



MTConnect[®] Standard
Part 4.0 – Assets Information Model
Version 1.5.0

Prepared for: MTConnect Institute
Prepared on: December 2, 2019

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

MTConnect Specification and Materials

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

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

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

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

Table of Contents

1	Purpose of This Document	2
2	Terminology and Conventions	3
2.1	Glossary	3
2.2	Acronyms	7
2.3	MTCConnect References	8
3	MTCConnect Assets	9
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
4	MTCConnect Assets Architecture	17
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
5	Extensions to Part 2.0 - Devices Information Model	21
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
6	Extensions to Part 3.0 - Streams Information Model	23
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
	Appendices	26
A	Bibliography	26

Table of Figures

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

List of Tables

Table 1: MTConnectAssets Header	12
Table 2: MTConnect Assets Element	13
Table 3: MTConnect Asset Element	13
Table 4: Attributes for Asset	14
Table 5: Elements for Asset	16
Table 6: DataItem Type for EVENT category	21
Table 7: Attributes for AssetChanged	24
Table 8: Attributes for AssetRemoved	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 *Agent*

31 Refers to an MTConnect Agent.

32 Software that collects data published from one or more piece(s) of equipment, orga-
 33 nizes that data in a structured manner, and responds to requests for data from client
 34 software systems by providing a structured response in the form of a *Response Doc-*
 35 *ument* that is constructed using the *semantic data models* defined in the Standard.

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

37 *Asset*

38 General meaning:

39 Typically referred to as an *MTConnect Asset*.

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

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

45 See description of *buffer*.

46 Used as an *Information Model*:

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

50 The *Asset Information Models* defines the structure for the *Assets Response Docu-*
 51 *ment*.

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

55 Used when referring to an *MTCConnect Asset*:

56 Refers to the information related to an *MTCConnect Asset* or a group of *MTCConnect*
 57 *Assets*.

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

59 Used as an XML container or element:

60 • When used as an XML container that consists of one or more types of *Asset*
 61 XML elements.

62 Appears in the documents in the following form: *Assets*.

63 • When used as an abstract XML element. It is replaced in the XML document
 64 by types of *Asset* elements representing individual *Asset* entities.

65 Appears in the documents in the following form: *Asset*.

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

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

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

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

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

74 Appears in the documents in the following form: *MTCConnectAssets*.

75 Used as an *MTConnect Request*:

76 Represents a specific type of communications request between a client software ap-
77 plication and an *Agent* regarding *MTConnect Assets*.

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

79 Used as part of an *HTTP Request*:

80 Used in the path portion of an *HTTP Request Line*, by a client software applica-
81 tion, to initiate an *Asset Request* to an *Agent* to publish an `MTConnectAssets`
82 document.

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

84 ***Asset Document***

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

87 ***buffer***

88 General meaning:

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

91 Used relative to *Streaming Data*:

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

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

95 Used relative to *MTConnect Assets*:

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

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

98 ***Data Entity***

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

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

103 ***Document***

104 General meaning:

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

106 Used to represent an *MTConnect Document*:

107 Refers to printed or electronic document(s) that represent a *Part(s)* of the MTCon-
108 nect Standard.

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

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

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

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

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

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

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

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

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

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

122 ***Equipment Metadata***

123 See *Metadata*

124 ***HTTP Request***

125 In the MTCConnect Standard, a communications command issued by a client soft-
126 ware application to an *Agent* requesting information defined in the *HTTP Request*
127 *Line*.

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

129 ***HTTP Request Line***

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

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

133 ***Information Model***

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

136 For example, an information model is used to define the structure for each *MTCCon-*
137 *nect Response Document*; the definition of each piece of information within those
138 documents and the relationship between pieces of information.

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

140 ***MTCConnect Document***

141 See *Document*.

142 ***MTCConnect Request***

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

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

146 ***MTCConnect XML Document***

147 See *Document*.

148 ***Request***

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

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

152 ***Response Document***

153 See *Document*.

154 ***semantic data model***

155 A methodology for defining the structure and meaning for data in a specific logical
156 way.

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

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

160 ***Streaming Data***

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

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

164 ***Valid Data Value***

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

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

168 **2.2 Acronyms**

169 **AMT**

170 The Association for Manufacturing Technology

171 **2.3 MTConnect References**

172 [MTConnect Part 1.0] *MTConnect Standard Part 1.0 - Overview and Fundamentals*. Ver-
173 sion 1.5.0.

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

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

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

179 3 MTConnect Assets

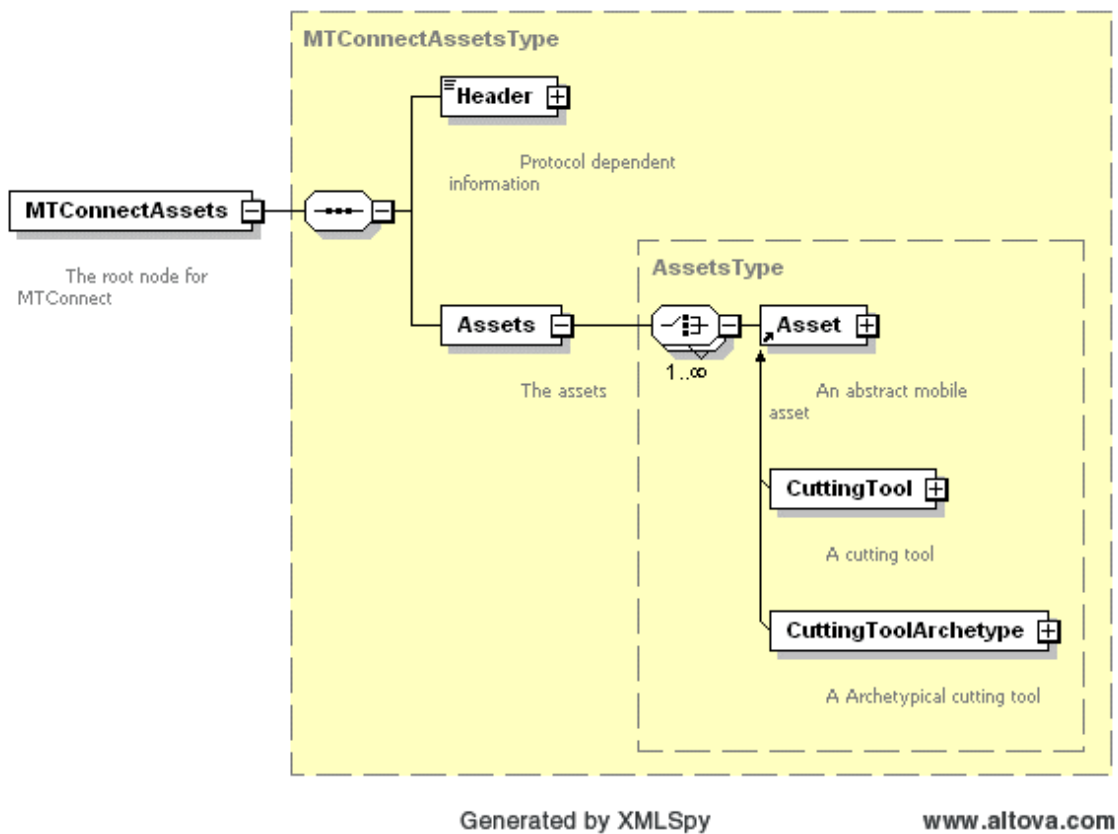
180 3.1 Overview

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

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

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

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

203 **3.2 MTConnectAssets****Figure 1: MTConnectAssets Schema**

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

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

211 **3.2.1 MTConnectAssets Header**

212 The `MTConnectAssets` header is where the protocol sequence information **MUST** be
 213 provided. The XML schema in *Figure 2* represents the structure of the `MTConnectAs-`
 214 `sets` header showing the attributes defined for `MTConnectAssets`.

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

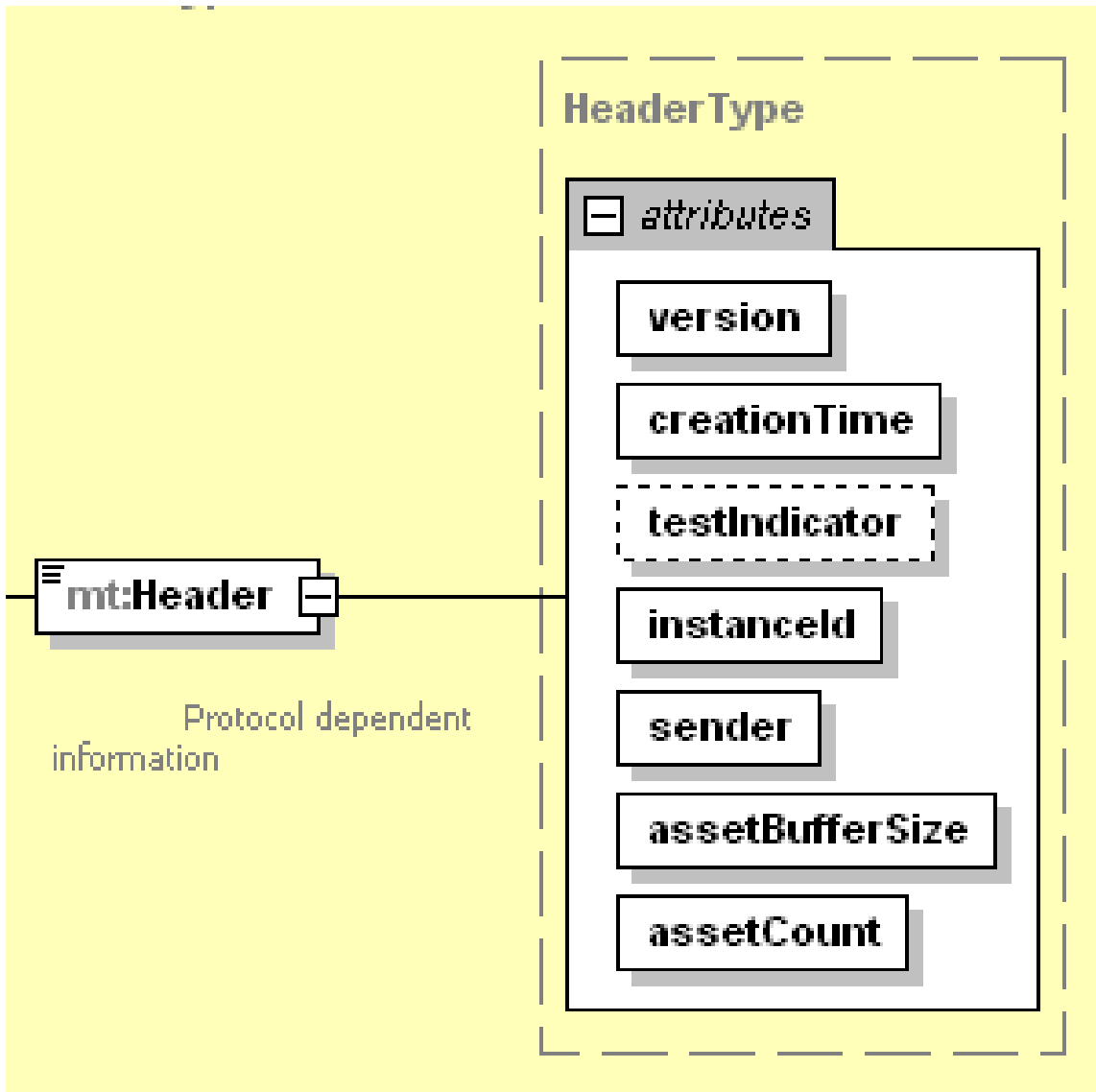


Figure 2: MTCConnectAssets Header

217 **3.2.1.1 Header Attributes**

218 *Table 1* defines the attributes used to provide information for an `MTCConnectAssets`
219 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 MUST have a maximum value of $2^{64} - 1$ and MUST 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 MUST 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 MUST be an unsigned positive integer value with a maximum value of $2^{32} - 1$. This value MUST NOT be greater than assetBufferSize. assetCount is a required attribute.	1

Example 1: MTConnectAssets Header Example

```

220 1 <Header creationTime="2010-03-13T07:59:11+00:00"
221 2     sender="localhost" instanceId="1268463594"
222 3     assetBufferSize="1024" version="1.1"
223 4     assetCount="12" />

```

224 3.2.2 Assets

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

Table 2: MTConnect Assets Element

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

227 3.2.3 Asset

228 An *Asset* XML element is a container type XML element used to organize information
 229 describing an entity that is not a piece of equipment. *Asset* is an abstract type XML
 230 element and will never appear directly in the MTConnect XML document. As an abstract
 231 type XML element, *Asset* will be replaced in the XML document by specific *MTConnect*
 232 *Asset* type.

Table 3: MTConnect Asset Element

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

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

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

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

239 **3.2.3.1 Common Asset Attributes**

240 The XML schema in *Figure 3* represents the structure of *Asset* showing the attributes
 241 defined for *Asset*.

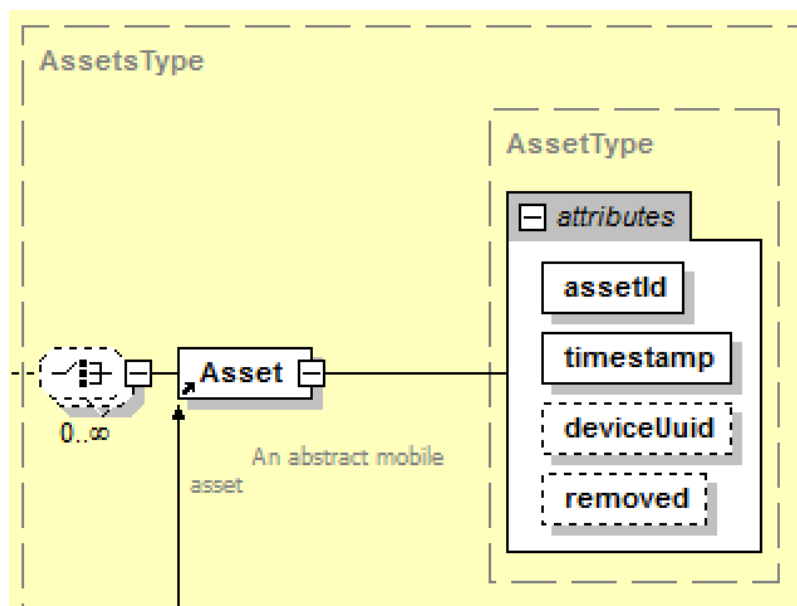


Figure 3: Asset Schema

242 *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 MUST be unique with respect to all other <i>Assets</i> in an <i>MTCConnect</i> installation. The identifier SHOULD 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> MUST be provided in UTC (Universal Time Coordinate, also known as GMT). This is the time the <i>Asset</i> data was last modified. timestamp 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 <i>Device</i> element. This can be any series of numbers and letters as defined by the XML type NMTOKEN.	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 MUST be assumed to be false. The value is an <code>xsi:boolean</code> type and MUST be <code>true</code> or <code>false</code> .	0..1

243 All *MTCConnect Assets* **MUST** have an `assetId` that differs from all the other *Assets* in
 244 a facility and preferably globally unique, such as a RFC 4122 UUID. There **MUST** never
 245 be more than one *Asset* provided by an *Agent* with the same `assetId` in the same shop.

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

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

257 Each piece of equipment or location may have a different view of this *Asset* and it is
 258 the responsibility of an application to collect and determine the aggregate information
 259 and keep a historical record if required. An *Agent* will allow any application or other
 260 equipment to request this information. The piece of equipment **MUST** supply the latest
 261 and most accurate information regarding a given *Asset*.

262 **3.2.3.2 Common Asset Elements**

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

264 The XML schema in *Figure 4* represents the structure of `Description`.

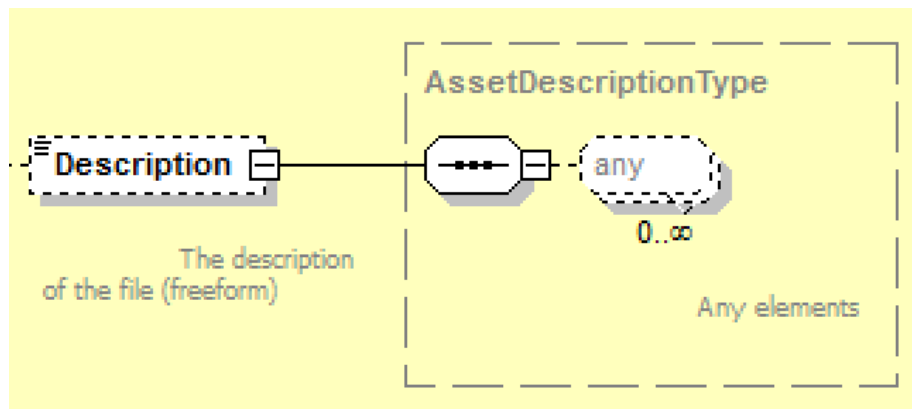


Figure 4: Description Schema

265 *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

266 4 MTConnect Assets Architecture

267 4.1 Agent Asset Storage

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

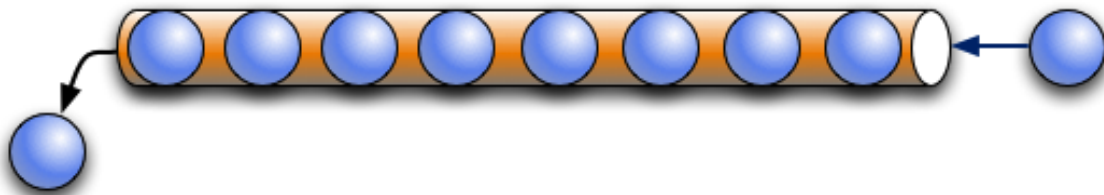


Figure 5: MTConnect Assets storage as First in First Out

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

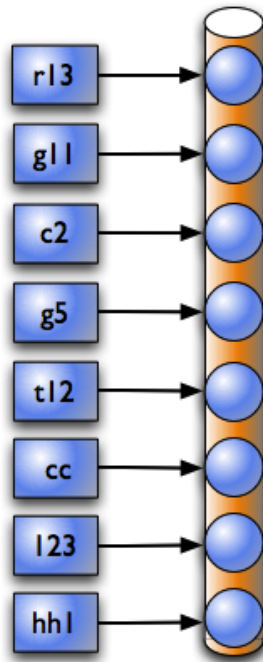


Figure 6: MTConnect Assets storage as Key/Value pairs

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

282 4.2 Asset Protocol

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

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

288 4.2.1 Asset by `assetId`

Example 2: Asset by `assetId` Example

```
289 1 url: http://example.com/asset/e39d23ba-ef2d-
290 2     11e6-b12c15028cfe91a82ef
```

291 *Example 2* returns the `MTConnectAssets` document for *Asset* e39d23ba-ef2d-
292 11e6-b12c-28cfe91a82ef

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

Example 3: Assets by assetId Example

294 1 url: http://example.com/asset/e39d23ba-ef2d-11e6-b12c155;
295 2 8cfe91a82ef;e46d5256-ef2d-11e6-96aa-28cfe91a82ef

296 *Example 3* returns the `MTConnectAssets` document for *Assets* e39d23ba-ef2d-
297 11e6-b12c-28cfe91a82ef and e46d5256-ef2d-11e6-96aa-28cfe91a82ef.

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

Example 4: Get all Assets Example

299 1 url: http://example.com/assets

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

303 4.2.2 Asset for a Given Type

Example 5: Asset for a Given Type Example

304 1 url: http://example.com/assets?type="CuttingTool"

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

308 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

309 1 url: http://example.com/assets?type="CuttingTool"

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

312 4.2.3 Assets Including Removed Assets

Example 7: Assets Including Removed Assets Example

313 1 url: http://example.com/assets?type=CuttingTool&removed=true

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

316 4.2.4 Assets for a Piece of Equipment

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

Example 8: Assets For a Piece of Equipment Example

```
319 1 url: http://example.com/Mill123/assets
```

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

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

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

```
325 1 url: http://example.com/Mill123/asset/  

  326 2           ?type=CuttingTool&count=100
```

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

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

```
329 1 url: http://example.com/Mill123/asset/  

  330 2           ?type=CuttingToolArchetype
```

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

333 5 Extensions to Part 2.0 - Devices Information Model

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

338 5.1 Data Item Types added for EVENT Category

Table 6: DataItem Type for EVENT category

DataItem Type SubType	Description
ASSET_CHANGED	The value of the CDATA for the event MUST be the <code>assetId</code> of the asset that has been added or changed. There will not be a separate message for new assets.
ASSET_REMOVED	The value of the CDATA for the event MUST 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.

339 5.1.1 ASSET_CHANGED Data Item Type

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

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

349 5.1.2 ASSET_REMOVED Data Item Type

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

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

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

364 6 Extensions to Part 3.0 - Streams Information Model

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

367 6.1 AssetChanged Extension to Events

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

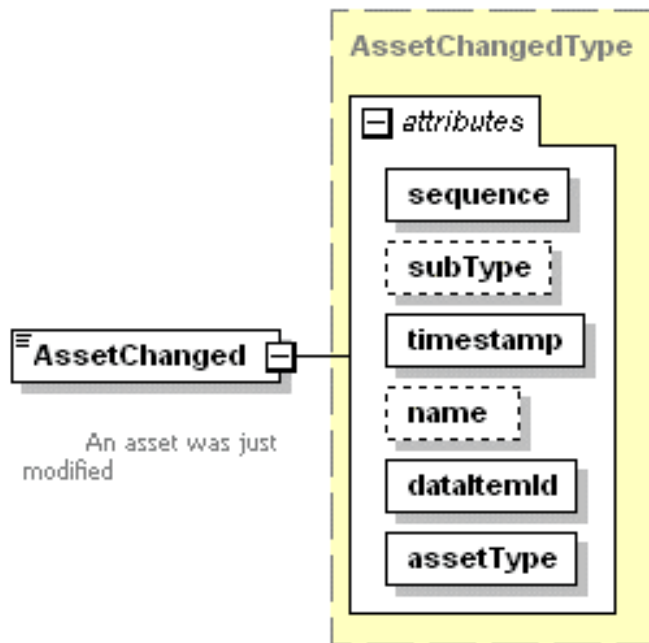


Figure 7: AssetChanged Schema

373 `AssetChanged`: An *MTConnect Asset* has been added or modified. The CDATA
 374 for the `AssetChanged` element **MUST** be the `assetId` of the *Asset* that has been
 375 modified.

376 **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

377 **6.2 AssetRemoved Extension to Events**

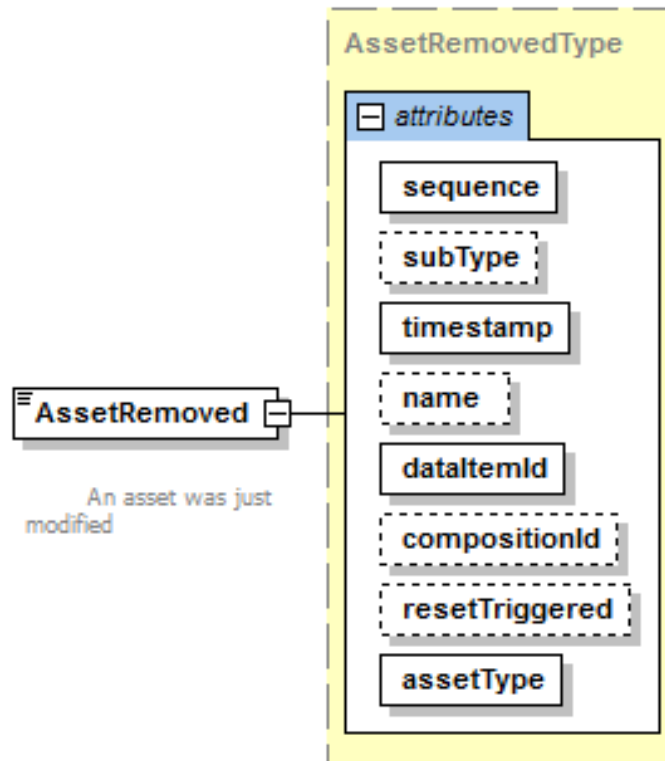


Figure 8: AssetRemoved Schema

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

380 **6.2.1 AssetRemoved Attributes****Table 8:** Attributes for AssetRemoved

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

381 The *MtConnect Asset* will still be available if requested if the removed=true argument is
382 supplied. The assetId is provide as the CDATA value and can be used to request the
383 *Asset* data from the *Agent*.

384 Appendices

385 A Bibliography

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

389 ISO TC 184/SC4/WG3 N1089. *ISO/DIS 10303-238*: Industrial automation systems and
390 integration Product data representation and exchange Part 238: Application Protocols: Ap-
391 plication interpreted model for computerized numerical controllers. Geneva, Switzerland,
392 2004.

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

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

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

402 Electronic Industries Association. *ANSI/EIA-494-B-1992*, 32 Bit Binary CL (BCL) and
403 7 Bit ASCII CL (ACL) Exchange Input Format for Numerically Controlled Machines.
404 Washington, D.C. 1992.

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

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

410 International Organization for Standardization. *ISO 10303-21*: 1996, Industrial automa-
411 tion systems and integration – Product data representation and exchange – Part 21: Imple-
412 mentation methods: Clear text encoding of the exchange structure. Geneva, Switzerland,
413 1996.

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

415 New York, 1984.

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

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

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

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

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