SEMI E121-1017 (Reapproved 0423)
GUIDE FOR STYLE AND USAGE OF XML FOR SEMICONDUCTOR MANUFACTURING APPLICATIONS

This Standard was technically approved by the Information & Control Global Technical Committee. This edition was approved for publication by the global Audits and Reviews Subcommittee on December 21, 2022. Available at www.semiviews.org and www.semi.org in April 2023; originally published March 2003; previously published October 2017.

1 Purpose

1.1 The inclusion of eXtensible Markup Language (XML) notation within current SEMI Standards reflects the industry's growing interest in XML technology. This Guide is the first step to prepare for XML protocols in SEMI Standards and to establish consistency in XML style and usage. Recommendations given in this Document are intended for the definition and conformance of XML based standards. This Document also attempts to highlight those areas where standardization and further work may be required.

2 Scope

2.1 This Guide is intended to promote interoperable application of XML in a variety of semiconductor manufacturing usage contexts. It addresses an approach for the usage and definition of XML specifications within the semiconductor industry with the focus on the use of XML in future SEMI Standards. These guidelines should be applied to any XML document or schema generated as part of a SEMI Standard for communication between software entities. This Guide includes the following:

2.1.1 Overview of XML Resources, Organizations, and Standards,

2.1.2 XML Three Tier Model Architecture, and

2.1.3 General and specific recommendations of XML style and usage.

NOTICE: SEMI Standards and Safety Guidelines do not purport to address all safety issues associated with their use. It is the responsibility of the users of the Documents to establish appropriate safety and health practices, and determine the applicability of regulatory or other limitations prior to use.

3 Limitations

3.1 Because of the rapid development of new standards in the XML arena at the time of the writing of this Document, it was found that there are areas in XML where standards are still in the development stage. Some examples of emerging XML technologies are: XMI (XML Metadata Interchange), SOAP (Simple Object Access Protocol), and Web Services Routing. Therefore, this Guide does not include recommendations in those areas until the governing bodies approve these specifications. A list of some of the areas where guidance may be necessary is provided below:

3.1.1 (UML) (Unified Modeling Language) models with derived schema definition.

3.1.2 XMI (XML Metadata Interchange) usage and recommendations.

3.1.3 XML Namespace definitions.

3.1.4 XSLT (eXtensible Stylesheet Language Transformations) usage and conversion rules.

3.1.5 WSDL (Web Service Definition Language) usage and recommendations.

3.1.6 SOAP (Simple Object Access Protocol) and transport recommendations.

4 Referenced Standards and Documents

4.1 *W3C Documents*

eXtensible Markup Language (XML) 1.0 (Second Edition) — W3C Recommendation, 6 October 2000

XML Schema Part 1: Structures — W3C Recommendation, 2 May 2001

XML Schema Part 2: Datatypes — W3C Recommendation, 2 May 2001

Documents Document Sharing Hub

Full document details: SEMI E121-1017-0423R

NOTICE: Unless otherwise indicated, all documents cited shall be the latest published versions.

5 Terminology

5.1 *Abbreviations and Acronyms*

5.1.1 *DTD* — document type definition

5.1.2 *HTTP* — hypertext transfer protocol

5.1.3 *SGML* — standard generalized markup language

5.1.4 *SOAP* — simple object access protocol

5.1.5 *W3C* — World Wide Web Consortium

5.1.6 *WSDL* — web services definition language

5.1.7 *WWW* — World Wide Web

5.1.8 *XMI* — XML metadata interchange

5.1.9 *XML* — eXtensible markup language

5.1.10 *XPATH* — XML path language

5.1.11 *XSL* — eXtensible stylesheet language

5.1.12 *XSLT* — eXtensible stylesheet language transformations

6 Overview of XML Resources, Organizations, and Standards

6.1 There are a large number of resources in the area of XML. Most of the information is readily available from organizations such as OMG, W3C, RosettaNet, ebXML, OASIS, Sun Micro Systems, IBM and other through their websites. There has been an extensive amount work in standardizing dictionaries and vocabularies for several industries. These documents can be used until further standards and guides specific to this industry are created.

6.2 *Resources*

6.2.1 *Extensible Markup Language (XML) 1.0 (Second Edition)* — W3C, 6 October 2000

6.2.2 *Namespaces in XML* — W3C, 14 January 1999[1]

6.2.3 *XML Schema Part 0: Primer* — W3C, 2 May 2001

6.2.4 *XML Schema Part 1: Structures* — W3C, 2 May 2001

6.2.5 *XML Schema Part 2: Datatypes* — W3C, 2 May 2001

6.2.6 *XML Path Language (Xpath)* — W3C, 16 November 1999

6.2.7 RFC 2396 *Uniform Resource Identifiers (URI): Generic Syntax*[2]

6.2.8 *Hyper Text Transfer Protocol 1.1*[3]

6.2.9 *Simple Object Access Protocol 1.1* — W3C Note, 8 May 2000[4]

6.2.10 *Web Service Definition Language 1.1* — W3C Note, 15 March 2001[5]

6.2.11 *ISO/IEC 11179 (1-6) Information Technology* — Standardization and Specification of Data Elements

[1] http://www.w3.org/TR/1999/REC-xml-names-19990114/
[2] Berners-Lee, T., Fielding, R., Masinter, L., August 1998.
[3] HTTP, http://www.w3.org/Protocols/rfc2616/rfc2616.html
[4] SOAP, http://www.w3.org/TR/SOAP
[5] WSDL, http://www.w3.org/TR/wsdl

Documents
Document Sharing Hub
Full document details: SEMI E121-1017-0423R

7 XML Three Tier Model Architecture

7.1 The following three-tier schema model is recommended:

- Primary Schemas (common SEMI models)
- Secondary Schemas (individual SEMI models)
- Tertiary Schemas (non-SEMI models)

7.1.1 *Primary Schemas* — Primary Schemas are schemas that have been defined by SEMI. These schemas usually cover a particular area such as equipment, transport system, scheduler/dispatcher, maintenance, etc.

7.1.2 *Secondary Schemas* — Secondary Schemas are schemas that have been defined by SEMI. These schemas are for a particular functionality required by the application such as carrier management, process job, data collection, etc. Secondary Schemas may derive from Primary Schemas as a base by extension or restriction.

7.1.3 *Tertiary Schemas* — Tertiary Schemas are defined by the supplier of the tool or application. These schemas may derive from either Primary or Secondary Schemas as a base by extension or restriction.

8 XML Recommendations

8.1 *XML Document Compliance*

8.1.1 All XML documents, by definition, comply with the W3C Recommendation, XML version 1.0 (refer to http://www.w3.org/TR/REC-xml for more detail on the specification). The purpose of this compliance is to assure that all XML documents are 'well-formed' and 'valid'. A 'well-formed' document means that there is exactly one root element, all sub-elements (and recursive sub-elements) have delimiting start- and end-tags, and that they are properly nested within each other. A 'well-formed' document also conforms to the XML syntax specification and does not include external references unless a schema reference is provided. An XML document is 'valid' if it has an associated schema and complies with the constraints expressed in the schema. 'Valid' XML data is also 'well-formed'.

8.2 *Use of XML Schema Versus DTD*

8.2.1 All XML documents should conform to a specified set of production rules. These production rules should be specified as XML Schemas. XML Schemas are the preferable way to define XML documents. XML Schemas are modular in structure and can be built from multiple components (files) by including or importing the components.

8.3 *Using Elements Versus Attributes*

8.3.1 When defining attributes from an object model, it is preferable to define them as XML elements such that they can be easily extended by other applications.

8.3.2 Use elements if the information is hierarchical, contains a content model, or could potentially become more complex or hierarchical in the future.

8.3.3 Use attributes when the information is metadata about the element.

8.4 *Naming Conventions*

8.4.1 The following list is a compendium of recommended practices to be used when naming attributes and simple or complex elements.

8.4.1.1 Use the UpperCamelCase convention for elements and the lowerCamelCase convention for attributes.

8.4.1.2 Do not use any punctuation in the element name; that is, underscores, dashes, etc. Use only alphanumeric characters.

8.4.1.3 Acronyms should be all in uppercase. Following a Three (or more) Letter Acronym (TLA), all in uppercase, the next letter is lowercase, that is: TLAlowercase, not TLAUppercase.

8.5 *Common XML Structures*

8.5.1 When defining a list of elements where each has the same element name, nest the list of elements inside a parent element. Form the parent's element name by concatenating the name of the list elements with the word 'List'.

8.5.1.1 For example:

Documents Document Sharing Hub
Full document details: SEMI E121-1017-0423R

```
<SubstrateIDList>
    <SubstrateID>Substrate1</SubstrateID>
    <SubstrateID>Substrate2</SubstrateID>
    :
    <SubstrateID>Substrate25</SubstrateID>
</SubstrateIDList>
```

8.6 *Schema Element Organization* — The following recommendations apply only to the Primary and Secondary Schemas.

8.6.1 With the purpose of readability of the schema and to aid in the understanding of the XML document, the following guidelines are recommended when elements are defined in the XML document or schema:

8.6.1.1 Elements of type ID or KEY should be listed at the top of the schema, and elements of type IDREF or KEYREF should be listed second.

8.6.1.2 Elements of simpleType should be declared first, followed by elements of complexType.

8.7 *XML Namespaces*

8.7.1 A namespace name is an Universal Resource Identifier (URI) that uniquely identifies the namespace. A namespace is a group of names usually with a related purpose and context. The namespace is a globally unique name. This is ensured by using a domain names as part of the namespace definition or a pre-determined identifier. SEMI Documents will use the Universal Resource Naming (URN) convention to define namespaces, suppliers or any other 3rd party suppliers can use either URN or Universal Resource Locator (URL) descriptions.

8.7.2 Each application or document definition is responsible for providing a namespace for the extension of its vocabulary definition. A namespace should be unique. To ensure its uniqueness the definitions of namespaces should follow the rules described below.

8.7.3 *Namespace Syntax*

8.7.3.1 SEMI Namespaces use Uniform Resource Names (URNs). All URNs have the following syntax:

<URN> ::= “urn:” <NID> “:”<NSS>

8.7.3.2 Where <NID> is the namespace identifier and <NSS> is the namespace specific string. The <NID> syntax is limited to upper or lower characters followed by upper or lower characters including a hyphen ‘-’.

```
<NID>   ::= <Letter> [<Letter-Number-Hyphen>]*
<Letter> ::= <upper> | <lower>
<Letter-Number-Hyphen> ::= <upper> | <lower> | <number> | ‘-’
```

8.7.3.3 The namespace identifier is case insensitive thus ‘ABC’ and ‘abc’ refer to the same namespace. All SEMI defined namespaces will use <NID> ::= ‘semi-org’.

8.7.3.4 The <NSS> syntax is limited to upper or lower case characters followed by upper lower characters including a hyphen ‘-’. The <NSS> is composed of more than one field and each field is separated by a period as shown below. No other characters than the ones shown are allowed and should be considered reserved.

```
<NSS> ::= <XML Domain>“.”<SEMI Domain>“.”<Version>“.”<Name>
<XML domain>  ::= <Letter> [ Letter-Number-Hyphen]
<SEMI Domain> ::= <Letter> [ Letter-Number-Hyphen]
<Version>     ::= <Letter> [ Letter-Number-Hyphen]
<Name>        ::= <Letter> [ Letter-Number-Hyphen]
```

8.7.3.5 *<XML Domain>* — A XML domain needs to be included to explain whether this is a schema (xsd) or a web service (ws) by using the following format: <XML domain> = ‘xsd’ for schema or ‘ws’ for web service.

8.7.3.6 *<SEMI Domain> or <Company Domain>* — Include in the SEMI domain identifier the Standard number as assigned by SEMI. For example, Specification for Carrier Management = ‘E087’. In those cases where the Standard has not been approved but a formal document number has been assigned, use this number instead. For example, Equipment Diagnostic Interface = ‘Doc3563’. When the definition is from a third party use the defined company domain. For example, Etch equipment = ‘EtchTool’ or ‘Model-3731-134’.

Documents
Document Sharing Hub
Full document details: SEMI E121-1017-0423R

8.7.3.7 <*Version*> — Properly identifying the revision level of the schema and it is required when there is an approved version of the Standard the namespace definition should include the date that corresponds to the Standard release date or when it was approved. Dates are represented as month (mm) and year (yy) and use the 'Vmmyy' format. This field is not required when SEMI Standards are non-existent. For third party applications it is recommended that the 'Vx-x' form be used. For example, 2001 Feb = 'V0201' or for Version 2.1 = 'V2-1'.

8.7.3.8 <*Name*> — This label corresponds to the actual name of the document to which the namespace belongs. For metadata definitions include the descriptive name of the schema using upper camel case. Example: CarrierManagement, EquipmentDiagnosticAcquisition, EquipmentMetadata, etc. For web services, the <Name> includes the service or interface role and whether it corresponds to the 'binding' or the 'portType' definition. (Refer to UDDI Version 3 from OASIS for WSDL structure.)

8.7.3.8.1 Examples of valid representations of namespaces are:

'urn:semi-org:xsd.E087.V0702.CarrierManagement'

'urn:semi-org:xsd.Doc3509.V12.DataCollectionManagement'

'urn:supplierXX-com:EtchTool.V07.Model123'

'urn:semi-org:ws.E132-1.V0305.SessionManager-binding'

'urn:semi-org:ws.E125-1.V1104.MetadataManager-portType'

8.7.3.8.2 Examples of invalid representation of namespaces are:

'urn:semi.org:xsd.Doc3509.V12.DataCollectionManagement' (Invalid use of period in <SEMI Domain>)

'urn:semi-org:xsd.E087.V7.22.CarrierManagement' (Invalid use of period in <Version>)

8.7.4 *Multiple Default Namespaces*

8.7.4.1 When more than one namespace is used in an XML document or schema, a prefix should be defined for each of the namespaces that uniquely identifies the elements that belong to each of the namespaces. This prefix is used to specify that a local element or attribute name belongs to a particular namespace. These prefixes are used to override attribute values and should be documented by the supplier of the application. The following rules should be applied to prefixes:

8.7.4.1.1 Rule 1: Prefixes should be defined in all uppercase or all lower case letters but not both.

Examples: <Definition xmlns: CMS='urn:semi-org:xsd.E087.V0201.CarrierManagement'>

<CMS:Data> abcd </CMS:Data>

<CMS:Data> xyze </CMS:Data>

</Definition>

8.7.4.1.2 Rule 2: Do not use the same prefix for more than one XML namespace. Prefixes defined with the same characters and in the same order are considered the same even when defined in different letter case.

Example: CMS and cms are equivalent.

8.7.5 *Target Namespace*

8.7.5.1 When more than one namespace is used in schema definition a target namespace should be defined using the 'targetNamespace' format definition. The target namespace definition should include the elementFormDefault = 'qualified' such that the elements defined within the target namespace are all qualified (in the namespace).

For example:

```
<schema targetNamespace="urn:semi-org:xsd.E125.V0303.Metadata"
        xmlns:CEM = "urn:semi-org:xsd.E120.V0303.CommonEquipmentModel"
        xmlns:CMS = "urn:semi-org:xsd.E087.V0304.CarrierManagement"
        elementFormDefault="qualified">
```


Documents Document Sharing Hub

Full document details: SEMI E121-1017-0423R