value
BrowseName	IDDataType
IsAbstract	False
Subtype of NodeId	

7.49 IndexRangeDataType

A single or range of indexes. A range can be a comma separated set of individual elements as in "1,2,3,4", or as a inclusive range of values as in "1-10" or multiple ranges "1-4,6-10"

Table 219 - IndexRangeDataType

Attribute	Value
BrowseName	IndexRangeDataType
IsAbstract	False
Subtype of string	

7.50 InstanceIdDataType

The instance number of the agent, used for fault tolerance

Table 220 - InstanceIdDataType

Attribute	Value
BrowseName	InstanceIdDataType
IsAbstract	False
Subtype of integer	

7.51 ItemIdDataType

An identifier for the insert

Table 221 - ItemIdDataType

Attribute	Value
BrowseName	ItemIdDataType
IsAbstract	False
Subtype of NMTOKEN	

7.52 InterfaceStateTypeEnum

Enumeration: InterfaceStateTypeEnum

Table 222 - InterfaceStateTypeEnum

Enum	Description
ACTIVE_1	
COMPLETE_2	
FAIL_3	
READY_4	

Interface state

7.53 ItemSourceDataType

The measurement source

Table 223 - ItemSourceDataType

Attribute	Value
BrowseName	ItemSourceDataType
IsAbstract	False
Subtype of string	

7.54 LocationSizeDataType

The number of location units required to hold this tool

Table 224 - LocationSizeDataType

Attribute	Value
BrowseName	LocationSizeDataType
IsAbstract	False
Subtype of integer	

7.55 LocationValueDataType

The tool location

Table 225 - LocationValueDataType

Attribute	Value
BrowseName	LocationValueDataType
IsAbstract	False
Subtype of integer	

7.56 LocationsTypeEnum

Enumeration: LocationsTypeEnum

Table 226 - LocationsTypeEnum

Enum	Description
CRIB_1	
POT_2	
STATION_3	

The type of tool location

7.57 LocusDataType

The location of the cutting item - not yet restricted.

Table 227 - LocusDataType

Attribute	Value
BrowseName	LocusDataType
IsAbstract	False
Subtype of string	

7.58 ManufacturersDataType

A comma delimited list of manufactures

Table 228 - ManufacturersDataType

Attribute	Value
BrowseName	ManufacturersDataType
IsAbstract	False
Subtype of string	

7.59 MassValueDataType

The weight of an object

Table 229 - MassValueDataType

Attribute	Value
BrowseName	MassValueDataType
IsAbstract	False
Subtype of string	

7.60 MaximumCountDataType

A maximum count value

Table 230 - MaximumCountDataType

Attribute	**Value**
BrowseName	MaximumCountDataType
IsAbstract	False
Subtype of integer	

7.61 MaximumDataType

A maximum value

Table 231 - MaximumDataType

Attribute	**Value**
BrowseName	MaximumDataType
IsAbstract	False
Subtype of float	

7.62 MeasurementValueDataType

A measurement value

Table 232 - MeasurementValueDataType

Attribute	**Value**
BrowseName	MeasurementValueDataType
IsAbstract	False
Subtype of float	

7.63 MinimumDataType

A minimum value

Table 233 - MinimumDataType

Attribute	**Value**
BrowseName	MinimumDataType
IsAbstract	False
Subtype of float	

7.64 ModelDataType

The model name

Table 234 - ModelDataType

Attribute	**Value**
BrowseName	ModelDataType
IsAbstract	False

Subtype of string

7.65 NameDataType

A short name for any element

Table 235 - NameDataType

Attribute	**Value**
BrowseName	NameDataType
IsAbstract	False
Subtype of string	

7.66 NativeCodeDataType

A Condition code as defined by the component

Table 236 - NativeCodeDataType

Attribute	Value
BrowseName	NativeCodeDataType
IsAbstract	False
Subtype of string	

7.67 NativeNotificationCodeDataType

A Notification code as defined by the component

Table 237 - NativeNotificationCodeDataType

Attribute	Value
BrowseName	NativeNotifcationCodeDataType
IsAbstract	false
Subtype of string	

7.68 NativeScaleDataType

The multiplier for the native value. Conversion divides by this value

Table 238 - NativeScaleDataType

Attribute	**Value**
BrowseName	NativeScaleDataType
IsAbstract	False
Subtype of float	

7.69 NativeSeverityDataType

The device's severity

Table 239 - NativeSeverityDataType

Attribute	Value
BrowseName	NativeSeverityDataType
IsAbstract	false
Subtype of string	

7.70 NextCalibrationDateDataType

The time the next calibration should be preformed

Table 240 - NextCalibrationDateDataType

Attribute	**Value**
BrowseName	NextCalibrationDateDataType
IsAbstract	False

Subtype of date

7.71 NominalDataType

A nominal value

Table 241 - NominalDataType

Attribute	**Value**
BrowseName	NominalDataType
IsAbstract	False
Subtype of float	

7.72 NotificationCodeTypeEnum

Enumeration: NotificationCodeTypeEnum

Table 242 - NotificationCodeTypeEnum

Enum	Description
CRASH_1	
ESTOP_2	
FAILURE_3	
FAULT_4	
JAM_5	
MATERIAL_6	
MESSAGE_7	An un-interpreted textual notification.
OTHER_8	Unsupported units
OVERLOAD_9	

Types of Notifications

7.73 NotificationDescriptionDataType

The description of the Notification

Table 243 – NotificationDescriptionDataType

Attribute	Value
BrowseName	NotificationDescriptionDataType
IsAbstract	false
Subtype of string	

7.74 OccurrenceTimeDataType

The time a sample occurred

Table 244 - OccurrenceTimeDataType

Attribute	**Value**
BrowseName	OccurrenceTimeDataType
IsAbstract	False
Subtype of DateTime	

7.75 OverlapDataType

The number of additional locations taken by a tool

Table 245 - OverlapDataType

Attribute	**Value**
BrowseName	OverlapDataType
IsAbstract	False
Subtype of integer	

7.76 PathModeTypeEnum

Enumeration: PathModeTypeEnum

Table 246 - PathModeTypeEnum

Enum	Description
INDEPENDENT_1	
MIRROR_2	
SYNCHRONOUS_3	

The actuator state of the component

7.77 PowerStateTypeEnum

Enumeration: PowerStateTypeEnum

Table 247 - PowerStateTypeEnum

Enum	Description
OFF_1	
ON_2	
UNAVAILABLE_3	No data item reading for this device is available.

7.78 ProgramToolNumberDataType

The number referenced in the program for this tool

Table 248 - ProgramToolNumberDataType

Attribute	Value
BrowseName	ProgramToolNumberDataType
IsAbstract	False
Subtype of integer	

7.79 QualifierTypeEnum

Enumeration: QualifierTypeEnum

Table 249 - QualifierTypeEnum

Enum	Description
HIGH_1	
LOW_2	

A qualifier for the condition

7.80 RateDataType

A sample rate in milliseconds per sample

Table 250 - RateDataType

Attribute	Value
BrowseName	RateDataType
IsAbstract	False
Subtype of float	

7.81 ReconditionCountValueDataType

The number of times the cutter has been reconditioned

Table 251 - ReconditionCountValueDataType

Attribute	Value
BrowseName	ReconditionCountValueDataType
IsAbstract	False

Subtype of integer

7.82 RotaryModeTypeEnum

Enumeration: RotaryModeTypeEnum

Table 252 - RotaryModeTypeEnum

Enum	Description
CONTOUR_1	
INDEX_2	
SPINDLE_3	
UNAVAILABLE_4	No data item reading for this device is available.

7.83 SampleRateDataType

The frequency a measurement is sampled expressed as samples per second

Table 253 - SampleRateDataType

Attribute	**Value**
BrowseName	SampleRateDataType
IsAbstract	False
Subtype of float	

7.84 SerialNumberDataType

A serial number for a piece of equipment

Table 254 - SerialNumberDataType

Attribute	**Value**
BrowseName	SerialNumberDataType
IsAbstract	False
Subtype of string	

7.85 SeverityTypeEnum

Enumeration: SeverityTypeEnum

Table 255 - SeverityTypeEnum

Enum	Description
CRITICAL_1	
ERROR_2	
INFORMATION_3	
WARNING_4	

The severity of the notification

7.86 SignificantDigitsValueDataType

The number significant digits

Table 256 - SignificantDigitsValueDataType

Attribute	**Value**
BrowseName	SignificantDigitsValueDataType
IsAbstract	False
Subtype of integer	

7.87 StationDataType

The station id for this device

Table 257 - StationDataType

Attribute	**Value**

BrowseName	StationDataType
IsAbstract	False
Subtype of string	

7.88 SenderDataType

The sender of the message

Table 258 - SenderDataType

Attribute	Value
BrowseName	SenderDataType
IsAbstract	False
Subtype of string	

7.89 SequenceDataType

A sequence number

Table 259 - SequenceDataType

Attribute	Value
BrowseName	SequenceDataType
IsAbstract	False
Subtype of integer	

7.90 SerialNumberDataType

A serial number for a piece of equipment

Table 260 - SerialNumberDataType

Attribute	Value
BrowseName	SerialNumberDataType
IsAbstract	False
Subtype of string	

7.91 SpeedDataType

A speed in RPM or mm/s

Table 261 - SpeedDataType

Attribute	Value
BrowseName	SpeedDataType
IsAbstract	False
Subtype of float	

7.92 TestIndicatorDataType

A debugging flag for testing.

Table 262 - TestIndicatorDataType

Attribute	Value
BrowseName	TestIndicatorDataType
IsAbstract	false
Subtype of Boolean	

7.93 ThreeDimensionalValueDataType

A three dimensional value 'X Y Z' or 'A B C'

Table 263 - ThreeDimensionalValueDataType

Attribute	Value
BrowseName	ThreeDimensionalValueDataType
IsAbstract	False

Subtype of string	

7.94 ToolEventValueDataType

A tool event

Table 264 - ToolEventValueDataType

Attribute	Value
BrowseName	ToolEventValueDataType
IsAbstract	False
Subtype of string	

7.95 ToolGroupDataType

The tool group associated with the tool

Table 265 - ToolGroupDataType

Attribute	**Value**
BrowseName	ToolGroupDataType
IsAbstract	False
Subtype of string	

7.96 ToolIdDataType

The identifier of the tool type

Table 266 - ToolIdDataType

Attribute	**Value**
BrowseName	ToolIdDataType
IsAbstract	False
Subtype of NMTOKEN	

7.97 ToolIdValueDataType

The tool identifier

Table 267 - ToolIdValueDataType

Attribute	Value
BrowseName	ToolIdValueDataType
IsAbstract	False
Subtype of string	

7.98 ToolLifeDirectionTypeEnum

Enumeration: ToolLifeDirectionTypeEnum

Table 268 - ToolLifeDirectionTypeEnum

Enum	**Description**
DOWN_1	
UP_2	

The direction of tool life count

7.99 ToolLifeTypeEnum

Enumeration: ToolLifeTypeEnum

Table 269 - ToolLifeTypeEnum

Enum	**Description**
MINUTES_1	

PART_COUNT_2	The current count of parts produced as represented by the controller. SHALL 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.
WEAR_3	

The direction of tool life count

7.100 ToolLifeValueDataType

The life of the tool in time, wear, or parts

Table 270 - ToolLifeValueDataType

Attribute	**Value**
BrowseName	ToolLifeValueDataType
IsAbstract	False
Subtype of float	

7.101 UnitsExtDataType

An extension point for data item types

Table 271 - UnitsExtDataType

Attribute	**Value**
BrowseName	UnitsExtDataType
IsAbstract	false
Subtype of string	

7.102 VersionDataType

A version number

Table 272 - VersionDataType

Attribute	**Value**
BrowseName	VersionDataType
IsAbstract	False
Subtype of NMTOKEN	

7.103 VibrationValueDataType

The vibration

Table 273 - VibrationValueDataType

Attribute	Value
BrowseName	VibrationValueDataType
IsAbstract	false
Subtype of string	

8 References

8.1 General

The following ReferenceTypes are defined by this companion specification.

8.2 ComputedBy

The ComputedBy reference is a Subtype of NonHierarchicalReferences.

The source of this reference shall be a variable of MTSampleDataItemType or a Subtype of it.

The Target of this reference shall be an instance of AggregateFunctionType.

This reference is used to indicate that the value of the given variable is calculated using the targeted AggregateFunctionType.

8.3 SourceOf

The SourceOf reference is a Subtype of NonHierarchicalReferences.

The source of this reference shall be a variable of MTSampleDataItemType or a Subtype of it.

The Target of this reference shall be an instance of MTComponentType or MTSampleDataItemType or one of their Subtypes.

This reference is used to indicate the actual source associated with the value of the item being reported.

9 Profiles for MTConnect in OPC UA

9.1 Overview

OPC UA has the concept of a Profile (See [UA Part 7]). Profiles are used to create subsets of functionality that a Server exposes or a Client consumes. These Profiles are composed of Conformance Units which are the definition of individual functionality. OPC UA systems are tested and validated against these Profiles and Conformance Units. Conformance Units. Once a profile is developed and released it cannot be modified by OPC standards, The MTConnect standard does not have testing available at this point in time, thus to allow for MTConnect testing to be developed and synchronized with this companion standard, no profiles or conformance units will be defined or released in this version of the companion specification. MTConnect testing and certification is being addressed and thus Profiles and Conformance units will be defined for the next release of this companion specification.

9.2 Test Cases

OPC UA provides tools that can auto generate test cases for information models. These tools do not generate a complete set of test cases, but do provide an initial set of test cases. For this release of the companion specification, these auto generated test cases will be available, but no additional test cases will be defined. Once Profiles and Conformance Units are defined than additional test cases will be generated.