Electronic Components Technical Dictionary (ECTD) Specification

Version 2.0-050100

Introduction

This specification defines the conceptual model, an XML DTD implementation, and related constraints and semantics for the Electronic Components Technical Dictionary (ECTD).

The ECTD is designed to support unambiguous and automated electronic exchange of product information by standardizing semantics and making them machine-sensible. Information consists of both product characteristics and product information objects of arbitrary format, encoding, and structure.

Users of product information need to automate location and analysis of data about available products that fit specific design criteria from potential suppliers worldwide. Yet, suppliers have different internal representation and external presentation schemas and formats, and even different semantics for this data. Often access to and understanding of the data requires extensive human intervention. Efficient, machine-sensible assessment of candidate products, and computer use of information for selected products, requires unambiguous representation of characteristics of those candidates.

The ECTD instance is comprised of entries that define semantics for different kinds of product information:

- Structured: characteristics and conditions;
- Non-structured: terms, classes, and sets;
- Administrative: object content types, target software, file formats, value types, and reasons for null values.

The ECTD schema and constraints provide support for:

- Consistent interpretation of semantics across all uses for:
  - product characteristics values.
  - dependent conditions affecting values of product characteristics.
  - characters, words, or phrases with specific significance.
  - named sets of characteristics, which may be nested to any level.
  - classification of groups of products with common characteristics.
- Graphics, formulas, descriptions, references, data types, value encoding formats.
- Resolution to standard naming conventions of customized names and symbols for product characteristics and conditions.
- Extensibility to allow addition of domain-specific semantics to entries.

- Revision control so that:
  - A dictionary can be edited without invalidating existing references to it (the number and location of which can never be known).
  - Changes guaranteed to be backward compatibile (minor) can be differentiated from ones that are not (major).

As a result, ECTD instance data, as well as messages and documents that reference entries in an ECTD instance, can be processed automatically for such tasks as:

- Component search and comparison.
- Recognition of domain specific message semantics.
- Extraction of characteristics, conditions, terms, and other product data necessary for:
  - component information system database import/export.
  - Automatic generation of CAD libraries.
  - use by EDA tools.

Requirements

The detailed requirements and problem statements driving this specification are documented in the “ECIX II Requirements”.

Compliance

An implementation that fails to satisfy any constraint identified herein, or in the relevant normative references, with the keywords “required", "shall", or "shall not", is not compliant to the specification. These keywords, along with the keywords "may", "should", "should not", and "optional", are to be interpreted as described in RFC 2119.

Notation Conventions

This specification uses the following conventions:

1. Characters in `courier` are literal values to be used in code or messages exactly as shown.

2. Characters in times are not to be used literally but instead describe how values shall be constructed when used in code or messages.

3. Information contained in the following notation shall be considered non-normative:

   (NOTE: This text is informative only and is not considered part of this specification.)

   (EXAMPLE: This text is informative only and is not considered part of this specification.)

4. Characters bounded by double quotes have a special significance referenced as a whole in the sentence, such

as a title, value, name, or other identifier.

5. An identifier bounded by angular brackets refers to a DTD element object. For example, <value> refers to the particular DTD element named “value”. If the letter “s” immediately follows the trailing right angle bracket it indicates one or more such elements.

6. An identifier bounded on the left by the percent character and on the right by the semicolon character refers to a DTD entity object. For example, %xml.atts; refers to the DTD entity named “xml.atts”.

7. An identifier bounded by single quotes refers to a DTD attribute object. For example, ‘NoValueReason’ refers to the DTD attribute named “NoValueReason”.

References

Normative

Documents identified as "Normative References" are referenced in part or in their entirety in the current document and the parts referenced shall have normative impact on compliant implementations. For each reference the edition indicated is the authority. All normative documents are subject to revision at different times and by different agencies. Consequently, application of more recent editions of such documents may affect interoperability of implementations and may have additional implications.

Informative

Documents identified as "Informative References" are relevant to the current document, and may assist in comprehension of it, but do not have normative impact on compliant implementations.

Terminology

The following term definitions are used in this specification. Additional terms may be defined inline, as needed, throughout the specification.

Characteristic	A feature, parameter, quality, aspect, or property of a product that is defined by a CharacteristicDefinition dictionary entry. A Characteristic occurs in an XML instance that references the dictionary, not in the dictionary itself.
CharacteristicDefinition	The dictionary entry that defines the semantics of a Characteristic.
CharacteristicSet	An ordered collection of one or more Characteristics and/or other CharacteristicSets. A CharacteristicSet occurs in an XML instance that references the dictionary, not in the dictionary itself.
current	The immediate or an indirect parent element (depending on the next word after “current”) of the element being described.
data dictionary	A valid XML instance that conforms to the ECTD DTD.
Condition	A dependent condition that is defined by a ConditionDefinition dictionary entry. A

	Condition occurs in an XML instance that references the dictionary, not in the dictionary itself.
ConditionDefinition	The dictionary entry that defines the semantics of a Condition.
dependent condition	An operational, environmental, or state-related factor or variable on which the measurement or interpretation of the value of a characteristic depends.
dictionary	A valid XML instance that conforms to the ECTD DTD.
ECTD dictionary	The standard dictionary.
entry	Any of the elements that are immediate children of the top-level <ec.dictionary> element.
external instance	An XML instance of a DTD other than the ECTD DTD.
instance	An XML instance of the ECTD DTD.
data.object	Product information that may have any arbitrary data format, encoding, and internal structure.
Property	An abstract supertype that includes both Conditions and Characteristics. A Property occurs in an XML instance that references the dictionary, not in the dictionary itself.
PropertyDefinition	An abstract supertype that includes both ConditionDefinitions, CharacteristicDefinitions, and PropertyDefinitionSets.
standard dictionary	A valid XML instance that conforms to the ECTD DTD and is the common set of entries referenced by messages exchanging electronic component data.
term	One or more characters, words, or phrases that have specific semantics relevant to the context in which they appear.

Conceptual Model

The UML class diagrams in Figures 1, 2, and 3, define the conceptual model for the EC Technical Dictionary.

Figure 1: ECTD Conceptual Model (Part 1 of 3)

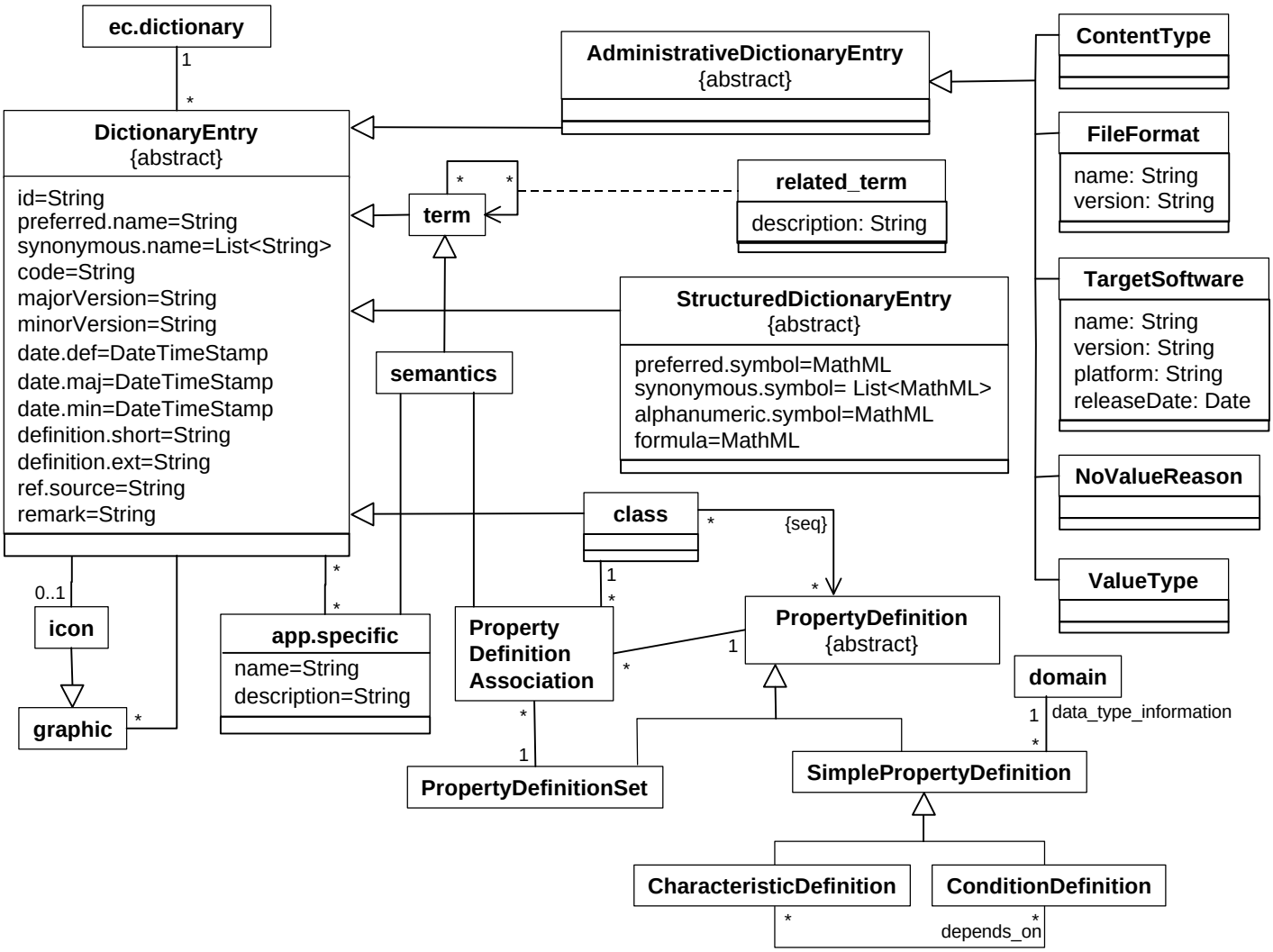


Figure 2: ECTD Conceptual Model (Part 2 of 3) – domain

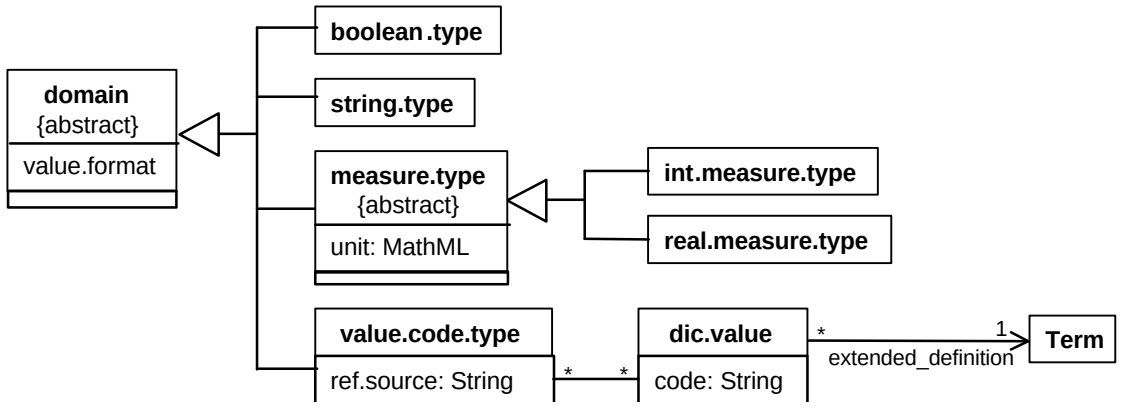
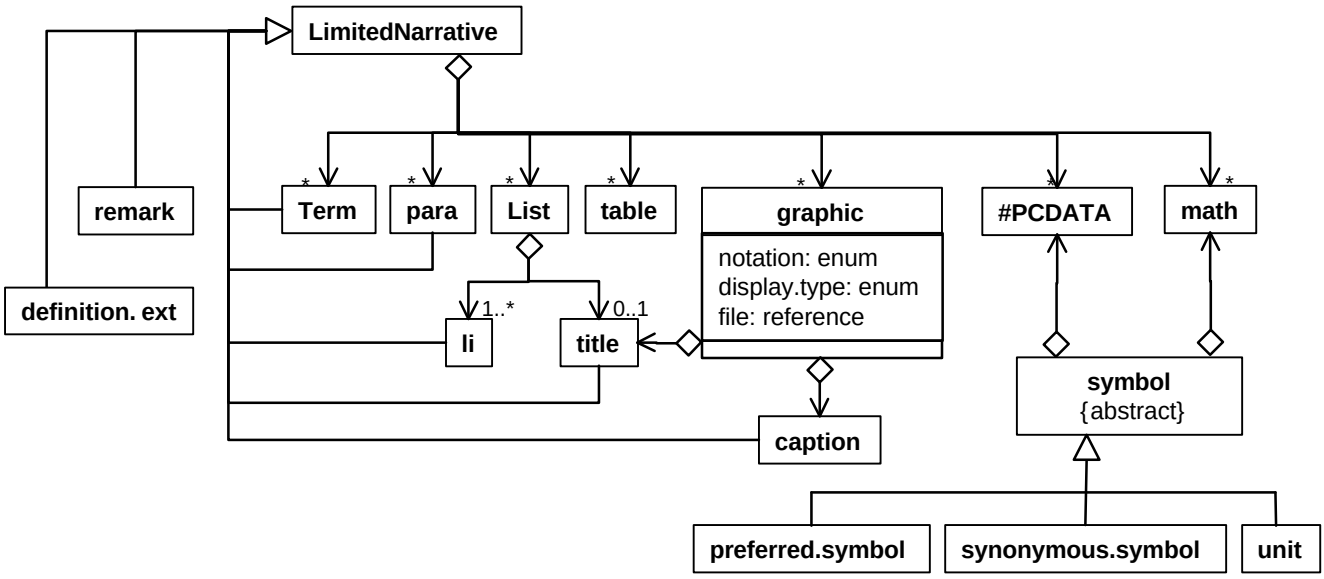


Figure 3: ECTD Conceptual Model (Part 3 of 3) – LimitedNarrative, symbol



Instance Constraints and Semantics

DTD

The ECTD DTD enforces instance constraints, such as container relationships, cardinality limitations, some presence/absence requirements, and a minimal level of data typing.

Version

An instance shall conform to the Extensible Markup Language (XML) 1.0 specification and begin with the following XML declaration:

```
<?xml version="1.0"?>
```

DTD Objects

Additional constraints and semantics on the use of DTD objects are defined in the tables that follow to ensure machine sensibility of dictionary content, interoperability of ECTD-based software, and support for business and technical requirements:

- Table 1: Element Constraints and Semantics
- Table 2: Attribute Constraints and Semantics
- Table 3: External Entity Constraints and Semantics
- Table 4: Parameter Entity Constraints and Semantics

Table 1: Element Constraints and Semantics

Element Name	Semantics and Constraints
alphanumeric.symbol	An abbreviated, plain text representation of a <preferred.symbol>, using Latin letters only, constructed according to the following rules: **1.** The first character shall be a letter, "@", or a "$" (when defining a specific Greek letter). If applicable, the "@" sign precedes the "$" sign. **2.** The length shall be limited to 15 characters. **3.** Greek letters that are graphically identical to Latin letters are not converted; **4.** Other Greek letters shall be represented by the corresponding ISO/R843 Latin letter preceded by a dollar " $ " character; **5.** An asterisk "*" character shall be used to indicate that the following letter is a superscript. **6.** An underline "_" character shall be used to indicate that the following letter is a subscript. **7.** There shall be no <alphanumeric.symbol> if there is no <preferred.symbol>.
app.specific	A general-purpose container for adding arbitrary information to any entry. **1.** The content shall be the value associated with the tag's "name" attribute. **2.** The semantics of the content, its relationship to the attribute, and any relationship to the containing dictionary entry, shall be defined by an external specification that describes a particular domain of application of the ECTD. **3.** If an entry includes an <app.specific> tag whose 'name' attribute has a value of "deprecated" then references should not be made to it.
boolean.type	A constraint limiting to either 0 or 1 the value[s] of a Property instance that is defined by the current PropertyDefinition. **1.** The <value.format> of the current PropertyDefinition shall be "B 1".
caption	Descriptive information.
class	A category that describes a set of related products, associating CharacteristicDefinitions with them.
code	An identifier for the containing entry. **1.** The content combined with the content of the <majRev> shall form a unique key in the namespace of the containing <ec.dictionary>.
CharacteristicDefinition	The definition of the semantics and related information for a value reported in an external instance for a Characteristic of an electronic component. **1.** The content shall not be empty within the context of an <ec.dictionary> instance.
ConditionDefinition	The definition of the semantics and related information for a value reported in an external instance for a Condition of a Characteristic. **1.** The content shall not be empty within the context of an <ec.dictionary> instance.
ContentType	A classification identifier that indicates a category of data object.
definition.short	A brief version of the normative definition of the entry. **1.** The content shall be one sentence free of any markup. **2.** The semantics of the content shall not contradict those of the definition.ext.

definition.ext	An extended version of the normative definition of the entry. **1.** The content may include additional markup. **2.** The semantics of the content shall not contradict those of the definition.short.
depends.on	The set of ConditionDefinitions, if any, that define the Conditions on which the interpretation or measurement of the value of a Property (defined by the current PropertyDefinition) depends.
dic.value	One of the enumerated values in a value code list. **1.** The content shall be one of the legal values for an element defined by the containing dictionary entry. **2.** Additional information that describes the <dic.value> shall be inserted into a <term> and the value of the 'ref.to.term' attribute shall match the 'id' attribute of that <term>. **3.** If no additional information is available for the <dic.value>, the 'ref.to.term' attribute shall be omitted.
domain	The definition of data type and related information for the current PropertyDefinition.
ec.dictionary	The top level container for an instance of the ECTD DTD.
FileFormat	Information identifying the type of encoding of information in a data object.
formula	A normative, mathematical expression related to the current entry.
graphic	A reference to, plus related information about, a non-text object.
icon	A normative, non-text identifier for an entry.
identifiers	Identification and version information for the immediate parent container.
li	One member element of a parent <List>.
List	A set of elements sharing some common semantics. Such commonality may be described by the child <title> element, by text outside the element, or may be implicit.
int.measure.type	A constraint limiting to integer[s] the value[s] of a Property instance that is defined by the current PropertyDefinition.
majRev	The minor revision code for the containing entry. **1.** The <majRev> of an entry shall be incremented by 1 and the <minRev> shall be set to 0 in the case where changes are made to any part of the containing entry that cannot be guaranteed to have no impact on conformance of any reference to that entry. **2.** The majRev of the ECTD shall be increased by 1 if any of the following occur: • Any <majRev> is increased. • Any entry is added. • A <majRev> change is made to the ECTD DTD that defines the transmission format.
minRev	The minor revision code for the containing entry. **1.** The <minRev> of an entry shall be incremented by 1 and the <majRev> shall not be changed in the case where changes are made to any part of the containing entry, and the total of those changes can be guaranteed to have no impact on conformance of any reference to that entry. **2.** If any entry's <minRev> is increased then the <minRev> of the ECTD shall be

	increased by 1.
math	A mathematical expression defined by the %MathModel; entity. **1.** "Presentation" MathML shall be used when the <math> element occurs: • within a content model defined by the %type.LimitedNarrative; entity. • within a <formula> element. **2.** "Content" MathML (and optional "presentation" MathML) shall be used when the <math> element occurs anywhere else.
names	Normative, symbolic identifiers for the current entry.
NoValueReason	A qualifier that defines special semantics for null values.
para	An element used for arbitrary grouping of text elements.
preferred.name	The primary text identifier for the current entry.
preferred.symbol	The primary symbolic representation of the current entry.
PropertyDefinitionSet	The definition of the semantics and related information of an ordered set of: **1.** PropertyDefinitions and/or PropertyDefinitionSets **2.** ConditionDefinitions and/or PropertyDefinitionSets **3.** A PropertyDefinitionSet that contains directly or indirectly a CharacteristicDefinition may not contain directly or indirectly any ConditionDefinition.
real.measure.type	A constraint limiting to real number[s] the value[s] of a Property instance that is defined by the current PropertyDefinition.
ref.source	A non-normative reference to the document from which a significant portion of the semantics of the parent element was derived. (NOTE: No standard encoding has been defined to make the content machine-sensible.)
related.term	A <term> with some semantic association to the current <term>. (NOTE: The content may define the type of semantic association, though not necessarily in a machine-sensible way.)
remark	Non-normative information related to the entry.
string.type	A constraint limiting to a string[s] the value[s] of a Property instance that is defined by the current PropertyDefinition.
synonymous.name	Alternate text identifier for the current entry.
synonymous.symbol	An alternate symbolic representation of the current entry.
table	Text information encoded in an ordered set of machine-sensible rows and columns. **1.** The table model shall be defined by the %TableModel; entity.
TargetSoftware	Information that identifies the software necessary to process a data object.
Term	An entry that defines the semantics of some entity that does not have a value.
TermDefinition	One or more of the components of the %type.LimitedNarrative; entity that is identified as defined by a <term>.
title	Descriptive information.
unit	The unit of measure for a value of a Characteristic defined by a CharacteristicDefinition with a quantitative domain type. The content shall be one of the following: **1.** 1, in the case there is no unit. **2.** A 3-letter code conforming to ISO 4217, in the case the value is a currency

	measure. **3.** A value selected from the list defined in Annex B of IEC 61360-1. **4.** A unit composed of base, derived, or commonly used "outside" units of the International System of Units (SI, a.k.a. the modern metric system), in the case no entry in IEC 61360-1 is appropriate. Prefixes or magnitude qualifiers are disallowed.
value.code.type	A constraint limiting the values of a Property instance that is defined by the current PropertyDefinition to the finite set of enumerated identifiers defined by the <value.domain> of the current PropertyDefinition.
value.domain	The set of enumerated value codes that are legal for values of a Characteristic or Condition defined by the current CharacteristicDefinition or ConditionDefinition.
value.format	A constraint on the characters allowed for representation of a value.
ValueType	A qualifier that defines special semantics for a value.

Table 2: Attribute Constraints and Semantics

Attribute Name	**Semantics and Constraints**
conditions	The definition (by reference) of a set of dependent conditions relevant for values of characteristics defined by the current CharacteristicDefinition.
display.type	The indication that the rendering of the current <graphic> is intended to be one of: • **inline,** with text on one or both sides of the same baseline. • **display**, with text above and/or below, only.
ent	The reference to the entity definition in the instance that can be resolved to a file that is the data for the current <graphic> element.
id	The unique identifier for an element.
list.type	The indication that the elements of the current list are one of: • **ordered**, meaning that the sequence is significant. • **unordered**, meaning that the sequence is insignificant.
name	An arbitrary identifier associated with the content of the current <app.specific>, and its parent element, with semantics that shall be defined by an external specification that describes a particular domain of application of the ECTD.
propDefs	The set of members of the current <class> element or <PropertyDefinitionSet> element. **1.** The normative child elements of the current element shall define membership semantics. **When the context is:** **2.** **<class>:** The definition (by reference) of the set of PropertyDefinitions relevant to the containing class. • Each of the IDREF values shall match the 'id' attribute of a CharacteristicDefinition or PropertyDefinitionSet element. **3.** **<PropertyDefinitionSet>:** The definition (by reference) of the set of member PropertyDefinitions that comprise the containing PropertyDefinitionSet. • Each of the IDREF values shall match the 'id' attribute of a

	CharacteristicDefinition or PropertyDefinitionSet element. • No IDREF value shall match the 'id' attribute of any PropertyDefinitionSet that contains a 'props' attribute with a reference to the current PropertyDefinitionSet, either directly or indirectly (ie, recursion is disallowed).
ref.TermDef	When the context is: **1. <related.term>:** The reference to the <term> that is related to the current <term>. **2. <ref.term>:** The reference to the <term> that defines the content of the current element. **3. <dic.value>:** The reference to the <term> that defines the content of the current element.
semantics	Additional, standardized semantics for members of the current <PropertyDefinitionSet> or <class>. **1.** If present, there shall be a one to one correspondence between the values of the 'semantics' attribute and the values of the 'propDefs' attribute in the current element. **2.** For a given 'propDefs' value the corresponding 'semantics' value shall match exactly one of: • The value of the 'id' attribute of a <TermDefinition> element that shall define additional semantics for the PropertyDefinition (referenced by the 'propDefs' value) within the context of the current <PropertyDefinitionSet> or <class>. • The value of the 'id' attribute of the current <PropertyDefinitionSet> or <class>, in which case there are no additional semantics for the PropertyDefinition (referenced by the 'propDefs' value).

Table 3: External Entity Constraints and Semantics

Element Name	Semantics and Constraints
bodyatt	Attribute definitions added to those already defined for the table element defined in the %TableModel; entity.
paracon	An override for the definition of the content model for table entries.
MathModel	The Mathematical Markup Language (MathML) Version 2.0 DTD which defines the elements used for mathematics content.
TableModel	The DTD that defines the elements that comprise a table (a structure of information presented in rows and columns) — an XML version of the OASIS Exchange Table Model, which is a subset of the CALS Table Model (MIL-M-28001B).

Table 4: Parameter Entity Constraints and Semantics

Element Name	Semantics and Constraints
common.elements	A content model fragment that includes components common to several elements.
type.LimitedNarrative	The content model for elements that represent descriptive information with

	optional internal markup structure .
ref.TermDef	The attribute definition of a reference to a TermDefinition. **1.** The value shall match the value of the ‘id’ attribute on a <TermDefinition> element.
type.date	The content model for elements that represent dates. **1.** This content shall be a 10-character date in ISO 8601 format, e.g., "1994-03-21".
type.depends	The attribute definition for references to <ConditionDefinition>s. **1.** Each value shall match the value of an 'id' attribute of a <ConditionDefinition>.
type.String	The content model for elements that represent descriptive information in plain text, free of any markup.
type.symbol	The content model for elements that represent symbols. Any content model identified with this entity shall be treated as one single string and be either of the following: • Alphanumeric characters, which may be used only if the symbol can be represented with only Latin letters and/or digits. • MathML, which may be used for any symbol.

Message Constraints

The ECTD may be referenced by elements in different kinds of transactions. A given transaction protocol may impose its own constraints on the use of ECTD information. (EXAMPLE: The “QuickData Message Specification” defines particular ECTD use semantics.)

Application Domain Constraints

Different applications of a given transaction protocol may impose additional, domain-specific constraints on the use of ECTD information. Refer to the appropriate application domain specification for details. (EXAMPLE: The “QuickData Application: Electronic Components Domain Specification” defines particular ECTD use semantics.).

Product Information Object Constraints

Particular sets of product information relevant to one or more application domains may be defined (in terms of semantics, structure, format, data encoding, and packaging) by a <term> entry. Such product information object (PIO) definitions shall be identified by inclusion of the following tag:

```
<app.specific name="object"/>
```

Such a <term> entry may, in addition, reference an external, normative specification for the PIO, which contains additional semantics and constraints on its use.

Dictionary Content, Maintenance

The process and procedures for maintenance of and access to an ECTD instance is defined in “The ECTD Maintenance Specification”.

Normative References

Requirements

ECIX II Requirements: requirements.html.

ECTD Requirements: http://www.si2.org/mallis/rn/use.cases.html

Other Specifications

Extensible Markup Language (XML) 1.0 W3C Recommendation 10-February-1998: http://www.w3.org/TR/REC-xml.

ISO 4217:1995, “Codes for the representation of currencies and funds”, http://www.iso.ch/cate/d23132.html.

International System of Units: http://physics.nist.gov/cuu/Units/introduction.html.

Value format:

- IEC 61360-1, Section 3.4.1
- value.format.html

RFC 2119, “Key words for use in RFCs to Indicate Requirement Levels”: http://andrew2.andrew.cmu.edu/rfc/rfc2119.html.

ECTD Maintenance Specification.

OMG Unified Modeling Language Specification, Version 1.3, June 1999: http://www.omg.org/cgi-bin/doc?ad/99-06-08.

DTDs

EC Technical Dictionary DTD: DTD\ECTD\ec-new.dtd.

XML Exchange Table Model: DTD\ECTD\soextblx.ent.

Mathematical Markup Language (MathML) W3C Recommendation 7 April 1998: DTD\ECTD\mmlents.

DTD Semantics Documentation

Exchange Table Model http://www.oasis-open.org/html/a503.htm.

Mathematical Markup Language (MathML) Version 2.0: http://www.w3.org/TR/2000/WD-MathML2-20000328.

Informative References

QuickData Specifications

QuickData Message Specification: qd_message.doc.

QuickData Application: Electronic Components Domain Specification: qd_application.doc.

Other Specifications

Currency code charts:

- http://www.triacom.com/archive/iso4217.en.html
- http://www.xe.net/gen/iso4217.htm,
- http://www.cybercash.com/cybercash/merchants/support/full_iso_4217.html
- http://www.jhall.demon.co.uk/currency/by_abbrev.html
- http://www.payline.com/paylinev1/programref/iso-4217.html

OMG Unified Modeling Language Specification, Version 1.3, June 1999: http://www.omg.org/cgi-bin/doc?ad/99-06-08