



**MTConnect<sup>®</sup> Standard**  
**Part 4.0 – Assets Information Model**  
**Version 1.6.0**

Prepared for: MTConnect Institute  
Prepared on: July 15, 2020

MTConnect<sup>®</sup> is a registered trademark of AMT - The Association for Manufacturing Technology. Use of *MTConnect* is limited to use as specified on <http://www.mtconnect.org/>.

## MTConnect Specification and Materials

The Association for Manufacturing Technology (AMT) owns the copyright in this *MTConnect* Specification or Material. AMT grants to you a non-exclusive, non-transferable, revocable, non-sublicensable, fully-paid-up copyright license to reproduce, copy and redistribute this *MTConnect* Specification or Material, provided that you may only copy or redistribute the *MTConnect* Specification or Material in the form in which you received it, without modifications, and with all copyright notices and other notices and disclaimers contained in the *MTConnect* Specification or Material.

If you intend to adopt or implement an *MTConnect* Specification or Material in a product, whether hardware, software or firmware, which complies with an *MTConnect* Specification, you shall agree to the *MTConnect* Specification Implementer License Agreement (“Implementer License”) or to the *MTConnect* Intellectual Property Policy and Agreement (“IP Policy”). The Implementer License and IP Policy each sets forth the license terms and other terms of use for *MTConnect* Implementers to adopt or implement the *MTConnect* Specifications, including certain license rights covering necessary patent claims for that purpose. These materials can be found at [www.MTConnect.org](http://www.MTConnect.org), or by contacting <mailto:info@MTConnect.org>.

*MTConnect* Institute and AMT have no responsibility to identify patents, patent claims or patent applications which may relate to or be required to implement a Specification, or to determine the legal validity or scope of any such patent claims brought to their attention. Each *MTConnect* Implementer is responsible for securing its own licenses or rights to any patent or other intellectual property rights that may be necessary for such use, and neither AMT nor *MTConnect* Institute have any obligation to secure any such rights.

This Material and all *MTConnect* Specifications and Materials are provided “as is” and *MTConnect* Institute and AMT, and each of their respective members, officers, affiliates, sponsors and agents, make no representation or warranty of any kind relating to these materials or to any implementation of the *MTConnect* Specifications or Materials in any product, including, without limitation, any expressed or implied warranty of noninfringement, merchantability, or fitness for particular purpose, or of the accuracy, reliability, or completeness of information contained herein. In no event shall *MTConnect* Institute or AMT be liable to any user or implementer of *MTConnect* Specifications or Materials for the cost of procuring substitute goods or services, lost profits, loss of use, loss of data or any incidental, consequential, indirect, special or punitive damages or other direct damages, whether under contract, tort, warranty or otherwise, arising in any way out of access, use or inability to use the *MTConnect* Specification or other *MTConnect* Materials, whether or not they had advance notice of the possibility of such damage.

## Table of Contents

<b>1</b>	<b>Purpose of This Document</b>	<b>2</b>
<b>2</b>	<b>Terminology and Conventions</b>	<b>3</b>
2.1	Glossary . . . . .	3
2.2	Acronyms . . . . .	8
2.3	MTCConnect References . . . . .	8
<b>3</b>	<b>MTCConnect Assets</b>	<b>9</b>
3.1	Overview . . . . .	9
3.2	MTCConnectAssets . . . . .	10
3.2.1	MTCConnectAssets Header . . . . .	10
3.2.1.1	Header Attributes . . . . .	11
3.2.2	Assets . . . . .	13
3.2.3	Asset . . . . .	13
3.2.3.1	Common Asset Attributes . . . . .	14
3.2.3.2	Common Asset Elements . . . . .	16
<b>4</b>	<b>MTCConnect Assets Architecture</b>	<b>17</b>
4.1	Agent Asset Storage . . . . .	17
4.2	Asset Protocol . . . . .	18
4.2.1	Asset by assetId . . . . .	18
4.2.2	Asset for a Given Type . . . . .	19
4.2.3	Assets Including Removed Assets . . . . .	19
4.2.4	Assets for a Piece of Equipment . . . . .	20
<b>5</b>	<b>Extensions to Part 2.0 - Devices Information Model</b>	<b>21</b>
5.1	Data Item Types added for EVENT Category . . . . .	21
5.1.1	ASSET_CHANGED Data Item Type . . . . .	21
5.1.2	ASSET_REMOVED Data Item Type . . . . .	22
<b>6</b>	<b>Extensions to Part 3.0 - Streams Information Model</b>	<b>23</b>
6.1	AssetChanged Extension to Events . . . . .	23
6.1.1	AssetChanged event Attributes . . . . .	24
6.2	AssetRemoved Extension to Events . . . . .	24
6.2.1	AssetRemoved Attributes . . . . .	25
	<b>Appendices</b>	<b>26</b>
A	Bibliography . . . . .	26

## Table of Figures

<b>Figure 1: MTConnectAssets Schema</b> . . . . .	10
<b>Figure 2: MTConnectAssets Header</b> . . . . .	11
<b>Figure 3: Asset Schema</b> . . . . .	14
<b>Figure 4: Description Schema</b> . . . . .	16
<b>Figure 5: MTConnect Assets storage as First in First Out</b> . . . . .	17
<b>Figure 6: MTConnect Assets storage as Key/Value pairs</b> . . . . .	18
<b>Figure 7: AssetChanged Schema</b> . . . . .	23
<b>Figure 8: AssetRemoved Schema</b> . . . . .	24

## List of Tables

<b>Table 1: MTConnectAssets Header</b> . . . . .	12
<b>Table 2: MTConnect Assets Element</b> . . . . .	13
<b>Table 3: MTConnect Asset Element</b> . . . . .	13
<b>Table 4: Attributes for Asset</b> . . . . .	14
<b>Table 5: Elements for Asset</b> . . . . .	16
<b>Table 6: DataItem Type for EVENT category</b> . . . . .	21
<b>Table 7: Attributes for AssetChanged</b> . . . . .	24
<b>Table 8: Attributes for AssetRemoved</b> . . . . .	25

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

2 This document, *MTConnect Standard: Part 4.0 - Assets Information Model* of the MTCon-  
3 nect Standard, details information that is common to all types of *MTConnect Assets*. Part  
4 4.0 and its sub-parts of the MTConnect Standard provide semantic models for entities that  
5 are used in the manufacturing process, but are not considered to be a piece of equipment.  
6 These entities are defined as *MTConnect Assets*. These *Assets* may be removed from a  
7 piece of equipment without detriment to the function of the equipment and can be associ-  
8 ated with other pieces of equipment during their lifecycle. The data associated with these  
9 *Assets* may be retrieved from multiple sources that are each responsible for providing their  
10 knowledge of the *Asset*.

## 11 2 Terminology and Conventions

12 Refer to Section 2 of *MTConnect Standard Part 1.0 - Overview and Fundamentals* for a  
13 dictionary of terms, reserved language, and document conventions used in the MTConnect  
14 Standard.

### 15 2.1 Glossary

#### 16 CDATA

17 General meaning:

18 An abbreviation for Character Data.

19 CDATA is used to describe a value (text or data) published as part of an XML ele-  
20 ment.

21 For example, "This is some text" is the CDATA in the XML element:

```
22 <Message ...>This is some text</Message>
```

23 Appears in the documents in the following form: CDATA

#### 24 NMTOKEN

25 The data type for XML identifiers.

26 Note: The identifier must start with a letter, an underscore "\_" or a colon. The next  
27 character must be a letter, a number, or one of the following ".", "-", "\_", ":". The  
28 identifier must not have any spaces or special characters.

29 Appears in the documents in the following form: NMTOKEN.

#### 30 XML

31 Stands for eXtensible Markup Language.

32 XML defines a set of rules for encoding documents that both a human-readable and  
33 machine-readable.

34 XML is the language used for all code examples in the MTConnect Standard.

35 Refer to <http://www.w3.org/XML> for more information about XML.

#### 36 *Agent*

37 Refers to an MTConnect Agent.

38 Software that collects data published from one or more piece(s) of equipment, orga-  
39 nizes that data in a structured manner, and responds to requests for data from client

40 software systems by providing a structured response in the form of a *Response Doc-*  
 41 *ument* that is constructed using the *semantic data models* defined in the Standard.

42 Appears in the documents in the following form: *Agent*.

### 43 ***Asset***

44 General meaning:

45 Typically referred to as an *MTCConnect Asset*.

46 An *MTCConnect Asset* is something that is used in the manufacturing process, but is  
 47 not permanently associated with a single piece of equipment, can be removed from  
 48 the piece of equipment without compromising its function, and can be associated  
 49 with other pieces of equipment during its lifecycle.

50 Used to identify a storage area in an *Agent*:

51 See description of *buffer*.

52 Used as an *Information Model*:

53 Used to describe an *Information Model* that contains the rules and terminology that  
 54 describe information that may be included in electronic documents representing *MT-*  
 55 *Connect Assets*.

56 The *Asset Information Models* defines the structure for the *Assets Response Docu-*  
 57 *ment*.

58 Individual *Information Models* describe the structure of the *Asset Documents* rep-  
 59 resent each type of *MTCConnect Asset*. Appears in the documents in the following  
 60 form: *Asset Information Models* or (asset type) *Information Model*.

61 Used when referring to an *MTCConnect Asset*:

62 Refers to the information related to an *MTCConnect Asset* or a group of *MTCConnect*  
 63 *Assets*.

64 Appears in the documents in the following form: *Asset* or *Assets*.

65 Used as an XML container or element:

66 ● When used as an XML container that consists of one or more types of `Asset`  
 67 XML elements.

68 Appears in the documents in the following form: `Assets`.

69 ● When used as an abstract XML element. It is replaced in the XML document  
 70 by types of `Asset` elements representing individual *Asset* entities.

71 Appears in the documents in the following form: `Asset`.

72 Used to describe information stored in an *Agent*:

73 Identifies an electronic document published by a data source and stored in the *assets*  
 74 *buffer* of an *Agent*.



75 Appears in the documents in the following form: *Asset Document*.

76 Used as an XML representation of an *MTCConnect Response Document*:

77 Identifies an electronic document encoded in XML and published by an *Agent* in  
78 response to a *Request* for information from a client software application relating to  
79 *MTCConnect Assets*.

80 Appears in the documents in the following form: `MTCConnectAssets`.

81 Used as an *MTCConnect Request*:

82 Represents a specific type of communications request between a client software ap-  
83 plication and an *Agent* regarding *MTCConnect Assets*.

84 Appears in the documents in the following form: *Asset Request*.

85 Used as part of an *HTTP Request*:

86 Used in the path portion of an *HTTP Request Line*, by a client software applica-  
87 tion, to initiate an *Asset Request* to an *Agent* to publish an `MTCConnectAssets`  
88 document.

89 Appears in the documents in the following form: `asset`.

#### 90 ***Asset Document***

91 An electronic document published by an *Agent* in response to a *Request* for infor-  
92 mation from a client software application relating to *Assets*.

#### 93 ***buffer***

94 General meaning:

95 A section of an *Agent* that provides storage for information published from pieces  
96 of equipment.

97 Used relative to *Streaming Data*:

98 A section of an *Agent* that provides storage for information relating to individual  
99 pieces of *Streaming Data*.

100 Appears in the documents in the following form: *buffer*.

101 Used relative to *MTCConnect Assets*:

102 A section of an *Agent* that provides storage for *Asset Documents*.

103 Appears in the documents in the following form: *assets buffer*.

#### 104 ***Data Entity***

105 A primary data modeling element that represents all elements that either describe  
106 data items that may be reported by an *Agent* or the data items that contain the actual  
107 data published by an *Agent*.

108 Appears in the documents in the following form: *Data Entity*.

109 **Document**

110 General meaning:

111 A piece of written, printed, or electronic matter that provides information.

112 Used to represent an *MTCConnect Document*:

113 Refers to printed or electronic document(s) that represent a *Part(s)* of the MTCConnect Standard.

115 Appears in the documents in the following form: *MTCConnect Document*.

116 Used to represent a specific representation of an *MTCConnect Document*:

117 Refers to electronic document(s) associated with an *Agent* that are encoded using XML; *Response Documents* or *Asset Documents*.

119 Appears in the documents in the following form: *MTCConnect XML Document*.

120 Used to describe types of information stored in an *Agent*:

121 In an implementation, the electronic documents that are published from a data source and stored by an *Agent*.

123 Appears in the documents in the following form: *Asset Document*.

124 Used to describe information published by an *Agent*:

125 A document published by an *Agent* based upon one of the *semantic data models* defined in the MTCConnect Standard in response to a request from a client.

127 Appears in the documents in the following form: *Response Document*.

128 **Equipment Metadata**

129 See *Metadata*

130 **HTTP Request**

131 In the MTCConnect Standard, a communications command issued by a client software application to an *Agent* requesting information defined in the *HTTP Request Line*.

134 Appears in the documents in the following form: *HTTP Request*.

135 **HTTP Request Line**

136 In the MTCConnect Standard, the first line of an *HTTP Request* describing a specific *Response Document* to be published by an *Agent*.

138 Appears in the documents in the following form: *HTTP Request Line*.

139 ***Information Model***

140 The rules, relationships, and terminology that are used to define how information is  
141 structured.

142 For example, an information model is used to define the structure for each *MTCConnect Response Document*; the definition of each piece of information within those  
143 documents and the relationship between pieces of information.  
144

145 Appears in the documents in the following form: *Information Model*.

146 ***MTCConnect Document***

147 See *Document*.

148 ***MTCConnect Request***

149 A communication request for information issued from a client software application  
150 to an *Agent*.

151 Appears in the documents in the following form: *MTCConnect Request*.

152 ***MTCConnect XML Document***

153 See *Document*.

154 ***Request***

155 A communications method where a client software application transmits a message  
156 to an *Agent*. That message instructs the *Agent* to respond with specific information.

157 Appears in the documents in the following form: *Request*.

158 ***Response Document***

159 See *Document*.

160 ***semantic data model***

161 A methodology for defining the structure and meaning for data in a specific logical  
162 way.

163 It provides the rules for encoding electronic information such that it can be inter-  
164 preted by a software system.

165 Appears in the documents in the following form: *semantic data model*.

166 ***Streaming Data***

167 The values published by a piece of equipment for the *Data Entities* defined by the  
168 *Equipment Metadata*.

169 Appears in the documents in the following form: *Streaming Data*.

170 ***Valid Data Value***

171 One or more acceptable values or constrained values that can be reported for a *Data*  
172 *Entity*.

173 Appears in the documents in the following form: *Valid Data Value(s)*.

174 ***XML Schema***

175 In the MTConnect Standard, an instantiation of a schema defining a specific docu-  
176 ment encoded in XML.

177 **2.2 Acronyms**

178 ***AMT***

179 The Association for Manufacturing Technology

180 **2.3 MTConnect References**

181 [MTConnect Part 1.0] *MTConnect Standard Part 1.0 - Overview and Fundamentals*. Ver-  
182 sion 1.5.0.

183 [MTConnect Part 3.0] *MTConnect Standard: Part 3.0 - Streams Information Model*. Ver-  
184 sion 1.5.0.

185 [MTConnect Part 4.0] *MTConnect Standard: Part 4.0 - Assets Information Model*. Ver-  
186 sion 1.5.0.

187 [MTConnect Part 4.1] *MTConnect Standard: Part 4.1 - Cutting Tools*. Version 1.5.0.

## 188 3 MTConnect Assets

### 189 3.1 Overview

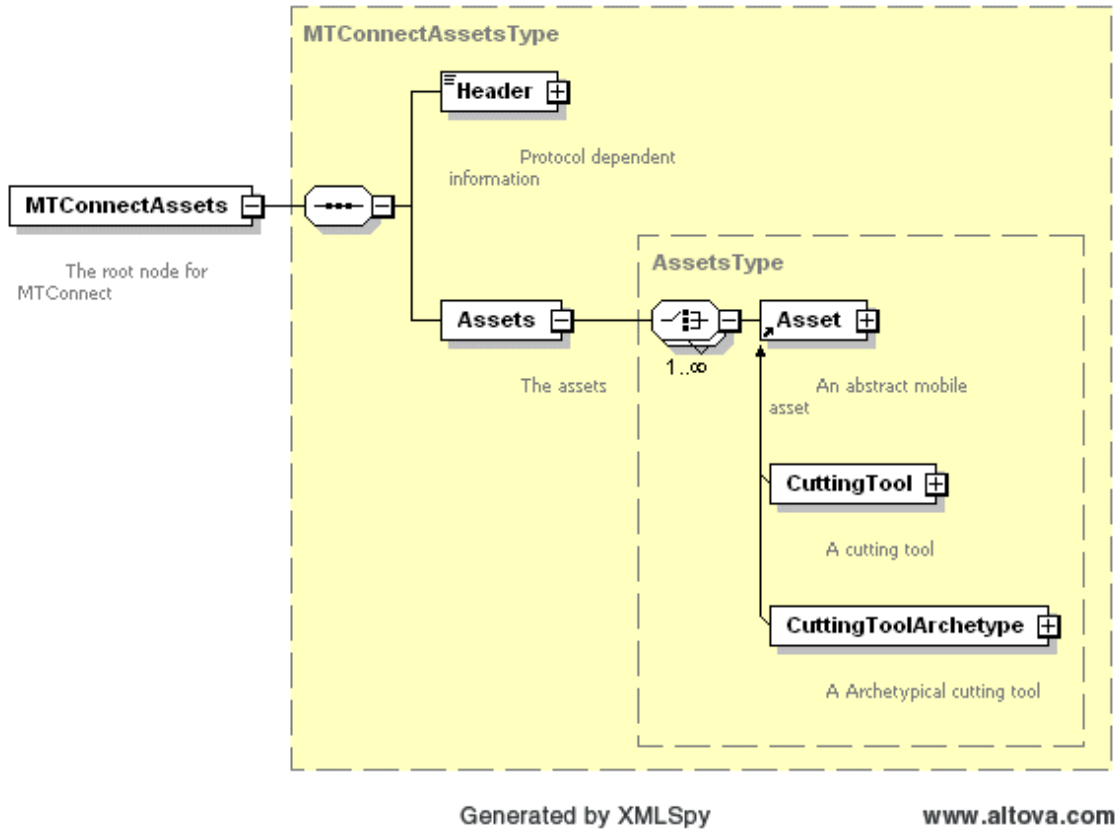
190 The MTConnect Standard supports a simple distributed storage mechanism that allows ap-  
191 plications and equipment to share and exchange complex information models in a similar  
192 way to a distributed data store. The *Asset Information Model* associates each electronic  
193 MTConnectAssets document with a unique identifier and allows for some predefined  
194 mechanisms to find, create, request, updated, and delete these electronic documents in a  
195 way that provides for consistency across multiple pieces of equipment.

196 The protocol provides a limited mechanism of accessing *MTConnect Assets* using the fol-  
197 lowing properties: `assetId`, *Asset* type (element name of *Asset* root), and the piece of  
198 equipment associated with the *Asset*. These access strategies will provide the following  
199 services and answer the following questions: What *Assets* are from a particular piece of  
200 equipment? What are the *Assets* of a particular type? What *Assets* is stored for a given  
201 `assetId`?

202 Although these mechanisms are provided, an *Agent* should not be considered a data store  
203 or a system of reference. The *Agent* is providing an ephemeral storage capability that will  
204 temporarily manage the data for applications wishing to communicate and manage data as  
205 need-ed by the various processes. An application cannot rely on an *Agent* for long term  
206 persistence or durability since the *Agent* is only required to temporarily store the *Asset*  
207 data and may require an-other system to provide the source data upon initialization. An  
208 *Agent* is always providing the best-known equipment centric view of the data given the  
209 limitations of that piece of equipment.

210 Note: Currently only cutting tools have been addressed by the MTConnect Standard  
211 and other MTConnect Assets will be defined in later versions of the Standard.

212 **3.2 MTConnectAssets**



**Figure 1: MTConnectAssets Schema**

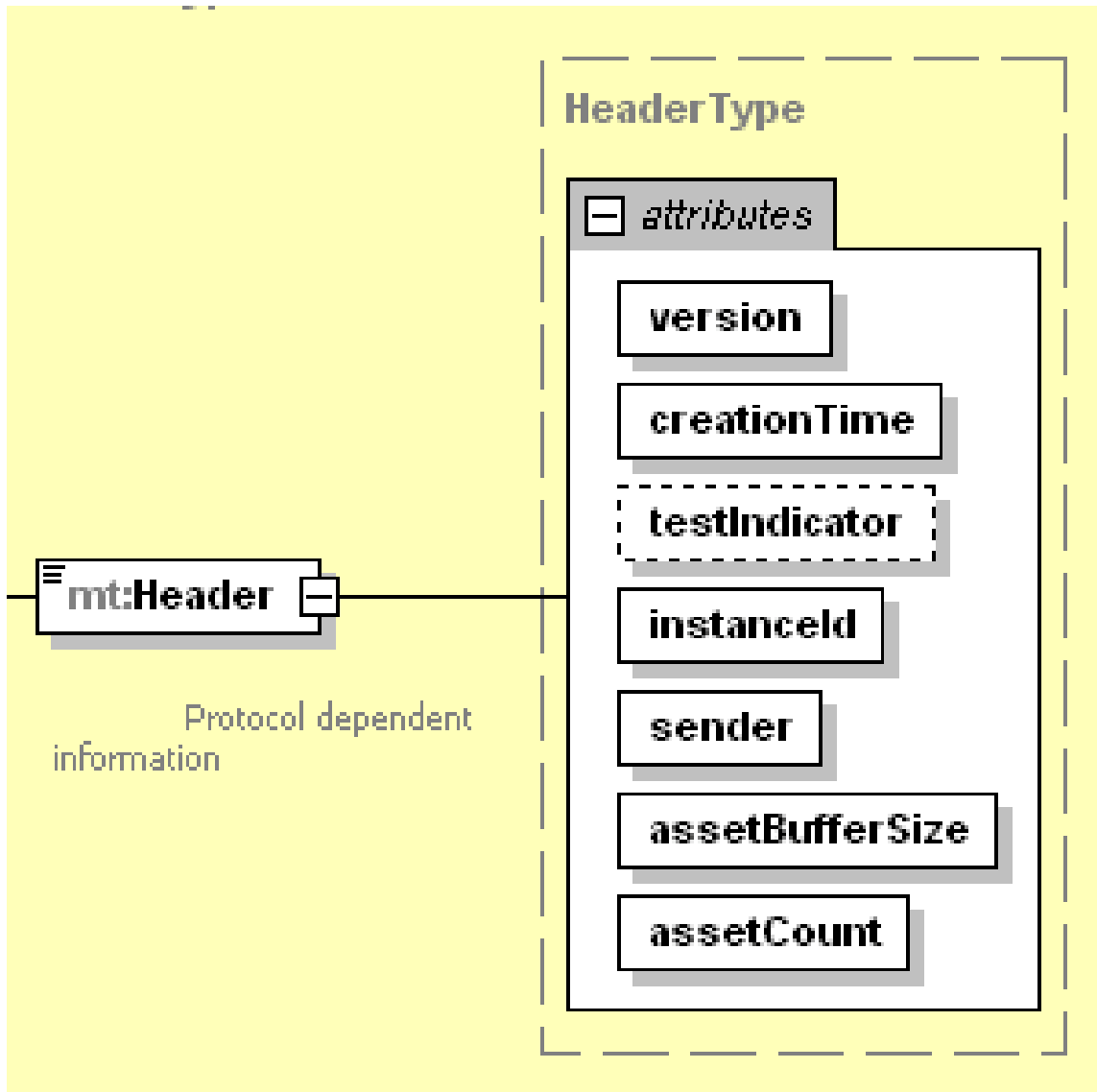
213 At the top level of the `MTConnectAssets` document is a standard header, as stated in  
 214 *MTConnect Standard Part 1.0 - Overview and Fundamentals*, and one or more *MTConnect*  
 215 *Assets*. Each *Asset* is required to have an `assetId` that serves as a unique identifier of  
 216 that *Asset*. `assetId` allows an application to request the *Asset* data from an *Agent*.

217 In the remaining *Part 4.x* sub-part documents of *MTConnect Assets*, various types of *As-*  
 218 *sets* will be introduced such as cutting tools and other *Asset* types. Currently only cutting  
 219 tools have been defined in *MTConnect Standard: Part 4.1 - Cutting Tools*.

220 **3.2.1 MTConnectAssets Header**

221 The `MTConnectAssets` header is where the protocol sequence information **MUST** be  
 222 provided. The *XML Schema* in *Figure 2* represents the structure of the `MTConnectAs-`  
 223 `sets` header showing the attributes defined for `MTConnectAssets`.

224 Refer to *MTCConnect Standard Part 1.0 - Overview and Fundamentals* for more informa-  
 225 tion on headers.



**Figure 2:** MTCConnectAssets Header

226 **3.2.1.1 Header Attributes**

227 *Table 1* defines the attributes used to provide information for an `MTCConnectAssets`  
 228 header.

**Table 1:** MTConnectAssets Header

Attribute	Description	Occurrence
version	The protocol version number. This is the <i>major</i> and <i>minor</i> version number of the MTConnect Standard being used. For example, if the version number of the Standard used is 10.21.33, the version will be 10.21.  version is a required attribute.	1
creationTime	The time the response was created.  creationTime is a required attribute.	1
testIndicator	Optional flag that indicates the system is operating in test mode. This data is only for testing and indicates that the data is simulated.  testIndicator is an optional attribute.	0..1
instanceId	A number indicating which invocation of the <i>Agent</i> . This is used to differentiate between separate instances of the <i>Agent</i> . This value <b>MUST</b> have a maximum value of $2^{64} - 1$ and <b>MUST</b> be stored in an unsigned 64-bit integer.  instanceId is a required attribute.	1
sender	The <i>Agent</i> identification information.  sender is a required attribute.	1
assetBufferSize	The maximum number of <i>MTConnect Assets</i> that will be retained by the <i>Agent</i> . The assetBufferSize <b>MUST</b> be an unsigned positive integer value with a maximum value of $2^{32} - 1$ .  assetBufferSize is a required attribute.	1
assetCount	The total number of <i>MTConnect Assets</i> in an <i>Agent</i> . This <b>MUST</b> be an unsigned positive integer value with a maximum value of $2^{32} - 1$ . This value <b>MUST NOT</b> be greater than assetBufferSize.  assetCount is a required attribute.	1



**Example 1: MTConnectAssets Header Example**

```

229 1 <Header creationTime="2010-03-13T07:59:11+00:00"
230 2     sender="localhost" instanceId="1268463594"
231 3     assetBufferSize="1024" version="1.1"
232 4     assetCount="12" />

```

**233 3.2.2 Assets**

234 `Assets` is an XML container used to group information about various *MTConnect Asset*  
235 types. `Assets` contains one or more `Asset` XML elements.

**Table 2: MTConnect Assets Element**

Element	Description	Occurrence
<code>Assets</code>	An XML container that consists of one or more types of <code>Asset</code> XML elements.	0..1

**236 3.2.3 Asset**

237 An `Asset` XML element is a container type XML element used to organize information  
238 describing an entity that is not a piece of equipment. `Asset` is an abstract type XML  
239 element and will never appear directly in the MTConnect XML document. As an abstract  
240 type XML element, `Asset` will be replaced in the XML document by specific *MTConnect*  
241 `Asset` type.

**Table 3: MTConnect Asset Element**

Element	Description	Occurrence
<code>Asset</code>	An abstract XML element. Replaced in the XML document by types of <code>Asset</code> elements representing entities that are not pieces of equipment.  There can be multiple types of <code>Asset</code> XML elements in the document.	1..*

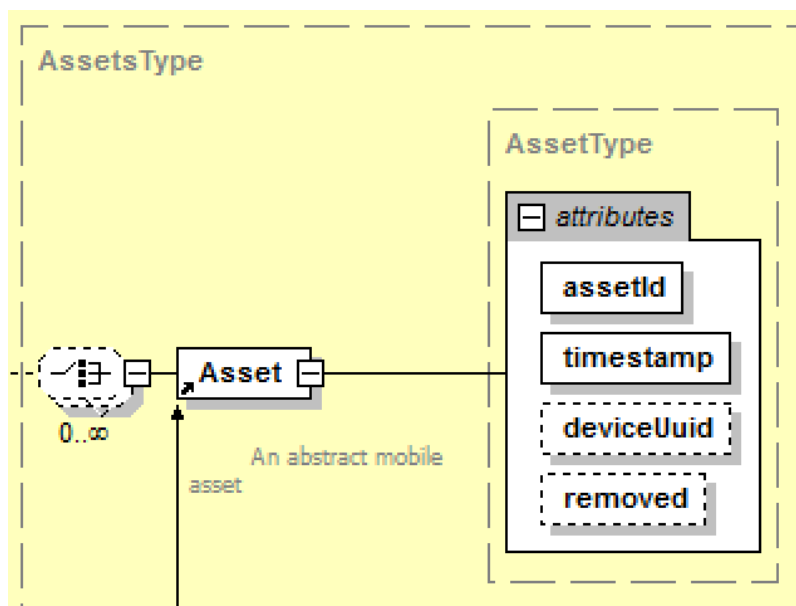
242 There are various types of entities or `Asset` types. Each type of `Asset` is described in sub-  
243 parts of *MTConnect Standard: Part 4.0 - Assets Information Model*. These sub-parts are

244 designated by a *Part 4.x* document number. Currently only the *MTCConnect Asset* type of  
 245 cutting tools has been defined in *MTCConnect Standard: Part 4.1 - Cutting Tools*.

246 For all *MTCConnect Asset* types there are some common attributes and elements that apply  
 247 to all of them. The following defines these common attributes and elements.

248 **3.2.3.1 Common Asset Attributes**

249 The *XML Schema* in *Figure 3* represents the structure of *Asset* showing the attributes  
 250 defined for *Asset*.



**Figure 3:** Asset Schema

251 *Table 4* defines the attributes that are used to provide information for the *Asset* element.

**Table 4:** Attributes for Asset

Attribute	Description	Occurrence
assetId	The unique identifier for the <i>MTCConnect Asset</i> . The identifier <b>MUST</b> be unique with respect to all other <i>Assets</i> in an <i>MTCConnect</i> installation. The identifier <b>SHOULD</b> be globally unique with respect to all other <i>Assets</i> .  assetId is a required attribute.	1

Continuation of Table 4		
Attribute	Description	Occurrence
timestamp	The time this <i>MTCConnect Asset</i> was last modified. Always given in UTC. The <code>timestamp</code> <b>MUST</b> be provided in UTC (Universal Time Coordinate, also known as GMT). This is the time the <i>Asset</i> data was last modified.  <code>timestamp</code> is a required attribute.	1
deviceUuid	The piece of equipments UUID that supplied this data. This is an optional element references to the UUID attribute given in the <code>Device</code> element. This can be any series of numbers and letters as defined by the XML type <code>NMTOKEN</code> .	0..1
removed	This is an optional attribute that is an indicator that the <i>MTCConnect Asset</i> has been removed from the piece of equipment. If the <i>Asset</i> is marked as removed, it will not be visible to the client application unless the <code>=true</code> parameter is provided in the URL. If this attribute is not present it <b>MUST</b> be assumed to be false. The value is an <code>xsi:boolean</code> type and <b>MUST</b> be <code>true</code> or <code>false</code> .	0..1

252 All *MTCConnect Assets* **MUST** have a unique value for `assetId` and it **SHOULD** be  
253 globally unique, such as a RFC 4122 UUID.

254 The following attributes **MUST** be provided and are common to all *MTCConnect Asset*  
255 types: the `assetId` attribute providing the unique identifier for the *Asset*, and the `times-`  
256 `tamp` providing the time the *Asset* was inserted or updated. A `removed` flag that if `true`  
257 indicates the *Asset* has been removed (deleted) from the equipment is optional, however  
258 the *Asset* will still be available if requested directly or a request is made that includes  
259 *removed Assets*.

260 An `MTCConnectAssets` document contains information pertaining to something that is  
261 not a direct component of the piece of equipment and can be relocated to another piece  
262 of equipment or location during its lifecycle. The `Asset` will contain data that will be  
263 changed as a unit, meaning that at any given point in time the latest version of the complete  
264 state for this *Asset* will be provided.

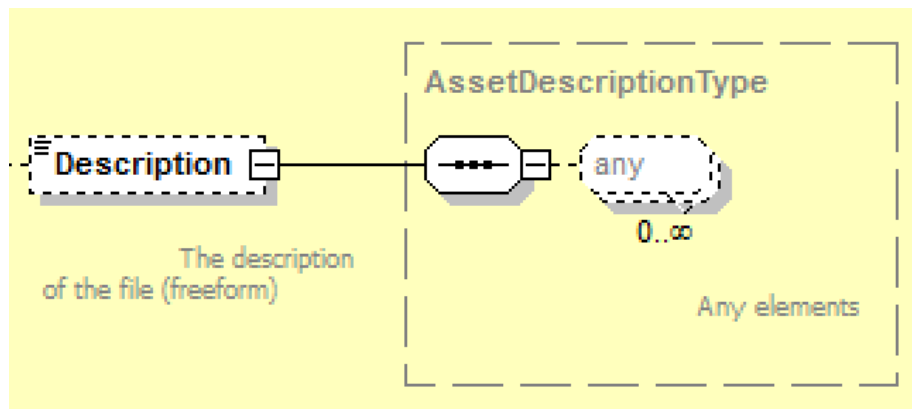
265 Each piece of equipment or location may have a different view of this *Asset* and it is

266 the responsibility of an application to collect and determine the aggregate information  
 267 and keep a historical record if required. An *Agent* will allow any application or other  
 268 equipment to request this information. The piece of equipment **MUST** supply the latest  
 269 and most accurate information regarding a given *Asset*.

270 **3.2.3.2 Common Asset Elements**

271 The element *Description* is the only element common to all *Asset* types.

272 The *XML Schema* in *Figure 4* represents the structure of *Description*.



**Figure 4:** Description Schema

273 *Table 5* defines the elements that are used to provide information for *Asset*.

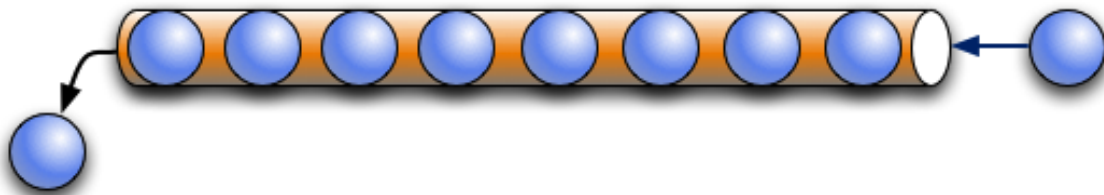
**Table 5:** Elements for Asset

Elements	Description	Occurrence
Description	An optional element that can contain any descriptive content. This can contain configuration information and manufacturer specific details. This element is defined to contain mixed content and XML elements can be added to extend the descriptive semantics of MTConnect Standard.	0..1

## 274 4 MTConnect Assets Architecture

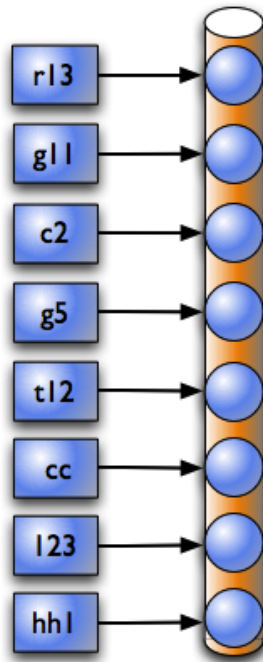
### 275 4.1 Agent Asset Storage

276 The *Agent* stores *MTConnect Assets* in a similar fashion as the *Agent* data storage de-  
 277 scribed in *MTConnect Standard Part 1.0 - Overview and Fundamentals*. The storage of  
 278 information is contained in the *asset buffer*. The *Agent* provides a limited number of *As-*  
 279 *sets* that can be stored at one time and uses the same method of pushing out the oldest  
 280 *Asset* when the *asset buffer* is full. The *asset buffer* size for the *Asset* storage is maintained  
 281 separately from the *Sample*, *Event*, and *Condition* storage.



**Figure 5:** MTConnect Assets storage as First in First Out

282 *MTConnect Assets* also behave like a key/value in memory database. In the case of the  
 283 *Asset*, the key is the `assetId` and the value is the XML document describing the *Asset*.  
 284 The key can be any string of letters, punctuation or digits and represent the domain specific  
 285 coding scheme for their assets. Each *Asset* type will have a recommended way to construct  
 286 a unique `assetId`, for example, a cutting tool **SHOULD** be identified by the tool ID and  
 287 serial number as a composed synthetic identifier.



**Figure 6:** MTConnect Assets storage as Key/Value pairs

288 As in *Figure 6* , each of the *Assets* is referred to by their key. The key is independent of  
 289 the order in the *asset buffer* storage.

## 290 4.2 Asset Protocol

291 MTConnect Standard provides methods to retrieve an *MTConnect Asset* or a set of *Assets*  
 292 given various criteria. These criteria are as follows: The `assetId`, the *Asset* type as de-  
 293 fined by the name of the *Asset*'s topmost element, and the originating piece of equipment.

294 The URL format is similar to the `probe` and `sample` structure. Reference each `as-`  
 295 `setId` directly to request an *MTConnect Asset* by `assetId`.

### 296 4.2.1 Asset by assetId

#### Example 2: Asset by assetId Example

```
297 1 url: http://example.com/asset/e39d23ba-ef2d-
298 2     11e6-b12c15028cfe91a82ef
```

299 *Example 2* returns the `MTConnectAssets` document for *Asset* `e39d23ba-ef2d-`  
 300 `11e6-b12c-28cfe91a82ef`

301 Request multiple *Assets* by each `assetId`:

**Example 3: Assets by assetId Example**

302 1 `url: http://example.com/asset/e39d23ba-ef2d-11e6-b12c155;`  
 303 2 `8cfe91a82ef;e46d5256-ef2d-11e6-96aa-28cfe91a82ef`

304 *Example 3* returns the `MTConnectAssets` document for *Assets* `e39d23ba-ef2d-`  
 305 `11e6-b12c-28cfe91a82ef` and `e46d5256-ef2d-11e6-96aa-28cfe91a82ef`.

306 Request for all the *Assets* in the *Agent*:

**Example 4: Get all Assets Example**

307 1 `url: http://example.com/assets`

308 *Example 4* returns all available *MTConnect Assets* in the *Agent*. The *Agent* **MAY** return  
 309 a limited set if there are too many *Asset* records. The *Assets* **MUST** be added to the  
 310 beginning with the most recently modified *Asset*.

### 311 4.2.2 Asset for a Given Type

**Example 5: Asset for a Given Type Example**

312 1 `url: http://example.com/assets?type="CuttingTool"`

313 *Example 5* returns all available `CuttingTool Assets` from the *Agent* of the type `Cut-`  
 314 `tingTool`. The *Agent* **MAY** return a limited set if there are too many *Asset* records. The  
 315 *Assets* **MUST** be added to the beginning with the most recently modified assets.

316 Request for all *Assets* of a given type in the *Agent* up to a maximum count:

**Example 6: Asset for a Given Type with Maximum count Example**

317 1 `url: http://example.com/assets?type="CuttingTool"`

318 *Example 6* returns all available `CuttingTool Assets` from the *Agent*. The *Agent* **MUST**  
 319 return up to 1000 *Assets* beginning with the most recently modified *Assets* if they exist.

### 320 4.2.3 Assets Including Removed Assets

**Example 7: Assets Including Removed Assets Example**

321 1 `url: http://example.com/assets?type=CuttingTool&removed=true`

322 *Example 7* returns all available `CuttingTool Assets` from the *Agent*. With the removed  
 323 flag, *Assets* that have been removed but are included in the result set.

#### 324 4.2.4 Assets for a Piece of Equipment

325 If no `assetId` is provided with a general *Assets* request, it would be as shown in *Exam-*  
 326 *ple 8*:

##### Example 8: Assets For a Piece of Equipment Example

327 1 url: `http://example.com/Mill123/assets`

328 All *MTCConnect Assets* will be provided for that piece of equipment (`Device`) up to the  
 329 *Agent*'s maximum count or as specified with the count parameter. These *Assets* will be  
 330 returned starting from the newest to oldest list.

331 Any of the previous constraints can also be applied to the request, for example, to get all  
 332 the `CuttingTool` instances for a given piece of equipment:

##### Example 9: Assets For a Piece of Equipment For a Given Type Example

333 1 url: `http://example.com/Mill123/asset/`  
 334 2 `?type=CuttingTool&count=100`

335 The request in *Example 9* will get the newest 100 `Cutting Tool Instance Assets` from the  
 336 *Agent* for `Mill123`. Similarly:

##### Example 10: Assets For a Piece of Equipment For a Given Type Example 2

337 1 url: `http://example.com/Mill123/asset/`  
 338 2 `?type=CuttingToolArchetype`

339 *Example 10* will provide all `Cutting Tool Archetype Assets` with the `deviceUuid` of  
 340 `Mill123`.



## 341 5 Extensions to Part 2.0 - Devices Information Model

342 This document will add the following data item types to support change notification when  
 343 an *MTCConnect Asset* is added or updated. The data item **MUST** be placed in the `DataItems`  
 344 container associated with `Device`. The `Device` **MUST** be the piece of equipment that  
 345 is supplying the asset data.

### 346 5.1 Data Item Types added for EVENT Category

**Table 6:** DataItem Type for EVENT category

DataItem Type SubType	Description
ASSET_CHANGED	The event generated when an asset is added or changed. <code>AssetChanged</code> <b>MUST</b> be discrete and the value of the <code>DataItem</code> 's <code>discrete</code> attribute <b>MUST</b> be <code>TRUE</code> .
ASSET_REMOVED	The value of the <code>CDATA</code> for the event <b>MUST</b> be the <code>assetId</code> of the asset that has been removed. The asset will still be visible if requested with the <code>includeRemoved</code> parameter as described in the protocol section. When assets are removed they are not moved to the beginning of the most recently modified list.

#### 347 5.1.1 ASSET\_CHANGED Data Item Type

348 When an *MTCConnect Asset* is added or modified, an `AssetChanged` event **MUST** be  
 349 published to inform an application that new asset data is available. The application can  
 350 request the new asset data from the piece of equipment at that time. Every time the asset  
 351 data is modified an `AssetChanged` event will be published. Since the asset data is a  
 352 complete electronic document, the system will publish a single `AssetChanged` event  
 353 for the entire set of changes.

354 The asset data **MUST** remain constant until the `AssetChanged` event is published.  
 355 Once it is published the data **MUST** change to reflect the new content at that instant.  
 356 The timestamp of the asset will reflect the time the last change was made to the asset data.

### 357 5.1.2 ASSET\_REMOVED Data Item Type

358 When an *MTConnect Asset* has been removed from an *Agent*, or marked as removed, an  
359 `AssetRemoved` event **MUST** be generated in a similar way to the `AssetChanged`  
360 event. The `CDATA` of the `AssetRemoved` event **MUST** contain the `assetId` that was  
361 just removed.

362 Every time an *MTConnect Asset* is modified or added it will be moved to the beginning  
363 of the *asset buffer* and become the newest *Asset*. As the *asset buffer* fills up, the oldest  
364 *Asset* will be pushed out and its information will be removed. The *MTConnect Standard*  
365 does not specify the maximum size of the *asset buffer*, and if the implementation desires,  
366 permanent storage **MAY** be used to store the *Assets*. A value of 4,294,967,296 or  $2^{32}$  can  
367 be given to indicate unlimited storage.

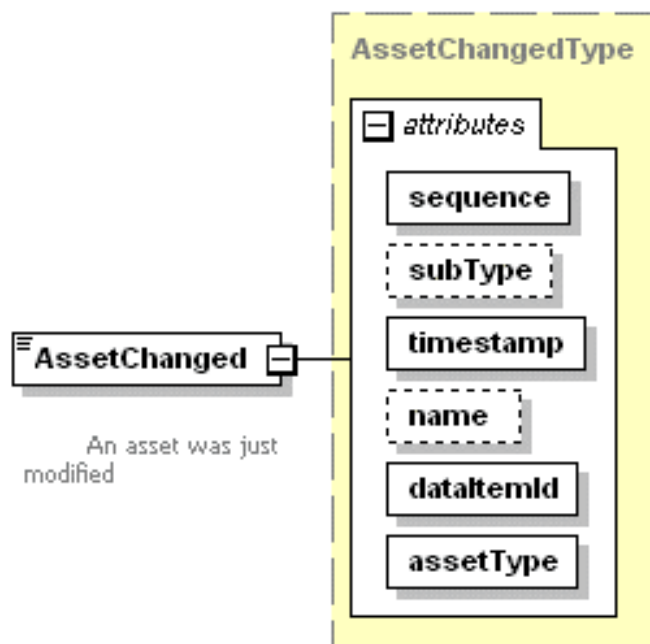
368 There is no requirement for persistent *Asset* storage. If the *Agent* fails, all existing *MT-*  
369 *Connect Assets* **MAY** be lost. It is the responsibility of the implementation to restore the  
370 lost *Asset* data and it is the responsibility of the application to persist the *Asset* data. The  
371 *Agent* **MAY** make no guarantees about availability of *Asset* data after the *Agent* stops.

## 372 6 Extensions to Part 3.0 - Streams Information Model

373 The associated modifications **MUST** be added to *MTConnect Standard: Part 3.0 - Streams*  
 374 *Information Model* to add the following event to the `Events` in the streams.

### 375 6.1 AssetChanged Extension to Events

376 The `AssetChanged` element extends the base `Event` type XML data element defined in  
 377 *MTConnect Standard: Part 3.0 - Streams Information Model* and adds the `assetType`  
 378 attribute to the base `Event`. This new `Event` will signal whenever a new *MTConnect*  
 379 *Asset* is added or the existing definition of an *Asset* is updated. The `assetId` is provided  
 380 as the CDATA value and can be used to request the *Asset* data from the *Agent*.



**Figure 7:** AssetChanged Schema

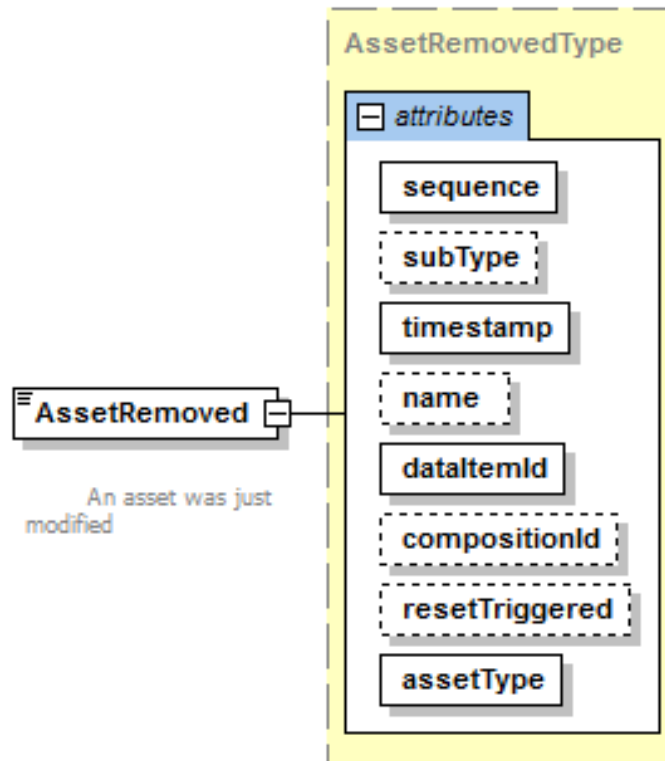
381 `AssetChanged`: An *MTConnect Asset* has been added or modified. The CDATA  
 382 for the `AssetChanged` element **MUST** be the `assetId` of the *Asset* that has been  
 383 modified.

384 **6.1.1 AssetChanged event Attributes**

**Table 7:** Attributes for AssetChanged

Attribute	Description	Occurrence
assetType	The type of asset changed. assetType is a required attribute. <i>Valid Data Values:</i> Cutting Tool	1

385 **6.2 AssetRemoved Extension to Events**



**Figure 8:** AssetRemoved Schema

386 AssetRemoved: An *MTConnect Asset* has been removed. The CDATA for the *As-*  
 387 setRemoved element **MUST** be the *assetId* of the *Asset* that has been removed.

388 **6.2.1 AssetRemoved Attributes****Table 8:** Attributes for AssetRemoved

Attribute	Description	Occurrence
assetType	The type of asset that was removed. assetType is a required attribute. <i>Valid Data Values:</i> Cutting Tool	1

389 The *MTConnect Asset* will still be available if requested if the removed=true argument is  
390 supplied. The assetId is provide as the CDATA value and can be used to request the  
391 *Asset* data from the *Agent*.

## 392 Appendices

### 393 A Bibliography

394 Engineering Industries Association. *EIA Standard - EIA-274-D*, Interchangeable Variable,  
395 Block Data Format for Positioning, Contouring, and Contouring/Positioning Numerically  
396 Controlled Machines. Washington, D.C. 1979.

397 ISO TC 184/SC4/WG3 N1089. *ISO/DIS 10303-238*: Industrial automation systems and  
398 integration Product data representation and exchange Part 238: Application Protocols: Ap-  
399 plication interpreted model for computerized numerical controllers. Geneva, Switzerland,  
400 2004.

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

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

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

410 Electronic Industries Association. *ANSI/EIA-494-B-1992*, 32 Bit Binary CL (BCL) and  
411 7 Bit ASCII CL (ACL) Exchange Input Format for Numerically Controlled Machines.  
412 Washington, D.C. 1992.

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

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

418 International Organization for Standardization. *ISO 10303-21*: 1996, Industrial automa-  
419 tion systems and integration – Product data representation and exchange – Part 21: Imple-  
420 mentation methods: Clear text encoding of the exchange structure. Geneva, Switzerland,  
421 1996.

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

423 New York, 1984.

424 International Organization for Standardization. *ISO 841-2001: Industrial automation sys-*  
425 *tems and integration - Numerical control of machines - Coordinate systems and motion*  
426 *nomenclature*. Geneva, Switzerland, 2001.

427 *ASME B5.59-2 Version 9c: Data Specification for Properties of Machine Tools for Milling*  
428 *and Turning*. 2005.

429 *ASME/ANSI B5.54: Methods for Performance Evaluation of Computer Numerically Con-*  
430 *trolled Machining Centers*. 2005.

431 OPC Foundation. *OPC Unified Architecture Specification, Part 1: Concepts Version 1.00*.  
432 July 28, 2006.

433 International Organization for Standardization. *ISO 13399: Cutting tool data representa-*  
434 *tion and exchange*. Geneva, Switzerland, 2000.