SEMI E138-0217
SPECIFICATION FOR XML SEMICONDUCTOR COMMON COMPONENTS

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 August 31, 2016. Available at www.semiviews.org and www.semi.org in February 2017; originally published March 2005; previously published May 2016.

1 Purpose

1.1 This Document is a central location for the definition of common XML elements that are not specific to a single standard, but common across multiple standards.

2 Scope

2.1 This Document defines the representation of Error, Data Types, Data Value Types, and Unit which are considered common components. The common components are used in association with other XML documents or schemas generated as part of the XML based SEMI Standard Interface Specifications.

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 The purpose of this Document is to define basic components and to allow for future additions as new common components are discovered and required in conjunction with an XML based interface. Four sets of components are defined at this time in this Standard: Error, Data Types, Data Value Types, and Unit. Future additions to this Standard may be possible when new common components are identified.

4 Referenced Standards and Documents

4.1 *SEMI Standards and Safety Guidelines*

SEMI E5 — SEMI Equipment Communications Standard 2 Message Content (SECS-II)

SEMI E87 — Specification for Carrier Management (CMS)

SEMI E121 — Guide for Style and Usage of XML for Semiconductor Manufacturing Applications

SEMI E125 — Specification for Equipment Self Description (EqSD)

SEMI E132 — Specification for Equipment Client Authentication and Authorization

SEMI E134 — Specification for Data Collection Management

SEMI E145 — Classification for Measurement Unit Symbols in XML

4.2 *Other References*

Extensible Markup Language (XML) 1.0 (Second Edition) — W3C Recommendation, 6 October 2000; http://www.w3.org/TR/2000/REC-xml-20001006

XML Schema Part 1: Structures — W3C Recommendation, 2 May 2001; http://www/w3/org/TR/xmlschema-1

XML Schema Part 2: Datatypes — W3C Recommendation, 2 May 2001; http://www/w3/org/TR/xmlschema-2

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

5 Terminology

5.1 *Abbreviations and Acronyms*

5.1.1 *W3C* — World Wide Web Consortium

SUPERSEDED

Documents Document Sharing Hub

Full document details: SEMI E138-0217

5.1.2 *WWW* — World Wide Web

5.1.3 *XML* — Extensible Markup Language

6 Resources, Organizations, and Standards

6.1 Important resources and organizations used in the generation and definition of this Document are found in:

6.1.1 *XML Schema Part 2: Datatypes* — W3C Recommendation.

7 XML Type Definitions

7.1 *XML Schema* — This Specification includes an XML schema file, 'E138-1-V0709-Schema.xsd', and should be provided with this Document. That XML schema file provides the normative definitions of the types described here.

7.2 *Data Type for Error Handling* — Error handling in most cases even with standard interface specifications is implementation-specific. It is not practical for one specification to define all possible error conditions, error codes, descriptions and supplementary information that is needed to properly recover from the error. Regardless, a common approach to communicating error conditions can simplify error management for standards implementers and application developers.

7.3 *Error Data Structure* — The following data structure defines a common error structure that can be extended with additional implementation-specific information that can be used by other standards or suppliers.

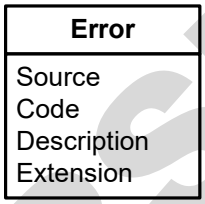

Figure 1
Error Structure

7.3.1 *Source* — The 'Source' attribute represents the document from where the attribute code value is defined. For error codes defined by SEMI Standards, the content of Source is the urn representation of the standard number. For error codes defined by a supplier, it is the urn representation of the supplier's name (i.e., for SEMI E132 errors, the content of 'Source' is: urn:semi-org:E132; for supplier ACME errors the content is: urn:ACME-com). Additional information may be included by the supplier by adding such information after the ':' colon symbol and separating the fields with a '.' period symbol.

Table 1 Example of Error Sources

#	*Source*	*Comment*
1	urn:semi-org:E005	Standard Error Codes for Equipment – SEMI E5.
2	urn:semi-org:E087	Standard Error Codes from Carrier Management.
3	urn:semi-org:E125	Standard Error Codes from Equipment Self-Description.
4	urn:<Supplier>-com[:<SupplierFields>]	Identification of the source of error codes.

7.3.2 *Code* — Error code is an integer value that directly maps to a particular specification error code list. The following is an example of a list of code ranges. This Specification does not require the allocation of any code range by other specification. The following is just an example of codes.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E138-0217

Table 2 Example of Error Code Range

#	*Code Range*	*Comment*
1	1–63, 32768–65535	Standard Error Codes for Equipment – SEMI E5.
2	10000–10500	Equipment Supplier Specific enumerated error codes (Example).
3	5000–5999	Common Standardized Core Components Errors.

7.3.3 *Description* — This is a human readable description of the error code being reported. It maps directly to the definition of the error code for each of the expected result of the services as defined by the specific standard. Additional information may be supplied after this string to provide more detail about the error. Use the ':' colon symbol as delimiter (i.e., Common Component, Code 5001, 'Insufficient Arguments Provided:Argument "SessionID" missing').

Table 3 Example of Error Descriptions

#	*Description*	*Comment*
1	Unknown Object in Object Specifier	Error description for Code '1' as defined in SEMI E5.
2	Loadport does not exist	Error code '49' description from SEMI E87 and defined in SEMI E5.
3	Operation Not Supported	Error description for Error Code '5000'.

7.3.4 *Extension* — This attribute allows the supplier to include additional information regarding the error code encountered. This field is used when the description does not explicitly cover the problem and more specific information is required.

7.4 *Common Standardized Errors* — This section defines common errors to be reported by any application processing a service. Most of these errors are related to the validation of the service.

Table 4 Common Standardized Error Codes

Code	*Description*	*Comment*
5000	Operation Not Supported [:Operation specific detail]	Description of the actual error encountered. Additional information may be included.
5001	Insufficient Arguments Provided [:Operation specific detail]	One or more arguments are missing in the request.
5002	Invalid Arguments Provided [:Operation specific detail]	One or more arguments in the request are invalid or incompatible.

8 Data Types

8.1 The following sections define data types that can be used in conjunction with a parameter definition (for example, see SEMI E125). A defined parameter in an equipment metadata has an associated TypeDefinition data type that provides a defined format. Parameters have references to a defined data type via the typeName. Figure 2 shows a class that can be used to describe re-usable data types. TypeDefinitions can be re-used elsewhere in the metadata wherever a Parameter's type can be declared.

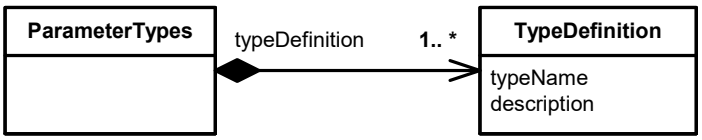

Figure 2
Describing Parameter Type Definitions

SUPERSEDED
Documents Document Sharing Hub
Full document details: SEMI E138-0217

8.1.1 *ParameterTypes* — This class defines all available *TypeDefinitions* that parameters can use to define the data format used to report value or content.

8.1.1.1 *ParameterType Association Definition Table*

Table 5 ParameterType Association Definition

Association Role Name	*Definition*	*Comments*
typeDefinition	The type definition being referenced.	One or more TypeDefinition classes used to represent and report a parameter data value or content.

8.1.2 *TypeDefinition* — The class from which all type description classes are derived. This class has two attributes: typeName and description. A parameter TypeDefinition can be a named SimpleTypeSpecifier, Array, Structure, Enumeration, or Variable class.

Table 6 TypeDefinition Attribute Definition

Attribute Name	*Definition*	*Required*	*Form*
typeName	The name for the type definition.	Y	Text, shall be unique across all defined TypeDefinitions. SimpleTypeSpecifier names shall use the names defined in Table 7, column 1. The remaining types (i.e., Array, Structure, and Variable) names are defined by the equipment or the standard from where the type originates.
description	A description of the meaning or purpose of the type definition.	Y	Text.

8.2 *TypeDefinitions* — The types described here should be considered a minimum set that could be refined and extended to work best with the type system in use. Figure 3 shows a basic set of classes that can be used to describe parameter types.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E138-0217

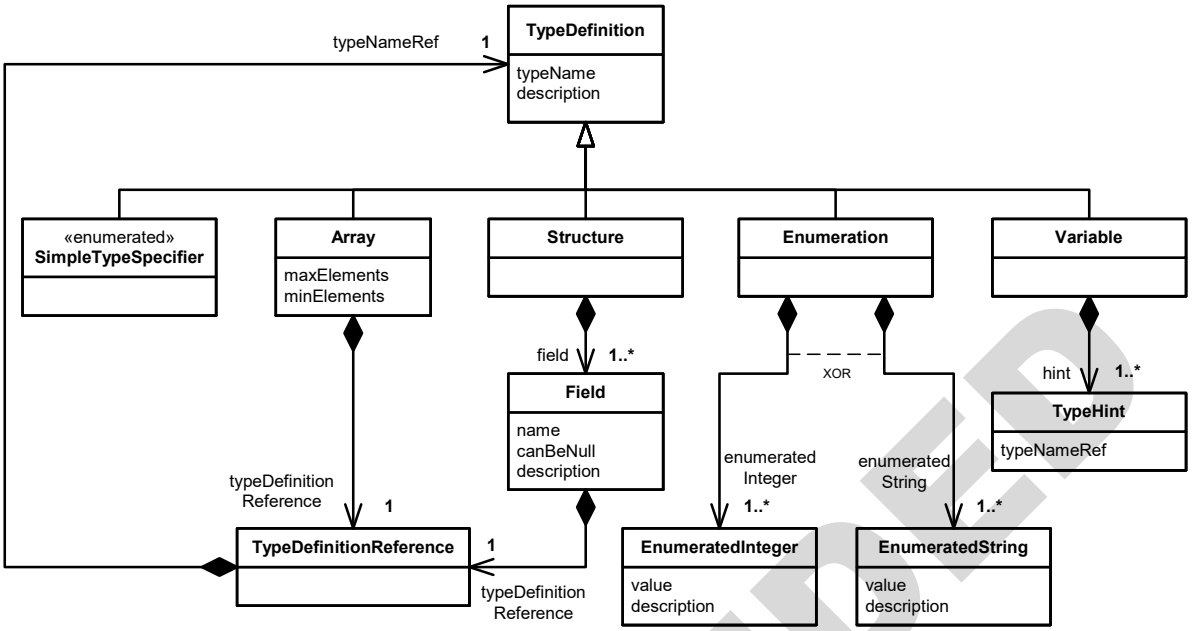

Figure 3
Describing Parameter Types

8.2.1 *SimpleTypeSpecifier* — The SimpleTypeSpecifier class maps the type definitions to common types as defined by SECS-II (see SEMI E5).

8.2.1.1 *SimpleTypeSpecifier Association Definition Table*

Table 7 SimpleTypeSpecifier Definition

typeName	*Enumeration Value*	*Definition*	*Comments*
double	F8	This simple type specifies a double type	The names included in column 1 shall be used when defining the name for the simple data types only. These types map directly to types defined in SEMI E134 and SECS-II. Additional mappings may be possible if a specific technology implementation type collection is required.
float	F4	This simple type specifies a float type	
long	I8	This simple type specifies a long type	
int	I4	This simple type specifies a int type	
short	I2	This simple type specifies a short type	
byte	I1	This simple type specifies a byte type	
string	S	This simple type specifies a string type	
dateTime	D	This simple type specifies a dateTime type	
boolean	B	This simple type specifies a boolean type	
anyURI	URI	This simple type specifies a anyURI type	
base64Binary	B64	This simple type specifies a base64Binary type	

8.2.1.1.1 *SECS-II Type Mapping*

8.2.1.1.1.1 Many equipment suppliers may provide data items via the traditional SECS/GEM control interface that can be described via SEMI E125. For specific types supported in SECS-II, see the following table.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E138-0217