SEMI E164-0712
SPECIFICATION FOR EDA COMMON METADATA

This Standard was technically approved by the global Information & Control Technical Committee. This edition was approved for publication by the global Audits and Reviews Subcommittee on June 18, 2012. Available at www.semiviews.org and www.semi.org in July 2012.

Contents

1 PURPOSE 2
2 SCOPE 2
3 LIMITATIONS 2
4 REFERENCED STANDARDS AND DOCUMENTS 2
4.1 SEMI Standards 2
5 TERMINOLOGY 3
5.1 *Abbreviations and Acronyms* 3
5.2 *Definitions* 3
6 CONVENTIONS 3
6.1 Requirements Identification 3
6.2 *Parameter Listings* 5
7 COMMON METADATA OVERVIEW 6
8 PREREQUISITES 6
8.1 Required Standards 6
9 REQUIREMENTS 7
9.2 *General Metadata Requirements* 7
9.3 *Content Standards Requirements* 14
APPENDIX 1 46
APPENDIX 2 61
RELATED INFORMATION 1 73
RELATED INFORMATION 2 78

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E164-0712

1 Purpose

1.1 The purpose of this Specification is to promote commonality among implementations by defining common representations and conventions of equipment metadata based on SEMI E125 Specification for Equipment Self-Description.

2 Scope

2.1 The scope of this Specification includes:

- Common approaches for defining specific elements of equipment metadata
- The metadata representation of selected SEMI Communication Standards, including those widely used in 300 mm semiconductor factories

2.2 This Specification defines equipment metadata based on SEMI E125-0710 and SEMI E120-0310.

2.3 This Specification applies only to production equipment that act on substrates.

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 This Specification defines metadata representations for data from selected SEMI Communication Standards.

3.1.1 This Specification is not intended to require those Standards be implemented in equipment. If a Standard is not implemented, its associated data would not be required.

3.2 Intellectual property protection is a known issue that is related data collection. A mechanism for data security may exist.

3.2.1 An EDA client's access to the items specified in an equipment's metadata may be subject to user-controlled limitations.

3.2.1.1 SEMI E132-0310 (§ 10.2.10.1) provides for the definition of Privileges by equipment suppliers and by other SEMI Specifications. Privileges may exist that grant or limit access to specific data.

3.2.1.2 Client-by-client data access limitations are not within the scope of this Document.

3.2.2 Data security and intellectual property protection mechanisms are beyond the scope of this Specification.

4 Referenced Standards and Documents

4.1 *SEMI Standards and Safety Guidelines*

SEMI E30 — Generic Model for Communications and Control of Manufacturing Equipment (GEM)

SEMI E39 — Object Services Standard: Concepts, Behavior, and Services

SEMI E40 — Standard for Processing Management

SEMI E87 — Specification for Carrier Management (CMS)

SEMI E90 — Specification for Substrate Tracking

SEMI E94 — Specification for Control Job Management

SEMI E116 — Specification for Equipment Performance Tracking (EPT)

SEMI E120 — Specification for the Common Equipment Model (CEM)

SEMI E125 — Specification for Equipment Self-Description (EqSD)

SEMI E128 — Specification for XML Message Structures

SEMI E132 — Specification for Equipment Client Authentication and Authorization

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E164-0712

SEMI E134 — Specification for Data Collection Management (DCM)

SEMI E138 — XML Semiconductor Common Components

SEMI E145 — Classification for Measurement Unit Symbols in XML

SEMI E148 — Specification for Time Synchronization and Definition of the TS-Clock Object

SEMI E157 — Specification for Module Process Tracking

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

5 Terminology

5.1 *Abbreviations and Acronyms*

5.1.1 *EDA* — equipment data acquisition

5.1.2 *EFEM* — equipment front-end module

5.1.3 *FIMS* — front-opening interface mechanical standard

5.1.4 *Uid* — unique identifier

5.1.5 *uuid* — universally unique identifier

5.2 *Definitions*

5.2.1 *camelCase* — the practice of writing compound words or phrases in which each word is capitalized and all spaces, hyphens, and underscores are removed. When abbreviations are included, they retain the all-capitals form (e.g., AMHSPort).

5.2.2 *content Standards* — for the purpose of this Specification, the term *content Standards* refers to the collection of Sstandards containing definitions of data and behavior for the equipment which are to be represented in the equipment metadata within the scope of this Specification. See ¶ 9.1 for a list of the content Standards.

5.2.3 *equipment data acquisition (EDA)* — refers to the collection of SEMI Standards that define the XML/SOAP-based data collection interface and consists of SEMI E125, SEMI E132, SEMI E134, SEMI E120, and those Specifications that these Standards reference.

5.2.4 *equipment front-end module (EFEM)* — A hardware component that provides access for carriers to the equipment. An EFEM encapsulates load ports, carrier locations, carrier handlers, and other associated mechanisms. In many cases, the EFEM also includes a substrate handler.

5.2.5 *FIMS location* — the 'docked' position of a FOUP at a load port where the FOUP may be opened and wafers inserted or removed.

5.2.6 *fixed buffer equipment* — production equipment that has only fixed load ports and no internal buffer for carrier storage. Substrates are loaded and unloaded directly from the carrier at the load port for processing.

5.2.7 *internal buffer* — a set of locations within the equipment to store carriers. These locations exclude load ports.

5.2.8 *internal buffer equipment* — equipment that uses an internal buffer.

5.2.9 *unique identifier (Uid)* — SEMI E120 Nameable instance identifier based on uuid.

5.2.10 *universally unique identifier (uuid)* — Defined in ISO/IEC 11578.

5.2.11 *undocked location* — the position of a FOUP at a load port where the FOUP may be loaded or unloaded from the equipment. See also the FIMS location definition.

6 Conventions

6.1 *Requirements Identification*

6.1.1 The following notation specifies the structure of requirement identifiers.

6.1.1.1 The following requirements prefix format is used at the beginning of requirement text. See Table 1 for the format notation of the requirements prefix.

SUPERSEDED

Documents Document Sharing Hub

Full document details: SEMI E164-0712

- [Esss.ss-RQ-nnnnn-nn]

6.1.1.2 To mark the end of the requirement text the following suffix format is used.

- [/RQ]

6.1.1.3 Requirements in the body text are highlighted with a border and light green background (may appear gray in black and white printouts).

[Esss.ss-RQ-nnnnn-nn] Requirement text. [/RQ]

Table 1 Requirement Identifiers

Format Notation	*Purpose*
Esss.ss	SEMI Standards Specification identifier. Examples: E087.00, E087.01, E134.00
RQ	Indicates this is a requirement identifier.
nnnnn	Unique five-digit number within this Specification. 90000–99999 are reserved for use by SEMI.
.nn	Two-digit number that indicates version level of the requirement (.00 is used for the first version of a requirement)
/RQ	Indicates the end of a requirement

6.1.2 Requirements in tables are delimited in one of two ways.

6.1.2.1 Where the requirement occupies an entire row in the table, the requirement ID is placed in column 1. No '[/RQ]' is used to mark the end of the requirements text in this case. The table may also contain rows that are not requirements. In this case, the column 1 cell is left blank.

6.1.2.2 Non-requirements text related to a requirements row may be included in an adjacent row. The relationship between the rows is indicated by a broken line separating the two.

6.1.2.3 Where the requirement occupies only one cell, the text in the cell includes the requirement ID prefix and suffix similar to requirements in the body text.

6.1.2.4 No cell in a table will contain both requirement and non-requirement text.

6.1.2.5 Cells that contain requirements are shaded light green.

6.1.2.6 Table 2 provides an example of requirements in a table. Note that in this example, the same requirement is presented in two alternative formats. In practice, mixing the two approaches in the same table is not typically done.

Table 2 Example Table with Requirements

RequirementID	*Statement*	*Additional Detail*
Esss.ss-RQ-nnnnn-nn	The light shall be blue	There shall be no similar color in the light panel.
	The blue light is typically placed at the leftmost position in the light panel.	
	[Esss.ss-RQ-nnnnn-nn] The light shall be blue. There shall be no similar color in the light panel. [/RQ]	The blue light is typically placed at the leftmost position in the light panel.

6.1.3 Only text marked with the requirement identifier is a requirement of this Specification.

6.1.3.1 Clarification, examples, and related recommendations may be provided near a requirement, but are not part of the requirement.

6.1.3.2 Note that the word "should" is used in some non-requirements text and it denotes a recommendation or a best practice, not a requirement.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E164-0712

6.1.4 Parent-child relationships of requirements are noted in the tables in Appendix 1. Where a parent requirement includes conditions or selection criteria, these are passed to their children. For example, if a parent requirement is stated to apply only to photolithography equipment, any child requirements also apply only to such equipment. The condition need not be restated for each child requirement.

6.2 *Parameter Listings*

6.2.1 Data variables from the content Standards are represented in the equipment metadata as Parameters. The mapping of the data variables to Parameters is specified in tables using the format of Table 3. Table 4 explains the contents of each column in the Parameter Table example.

Table 3 Example Parameter Table

Requirement ID	*EDA Parameter*	*Req*	*isTransient*	*SECS Variable*	*Type*	*Explanation*
[Esss.00-RQ-00xxx-00]	Parmeter1	Y	Unrestricted	DataVariable1	N/S	See E999-0707, ¶ 4.3.5.

Table 4 Parameter Table Columns

Column	*Description*
RequirementID	Requirement identifier defined according to ¶ 6.1
EDA Parameter	Name of the Parameter as it appears in the XML metadata files for the associated content Standard.
Req	Whether the Parameter is required: Y – Required N – Not Required C – Conditional (required in specified circumstances)
isTransient	Value of the isTransient attribute of the Parameter: Transient Unrestricted Restricted.
SECS Variable	The name of the SECS data variable or object attribute represented by this metadata Parameter. This parameter is used in EDA wherever these variables are used in SECS (e.g., for corresponding events).
Type	Type of SECS data variable: SV – Status Value DVVAL – Data Value N/S – Not Specified
Explanation	Text explaining conditional requirements and/or referencing the content Standard specification of the variable.

6.2.2 The column labeled Req shows whether a Parameter is required for conformance to this Specification.

6.2.2.1 For Parameters based upon a corresponding variable from a content Standard, the Req value is generally based on the Specification of that variable in the content Standard. If the content Standard requires the variable, this Specification nearly always requires it. If the content Standard specifies conditions, those are generally replicated here.

6.2.2.2 Where the variable is explicitly not required in the content Standard, the corresponding Parameter is specified as not required. Parameters listed as not required are typically included so that when the implementer chooses to include them, they are implemented consistently.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E164-0712

6.2.2.3 Note that Parameters that are required or conditional are specified within the context of the parent requirements. For example, in Table 9, PRJobState is specified as required. A parent[1] requirement, RQ-00061, is conditional upon support by the equipment of SEMI E40. PRJobState inherits that condition. Thus, the PRJobState is also conditional upon SEMI E40 support by the equipment. The ultimate parent requirement for this Specification, RQ-00002, also includes a condition that, in this example, means that if PRJobState is not included in the equipment's SECS interface, it is not required in EDA.

7 Common Metadata Overview

7.1 SEMI E125 specifies equipment self-description. The equipment describes itself using metadata in XML format as defined in SEMI E125.

7.2 Equipment metadata is obtained from the equipment using seven services specified by SEMI E125:

- GetEquipmentStructure
- GetEquipmentNodeDescriptions
- GetSEMIObjTypes
- GetStateMachines
- GetExceptions
- GetTypeDefinitions
- GetUnits

7.3 The seven responses combine to describe the equipment for the purpose of data collection. There are many references from the metadata in one response to the contents of another response.

7.3.1 The equipment structure provides the basis for the definition. The EquipmentNodeDescriptions add parameters, exceptions, SEMI objects, and state machines to each equipment component. The SEMI objects, state machines, exceptions, type definitions, and units are defined separately and included by reference.

7.4 The specification of metadata in SEMI E125 and its subordinate Standard, SEMI E120 focuses primarily on the form of the metadata. It does not specify content.

7.5 In order to achieve the desired commonality among equipment metadata implementations, this Standard specifies content for selected parts of the metadata.

7.5.1 Part of these requirements relate to general content. This includes the content of certain identifiers. For example, EventIDs are a key part of each SEMI E134 event report. The format for the EventID is specified in ¶ 9.2.5.6.

7.5.2 More specific metadata content requirements draw from concepts defined by commonly used SEMI Communication Standards such as SEMI E40 (ProcessJobs), SEMI E87 (Carriers and Load Ports), and SEMI E90 (Substrates). Each of the content Standards listed in ¶ 9.1 are addressed in separate subsections.

7.6 In addition to the requirements in this Document, this Specification also includes XML files containing the actual renderings of the content files into SEMI E125 equipment metadata. These files support one version of each of the content Standards. Other versions can be supported by following the directions in Appendix 2. This is discussed further below.

8 Prerequisites

8.1 *Required Standards*

[E164.00-RQ-00001-00] Conformance to this Specification requires conformance to SEMI E125-0710 and its subordinate Specifications and associated XML schemas and WSDL files. [/RQ]

8.1.1 Note that the SEMI E125-0710 Standard specifies conformance to SEMI E120-0310.

[1] This requirement is actually the parent of the parent requirement. Such conditions cascade to each succeeding inheritance level.

SUPERSEDED
Documents
Document Sharing Hub
Full document details: SEMI E164-0712

9 Requirements

[E164.00-RQ-00002-00] Conformance to this Specification requires conformance to all other identified requirements in this Document as conditions apply. The exception to this requirement is that any Parameter derived from a content Standard that is not implemented in the SECS interface is not required to be implemented in the EDA Common Metadata. [/RQ]

9.1 The requirements in this Specification are divided into two parts. The first part contains requirements that are generally applicable to metadata sets. The second part contains those requirements that are specific to the individual content Standards addressed by this Specification. Each of the content Standards is addressed in a separate subsection. The content Standards are:

- SEMI E30 — Generic Model for Communications and Control of Manufacturing Equipment (GEM)
- SEMI E40 — Standard for Processing Management
- SEMI E87 — Specification for Carrier Management (CMS)
- SEMI E90 — Specification for Substrate Tracking
- SEMI E94 — Specification for Control Job Management
- SEMI E116 — Specification for Equipment Performance Tracking (EPT)
- SEMI E148 — Specification for Time Synchronization and Definition of the TS-Clock Object
- SEMI E157 — Specification for Module Process Tracking

9.1.1 Table A1-1 catalogs the requirements for this Specification.

9.2 *General Metadata Requirements*

9.2.1 In this section requirements are defined that apply generally to all applicable constructs within a metadata set. These requirements are reflected in the attached metadata files (see ¶ 9.3 for more detail). Some requirements in this section apply to all metadata content that models elements taken from other SEMI Standards. This includes the content Standards listed above and others not include here, such as SEMI E109, Reticle and Pod Management.

9.2.1.1 SEMI E125 and SEMI E120 include a number of string elements. Some of the elements are required, for example Uid and Name for Nameables. Other elements are optional and can be excluded, such as Supplier, Make, and Model for EquipmentElement. When such strings are supplied in the metadata, it is important that they contain accurate and useful information.

[E164.00-RQ-00003-00] Unless SEMI E125 or SEMI E120 explicitly allows it, included string elements in the metadata shall not be zero length strings or values that do not satisfy the intent of the attribute, including "Unknown", "NA", "N/A", and "None". [/RQ]

9.2.1.1.1 Note that RQ-00003 does not apply to the values of Parameters defined for NodeDescriptions and SEMIObjTypes because their values are provided through SEMI E134 services, not in the metadata.

9.2.1.2 An equipment's metadata is intended to model only the components included on that instance of the equipment.

[E164.00-RQ-00004-00] The metadata shall not describe optional Modules that do not exist on the equipment. [/RQ]

9.2.1.3 In most cases, the metadata should not describe optional Subsystems and IODevices that do not exist on the equipment.

9.2.2 *EquipmentNodeDescription*

9.2.2.1 The EquipmentNodeDescription is at the heart of the equipment metadata. It links the static equipment structure to the metadata components that provide behavior and dynamic data.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E164-0712

9.2.3 *Equipment Structure*

9.2.3.1 The SEMI E120 Standard specifies how to create the equipment structure definition. The equipment structure provides the static context information about the equipment and its components as sources of data and behavior. The metadata equipment structure should match that of the equipment it models from a functional point of view. It is important that the metadata model promotes understanding of the equipment and accessibility to the data without sacrificing accuracy. It should relate the equipment function to its structural components. See Related Information 1 for more discussion.

9.2.3.2 *Nameable*

9.2.3.2.1 Each node in the equipment structure is a type of Nameable. Types of Nameable include Equipment, Module, Subsystem, IODevice, MaterialLocation, and LogicalElement. Each Nameable is linked to an EquipmentNodeDescription.

9.2.3.3 *EquipmentElement*

9.2.3.3.1 Several attributes for EquipmentElement are allowed to be excluded. However, there are some elements which, when included, need other attributes to provide needed context.

[E164.00-RQ-00005-00] Any EquipmentElement that has a Model attribute value included shall also have an accurate value for the Supplier attribute. [/RQ]

[E164.00-RQ-00006-00] Any EquipmentElement that has an ImmutableID value included shall also have accurate values for the Supplier and Model attributes. [/RQ]

9.2.3.4 *SoftwareModule*

9.2.3.4.1 The SoftwareModule version attribute is used to help identify changes to the equipment software components.

[E164.00-RQ-00007-00] The equipment metadata shall contain a SoftwareModule to represent the main equipment control software. Additional SoftwareModules shall be included for any other software components that can change independently from the main equipment control software. [/RQ]

9.2.3.5 *LogicalElement*

9.2.3.5.1 A LogicalElement defines a persistent, non-structural element in the equipment structure. Examples of LogicalElements include the MaterialManager and JobManager, which are discussed in the immediately following sections, and also the SEMI E116 EPTTracker object discussed in ¶ 9.3.8. See also the discussion of persistent objects in ¶ 9.3.2.5.5.

9.2.3.5.2 *Material*

9.2.3.5.2.1 The content Standards define various SEMI objects and parameters that are related to material (see ¶ 9.3).

9.2.3.5.2.2 MaterialManager is a SEMI E120 LogicalElement defined for the purpose of holding material-related object instances and parameters. The MaterialManager function of grouping together these material-related items serves to organize information that would otherwise be defined in the NodeDescription for the Equipment.

[E164.00-RQ-00008-00] The equipment metadata shall include a MaterialManager, modeled as a LogicalElement associated with the Equipment node. The value for the name and ElementType attributes shall be "MaterialManager". [/RQ]

9.2.3.5.2.3 Figure 1 illustrates the MaterialManager concept. It shows that the MaterialManager may include material-related Parameters, SimpleEvents, Exceptions, and object instances. The actual content is determined by requirements in ¶ 9.3 for the various content Standards and by the implementer.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E164-0712

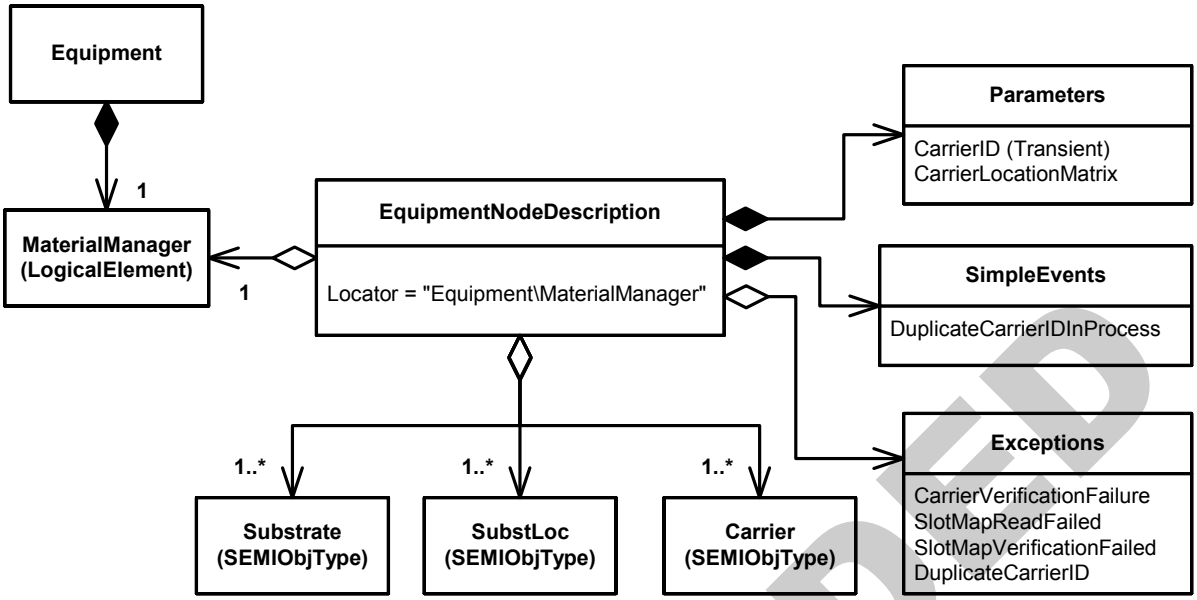

Figure 1
MaterialManager Concept

9.2.3.5.3 *Jobs*

9.2.3.5.3.1 The content Standards define various SEMI objects and parameters that are related to processing activities or jobs (see ¶ 9.3).

9.2.3.5.3.2 JobManager is a SEMI E120 LogicalElement defined for the purpose of holding job-related object instances and parameters. The JobManager function of grouping together these job-related items serves to organize information that would otherwise be defined in the NodeDescription for the Equipment.

[E164.00-RQ-00009-00] The equipment metadata shall include a JobManager, modeled as a LogicalElement associated with the Equipment. The value for the name and ElementType attributes shall be "JobManager". [/RQ]

9.2.3.5.3.3 In ¶ 9.3, specific object instances and parameters from the content Standards are designated for inclusion in the JobManager. Figure 2 illustrates the JobManager concept. It shows that the JobManager may include job-related Parameters, Exceptions, and object instances. The actual content is determined by requirements in ¶ 9.3 for the various content Standards and by the implementer.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E164-0712

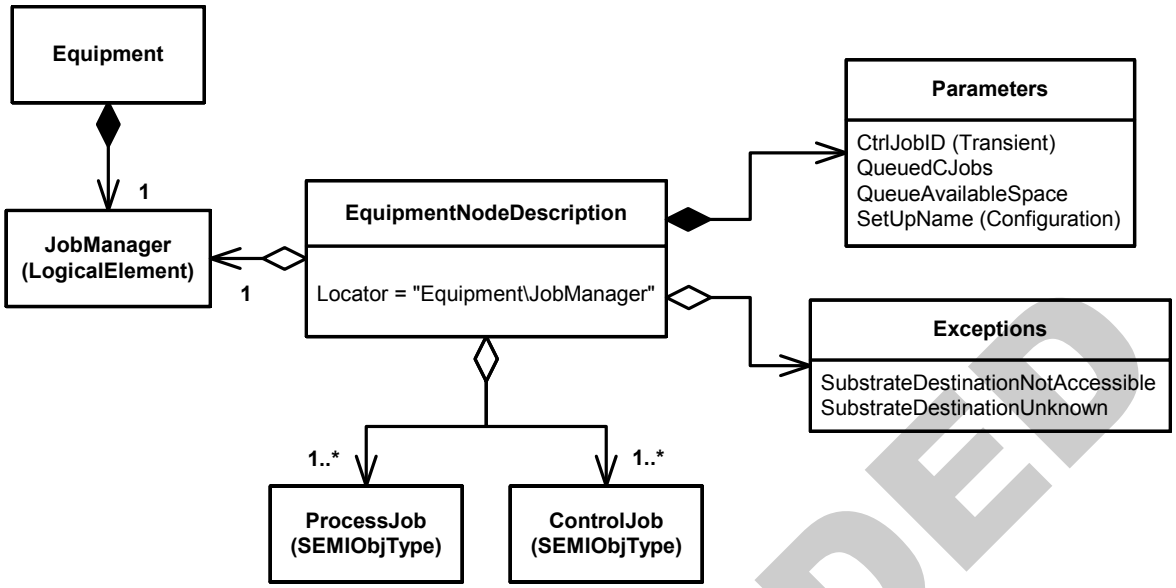

Figure 2
JobManager Concept

9.2.3.6 *Equipment*

9.2.3.6.1 Each metadata set has an Equipment node at the top of the equipment structure hierarchy. There should be only one Equipment node in the equipment structure except in the case of linked equipment (e.g., a coat/bake/develop track linked to wafer stepper).

9.2.3.6.2 SEMIObjTypes that apply to the entire equipment should be modeled as part of the EquipmentNodeDescription for the Equipment or as part of a LogicalElement of the Equipment.

9.2.3.7 *IODevice*

9.2.3.7.1 IODevices are intended to represent equipment components with direct links to the physical equipment via sensors and actuators.

[E164.00-RQ-00010-00] Each IODevice shall include at least one Parameter that represents an input to the IODevice from a sensor or an output from the IODevice to an actuator. [/RQ]

9.2.3.7.2 In most cases, control parameters representing actuators should be modeled in the same IODevice as the sensor values that they affect.

9.2.3.7.3 Other parameters that are closely related to sensor values should also be modeled in the same IODevice as the sensor.

9.2.3.7.4 For example, an IODevice with a parameter measuring tank level might also contain parameters for level set point as well as parameters for maximum level and minimum level settings.

9.2.3.8 *MaterialLocation*

[E164.00-RQ-00011-00] With the exception of slots in carriers, all locations defined in the equipment metadata to hold Substrates, Carriers, or ProcessDurables shall be modeled as SEMI E120 MaterialLocations. [/RQ]

9.2.3.8.1 For the details of modeling slots in carriers, see requirement RQ-00220 in ¶ 9.3.6.5.

9.2.3.8.2 A parameter is specified by RQ-00012 to hold the identifier of the material in a MaterialLocation. This provides a uniform way to access this information in all MaterialLocations defined in the metadata.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E164-0712

[E164.00-RQ-00012-00] All MaterialLocations that are not batch locations (see SEMI E90) shall have a Parameter that identifies the material currently held in the MaterialLocation. That parameter shall be formatted as a string and named "MaterialID." [/RQ]

9.2.3.8.3 If the MaterialLocation is for Carriers and SEMI E87 is supported, then the MaterialID would contain the Carrier identifier (see ¶ 9.3.5.7 RQ-00126).

9.2.3.8.4 If the MaterialLocation is for Substrates and SEMI E90 is supported, the MaterialID would contain the Substrate identifier (See ¶ 9.3.6.5 RQ-00218).

9.2.3.8.5 Batch locations are defined in SEMI E90. They are exempted from RQ-00012 because there is no standard batch identifier defined. MaterialID is allowed for batch locations, but not required.

9.2.3.8.6 All other types of MaterialLocations should use standard defined values for MaterialID if available.

9.2.4 *Objects*

9.2.4.1 According to SEMI E125-0710, ¶ 10.8.1.1, each SEMIObjType must include a Parameter in its Attribute set that contains its instance ID.

[E164.00-RQ-00013-00] The SEMIObjType Attribute Parameter that contains the object instance ID shall have a name attribute equal to ObjID. [/RQ]

9.2.5 *StateMachines*

9.2.5.1 StateMachines can appear complex because they combine several elements to describe behavior, including States, Transitions, Events, and embedded StateMachines. Below, each of these elements is considered, along with the SimpleEvent construct used for discrete events.

9.2.5.2 It is very useful to know ahead of time the exact IDs of standard StateMachines, States, Transitions, and, most especially, Events. In a SEMI E134 EventReport, it is the EventID that is included.

[E164.00-RQ-00014-00] The stateMachineId attribute of each StateMachine representing a state model defined by a SEMI Standard shall take the form <standard ID>:<state model>, where <standard ID> is of the form Vxxx-mmyy where 'V' is the SEMI Standard volume designator, 'xxx' is the Standard number (1–3 characters, no leading zeros), 'mmyy' is the month and two-digit year of Standard release, and <state model> is the name of the state model as it appears in the Standard, presented in camelCase. [/RQ]

9.2.5.3 For example, using a recent version of SEMI E87, the LoadPortTransfer StateMachine ID would be: E87-0709:LoadPortTransfer.

9.2.5.4 *States*

[E164.00-RQ-00015-00] The ID attribute of a State which is defined in a StateMachine based on a SEMI Standard state model shall be the name of that State as specified in that SEMI Standard, presented in CamelCase, except in the case where this will result in duplicate ID values. If the rule above will result in duplicate ID values, the <state machine> value as defined in RQ-00014, followed by a colon shall be prepended to the name of the State from the SEMI Standard as necessary to eliminate the duplication. [/RQ]

9.2.5.4.1 Note that SEMI E125-0710, Table 38 requires that the State ID attribute be "unique across all states in the top level StateMachine".

9.2.5.4.2 For example, where an embedded StateMachine B exists within StateMachine A, the ID for the NoState for the embedded StateMachine will conflict with the parent StateMachine's NoState. In this case, the ID for NoState in the embedded StateMachine could be "B:NoState". The ID for NoState in the parent could remain "NoState", since the ID duplication has been eliminated.

9.2.5.4.3 *Top State*

[E164.00-RQ-00016-00] The Top State's stateName and stateId attributes shall equal the <state machine> portion of the stateMachineId that contains it (see RQ-00014) unless doing so conflicts with mandated state names as defined in one of the content Standards. [/RQ]

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E164-0712

9.2.5.4.3.1 For example, the SEMI E40 ProcessJob StateMachine might have a StateMachine ID of E40-0709:ProcessJob. The Top State for that StateMachine would have stateName and stateId attributes equal to "ProcessJob".

9.2.5.4.3.2 An example of an exception for RQ-00016 is found in the SEMI E87 Carrier state model. For this state model, SEMI E87 requires a Carrier state which is not the Top State to contain the three embedded StateMachines: CarrierIDStatus, CarrierSlotMapStatus, and CarrierAccessingStatus. Since the state name "Carrier" was already used, the Top State could be named CarrierObject. This would avoid the naming conflict.

[E164.00-RQ-00017-00] The Top State of a StateMachine shall be neither the source nor the target of any transition. [/RQ]

9.2.5.4.4 *NoState*

[E164.00-RQ-00018-00] Each StateMachine shall include a "NoState" pseudo-state as described in SEMI E125-0710, Section 10.7.3.3.2.1. [/RQ]

9.2.5.4.4.1 The NoState represents the initial and (if needed) final pseudo-states of the StateMachine.

9.2.5.4.4.2 SEMI E125-710 specifies that "NoState" be the name of this pseudo-state (that is, "NoState" is the last part of the <state path name>). For example, the SEMI E90 SubstrateObject StateMachine would have a Top State with stateName "SubstrateObject". This would contain a NoState with stateName "SubstrateObject.NoState".

9.2.5.5 *Transitions*

[E164.00-RQ-00019-00] Transitions defined for a StateMachine shall correspond to each state change that can occur in practice. [/RQ]

9.2.5.5.1 Transitions are reportable to the EDA client only if a corresponding Event is defined.

[E164.00-RQ-00020-00] Within a StateMachine based on a SEMI Standard state model, each Transition's ID attribute value shall be the transition number, if one is specified in that Standard. The transition number may be amended as specified in requirement RQ-00021. [/RQ]

[E164.00-RQ-00021-00] Within a StateMachine based on a SEMI Standard state model, transitions that end in a History or Conditional symbol shall be represented as multiple Transitions, one for each possible destination state. Each resulting Transition shall have a unique ID made up of its original transition number appended by a letter (e.g., 3a, 3b, etc.). [/RQ]

9.2.5.6 *Events*

[E164.00-RQ-00022-00] Within a StateMachine based on a SEMI Standard state model, each Event's ID attribute value shall be of the form <state machine>:<transition number>:<source state>-<target state> where <state machine> is the name of the top level state machine as it appears in the Standard, <transition number> is the number of the Transition as it appears in the SEMI Standard definition of the state model, and the <source state> and <target state> are the values from the associated Transition's sourceStateID and targetStateID attributes respectively. Excepted from this requirement are any Events specified in the source SEMI Standards to be used for multiple Transitions. [/RQ]

9.2.5.6.1 Among the content Standards, the exceptions to RQ-00022 are the SEMI E30 Control state model events Offline, Local, and Remote and the SEMI E116 event EPTStateChangeEvent.

9.2.5.7 *StateMachineInstances*

9.2.5.7.1 In SEMI E125, StateMachines can be used by multiple StateMachineInstances. When different equipment components exhibit the same behavior (that is, the same states, transitions and events), it is expected that the same StateMachine definition will be used by each of the equipment components.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E164-0712

9.2.5.7.2 *CurrentState and PreviousState*

9.2.5.7.2.1 The StateMachineInstance includes attributes that reference parameters to hold the value of the current state and the previous state of the StateMachine as it applies to the EquipmentNodeDescription that holds the StateMachineInstance.

[E164.00-RQ-00023-00] The Parameters referenced by a StateMachineInstance's CurrentState and PreviousState attributes shall be defined with isTransient="Unrestricted" and shall hold the stateID attribute value of the state being referenced. [/RQ]

9.2.5.8 *SimpleEvents*

[E164.00-RQ-00024-00] When a SEMI Standard defines a discrete event that is independent of a state model, it shall be modeled in the metadata as a SimpleEvent. The ID attribute of the SimpleEvent shall be set to the name of the event from the source Standard, presented in CamelCase format. The name attribute of the SimpleEvent shall have the form

urn:semi-org:event:<standard ID>:<SimpleEvent ID>

where <standard ID> follows the definition used in RQ-00014 and <SimpleEvent ID> is the ID attribute of this SimpleEvent. [/RQ]

[E164.00-RQ-00025-00] Where a SEMI Standard specifies a state model with more than one state, it shall be represented in the equipment metadata as a SEMI E125 StateMachine. [/RQ]

9.2.6 *Exceptions*

9.2.6.1 Exceptions should be referenced by the EquipmentNodeDescriptions that represent the equipment component to which that exception pertains.

9.2.6.2 If Exceptions from multiple sources within the equipment represent the same condition, they should have the same ExceptionID.

9.2.6.3 The Exception Description field text should provide a clear picture of the nature of the problem and, in conjunction with the ExceptionParameters, provide the user with enough information to diagnose the cause of the exception.

9.2.6.4 An Exception should report any Parameters that describe the status of the source of the exception, especially any sensor data. It should also report any Parameters related to external influences on the source of the exception, including substrates, process jobs, carriers, status of related systems, etc.

9.2.7 *Parameters*

9.2.7.1 A Parameter should be a part of the EquipmentNodeDescription which it describes or should be included in a LogicalElement that is part of that EquipmentNodeDescription.

9.2.7.2 *AvailableParameters*

9.2.7.2.1 The purpose of the AvailableParameter construct is to designate the transient Parameters that will have valid values available for reporting with a given event.

[E164.00-RQ-00026-00] Each Parameter referenced in an AvailableParameter construct shall have the isTransient attribute set to Transient. [/RQ]

9.2.7.2.2 AvailableParameter constructs are found in SimpleEvents and in StateMachineInstances (within EventMaps).

9.2.7.3 *ParameterTypes*

9.2.7.3.1 The use of the Structure data type to hold process data presents problems in the collection and use of that data. Important process parameters are collected using trace reports. If the parameter of interest is available only in a structure, then the entire structure must be collected and reported at each trace interval. This significantly increases

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E164-0712

the amount of data that must be transferred via the EDA interface and stored by the factory system, but with no added value.

[E164.00-RQ-00027-00] Non-transient, numeric parameter data shall be defined as individual Parameters or arrays. Numeric data include those with data types of integer, floating point, or Boolean. [/RQ]

9.2.7.3.2 These data values may additionally be included in Parameters formatted as Structures if necessary.

[E164.00-RQ-00028-00] Parameters that report numeric data shall not be defined as string type unless mandated in the SEMI Standards. [/RQ]

9.2.7.3.3 Date and time values are not considered to be numeric data for the purpose of RQ-00028.

9.2.7.3.4 *Unlimited Strings and Arrays*

9.2.7.3.4.1 Data constructs that are of an undetermined size are difficult for many database systems to handle. Wherever possible, the maximum size should be specified.

[E164.00-RQ-00029-00] The metadata shall include simple string type definitions named using the format "Stringx" where x is one or more numeric characters and represents the maximum number of characters that any Parameter using this data type can contain. [/RQ]

9.2.7.3.4.2 For example, a Parameter of type String80 would have a maximum of 80 characters. The EDA Common Metadata xml files contain type definitions for String80, String128, and String255. See also ¶ 9.3.2.4.

9.2.7.3.4.3 The maximum length should be kept as small as practical. Very large strings are handled inefficiently by most databases.

9.2.7.3.4.4 The "Stringx" data types should be used in place of the unlimited length "string" data type.

9.2.7.3.4.5 Any use of unlimited strings, should be documented and justified by the supplier.

[E164.00-RQ-00030-00] Parameters shall not be defined as unlimited arrays (MaxElements = 0 in XML). This includes arrays within structures. [/RQ]

9.2.7.4 *Units*

[E164.00-RQ-00031-00] The equipment metadata shall employ unit symbols, prefixes, powers, and names from SEMI E145 wherever applicable. [/RQ]

[E164.00-RQ-00032-00] Combined units shall be formed using the syntax defined in SEMI E145. [/RQ]

9.2.8 *Mapping SECS Data to E125 Equipment Metadata*

9.2.8.1 Data variables that are available via the SECS-II interface should typically be available via the EDA interface as well. SEMI E125-0710 supports and requires mapping of SECS Data Variables, Alarms, and Events to the corresponding constructs in the equipment metadata.

9.2.8.2 As a result, each SECS VID (variable identifier) should typically appear in the equipment metadata mapped to one or more Parameters.

9.3 *Content Standards Requirements*

9.3.1 The requirements relating to each content Standard are included within this section (¶ 9.3). The content Standards are listed in ¶ 9.1.

9.3.2 *EDA Common Metadata XML Files*

9.3.2.1 Included in the requirements for this Standard are the contents of the attached XML files. These files address the modeling of each of the content Standards in SEMI E125 equipment metadata.

9.3.2.2 For each of the content Standards, a separate requirement is included to address the XML file content. For example, see RQ-00037 in ¶ 9.3.3.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E164-0712

9.3.2.3 The XML files are delivered in a .zip format with relative directory structure included.

9.3.2.4 Included in the XML file collection are two files, listed in Table 5, that hold type definitions and units that apply to more than one of the Standards.

[E164.00-RQ-00033-00] All equipment metadata shall use the type definitions and units defined in the files listed in Table 5 they apply to the content Standard-specific XML file content. [/RQ]

Table 5 XML Files The Apply To Multiple Content Standards

RequirementID	*File Name*	*File Location Within collection*
[E164.00-RQ-00034-00]	CommonGetTypeDefinitions.xml	\EDA 0710 Common Metadata
[E164.00-RQ-00035-00]	CommonGetUnits.xml	\EDA 0710 Common Metadata

9.3.2.5 *Conventions Used in the EDA Common Metadata XML Files*

9.3.2.5.1 When constructing the EDA Common Metadata XML files, some conventions were used to promote consistency in the results. Some of these conventions are noted here as an aid to the reader.

9.3.2.5.2 *General*

9.3.2.5.2.1 All name and ID fields formatted as strings are populated using CamelCase notation unless there is a content Standard requirement that prevents it.

9.3.2.5.2.2 Where possible, description fields are populated including text from the content Standard that specifies the corresponding element.

9.3.2.5.3 *Parameters*

9.3.2.5.3.1 To create the representations of the content Standards in the EDA Common Metadata files, SECS data variables and object attributes were translated into EDA parameters. This is noted in various tables in ¶ 9.3 (for example, see Table 7).

9.3.2.5.3.2 SECS uses a simplistic view of the equipment. It does not specify the equipment component that is the source of a data variable. In many cases, state model events (CEIDs) represent all instances of an object or equipment component. When the event report is sent, it represents a single instance. To support this style of reporting, DVVALs were created so that a value can be reported for whichever instance is the subject of a particular event report.

9.3.2.5.3.3 EDA provides a more sophisticated view of the equipment. The equipment structure is provided. Independent StateMachineInstances are defined so that separate data collection plans can be defined for each possible source of the event. This means that unique data parameters for each source can be included in the event report for that source. For this reason many SECS DVVALs are translated into EDA non-transient (unrestricted) parameters. In some content Standards a SECS SV and DVVAL existed for the same data. Some of these are combined into a single EDA parameter.

9.3.2.5.3.4 Note that there remain some cases where transient parameters are needed in EDA. Some SECS DVVALs were translated to transient parameters.

9.3.2.5.4 *Dynamic Objects*

9.3.2.5.4.1 Dynamic objects are those that are created and destroyed over time, such as SEMI E40 ProcessJobs. Persistent objects are those that are consistently present on the equipment, such as the SEMI E90 Substrate Location located in the process module.

9.3.2.5.4.2 SEMI E125 SEMIObjTypes are used to model dynamic objects defined in the content Standards listed in ¶ 9.1.

9.3.2.5.4.3 In this Specification, each SEMIObjType attribute has been defined as a non-transient (isTransient = Unrestricted) parameter in the Attributes collection of the SEMIObjType. Where the object is taken from one of the

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E164-0712

content Standards, each of these parameters is typically named to match the standard attribute, but formatted in camelCase.

9.3.2.5.4.4 The attribute values are non-transient in relation to an object instance. During the lifetime of the object instance, its attributes are always available for reporting or ad hoc data collection. The attribute names are the same for each instance of the object and are supplied in the SEMIObjType. The attribute for a specific object instance is differentiated using the Source, which includes the ObjID (see SEMI E134.1-0710, ¶ 7.2.5.5).The metadata specifies only the SEMIObjType definitions.

9.3.2.5.4.5 As shown in Figure 3, the SEMIObjType is a template for each object instance. Data collection plans are typically defined in terms of the SEMIObjType (that is, applying to all instances), but each event report provides data for the specific instance that generated the event.

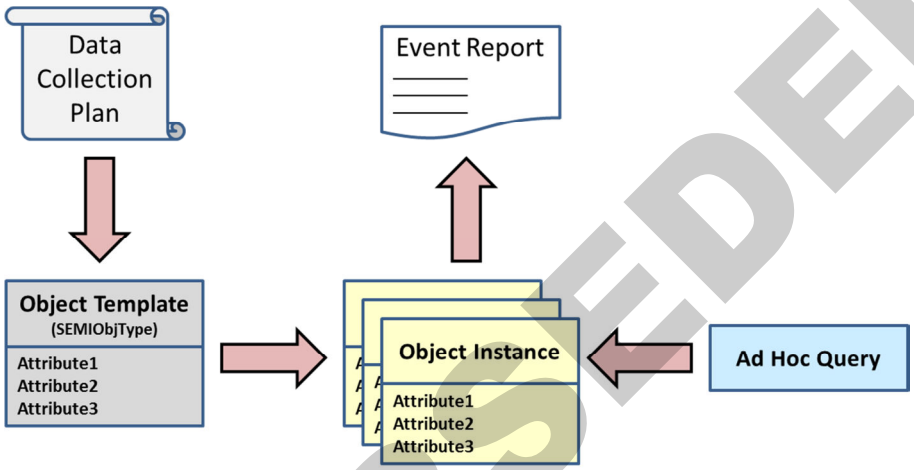

Figure 3
Illustration of SEMIObjType Concept

9.3.2.5.4.6 Any data variables (SECS DVVAL, SV, or ECV) specified in the content Standard that are associated with a SEMIObjType are modeled as Parameters in the EventData collection within the SEMIObjType definition. Where a resulting EventData parameter duplicated the value and usage of an Attribute parameter, the EventData parameter was removed. Where both SV and DVVAL versions of the same value were specified in the content Standard, the DVVAL version was eliminated if not needed (see ¶ 9.3.2.5.3).

9.3.2.5.5 *Persistent Objects*

9.3.2.5.5.1 Persistent objects defined in the content Standards are modeled as LogicalElements (see ¶ 9.2.3.5), MaterialLocations (see ¶ 9.2.3.8), or other equipment components represented by EquipmentNodeDescriptions. This approach allows the instances of these object types to be represented in the metadata. The SEMIObjType construct is not used to model persistent objects because it does not allow specification of object instances in the metadata.

9.3.2.5.5.2 Object attributes and related data variables are represented as parameters in the EquipmentNodeDescription corresponding to the object. As with SEMIObjTypes, duplicate values are eliminated.

9.3.2.5.5.3 For example, Figure 4 illustrates the mapping of the SEMI E90 Substrate Location object to a MaterialLocation (for Substrate Location requirements, see ¶ 9.3.6.5).

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E164-0712

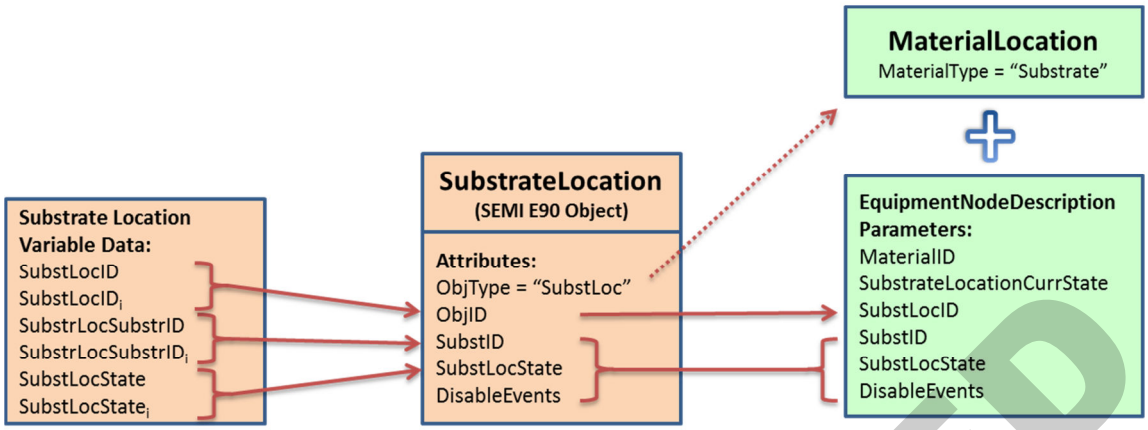

Figure 4
SEMI E87 Substrate Location Mapping to MaterialLocation

9.3.2.5.5.3.1 The Substrate Location variable data all duplicate attribute values. These were combined with the corresponding attributes during the mapping process. The transient form for these values was determined to be adequate and the non-transient form eliminated.

9.3.2.5.5.3.2 The Substrate Location attributes SubstID, SubstLocState, and DisableEvents are represented as parameters of the EquipmentNodeDescription representing the MaterialLocation. The ObjID attribute is represented by the SubstLocID parameter (using the more descriptive data variable name from SEMI E90). The ObjType attribute is not included because it is implied by the MaterialLocation MaterialType value.

9.3.2.5.5.3.3 MaterialID and SubstrateLocationCurrState are included based on requirements unique to this Specification.

9.3.2.5.5.3.4 A similar process was used to map the attributes and variable data of other persistent objects to the corresponding EquipmentNodeDescriptions.

9.3.2.5.6 *StateMachine Transitions*

9.3.2.5.6.1 In the metadata for the content Standards, transitions that are not numbered are assigned transition numbers. These are almost exclusively initial transitions. These initial transitions are most often assigned transition number zero.

9.3.2.5.7 *State Parameters*

9.3.2.5.7.1 Most of the content Standards define parameters to report the state and sometimes the previous state. However, the naming and the values contained are inconsistent.

9.3.2.5.7.2 The StateMachineInstance provides attributes that can reference parameters for current and previous state. For the EDA Common Metadata XML files, a consistent approach is taken for the parameters referenced by these CurrentState and PreviousState attributes.

9.3.2.5.7.3 For all StateMachineInstances, a current state parameter is defined in the metadata. The name of the parameter follows the convention "xxxCurrState", where xxx = the name of the state model in the content Standard. For example, the SEMI E40 ProcessJob StateMachine will yield a parameter named ProcessJobCurrState.

9.3.2.5.7.4 If the StateMachine has only two states (excluding NoState), then the previous state is obvious. For StateMachines with more states, a previous state parameter is also defined in the metadata. Its naming convention is "xxxPrevState" where xxx is the same as above. The previous state parameters are not required since the previous state might not be known in all cases.

9.3.2.5.8 The content Standard's state parameters are also included in the metadata files.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E164-0712

9.3.2.6 *Constructing an Equipment's Metadata from the EDA Common Metadata*

9.3.2.6.1 The EDA Common Metadata XML files contain the metadata constructs defined for each of the content Standards. The files contain the Parameter, StateMachine, Exception, SEMI Object, and SimpleEvent definitions. They also contain supporting definitions for Units and TypeDefinitions.

9.3.2.6.2 The metadata constructs for each content Standard are provided in multiple XML files – one corresponding to each SEMI E125 service (listed in ¶ 7.2) to obtain the metadata from the equipment (for example E30GetEquipmentStructure.xml).

9.3.2.6.3 Each content Standard metadata set contains only those files that are affected by that Standard. They have as few as three and as many as six files. For example, the SEMI E30 XML file set does not contain a file named E30GetSEMIObjTypes.xml because SEMI E30 does not define any objects.

9.3.2.6.4 Constructing the metadata for equipment requires merging the metadata for each of the supported content Standards and then adding the equipment-specific metadata. A suggested merger process is described here.

9.3.2.6.5 The EDA Common Metadata XML files must be modified to fit the substance of the equipment. The implementer must determine which items to include among the items that are conditionally required or optional.

9.3.2.6.5.1 Requirement RQ-00002 says that content Standard parameters not implemented in the SECS interface are not required in EDA. Any such Parameters would be removed during the merging process.

9.3.2.6.5.2 Most of the metadata files contain embedded comments that explain actions or choices the implementer should make during the implementation process. All comments should be read carefully and acted upon as appropriate. These comments should be removed from the final version of the metadata.

9.3.2.6.6 To construct the final metadata for the equipment manually, the user should take the following steps:

1. Identify the content Standards that will be implemented. Begin with the XML files provided in this Specification for those Standards.
2. Identify the version of each Standard that will be supported. For any supported Standard version not directly provided, modify the XML files provided according to the instructions in Appendix 2 that are applicable to that Standard.
3. Update the metadata files by reading and following the directions embedded as comments in each metadata file for each Standard. These represent optional and conditional requirements that the implementer must resolve.
4. For each of the seven responses, combine the metadata from the supported content Standards.
    a. GetUnitsResponse – Concatenate the lists of units provided for each supported content Standard. Add any needed entries from the CommonGetUnits.xml file for this version of EDA. It is suggested that the list be sorted using the unit symbol.
    b. GetTypeDefinitions Response – Concatenate the lists of units provided for each supported content Standard. Add any needed entries from the CommonGetTypeDefinitions.xml file for this version of EDA. It is suggested that the list be sorted using the TypeDefinition name attribute.
    c. GetExceptionsResponse – Concatenate the lists of Exceptions provided for each supported content Standard. It is suggested that the list be sorted using the Exception ID attribute.
    d. GetStateMachinesResponse – Concatenate the collections of StateMachines provided for each supported content Standard. It is recommended that the StateMachines be grouped by Standard with the lowest Standard number first.
    e. GetSEMIObjTypesResponse – Concatenate the collections of SEMIObjTypes provided for each supported content Standard. It is recommended that the SEMIObjTypes be grouped by Standard with the lowest Standard number first.
    f. GetEquipmentStructureResponse – Construct the basic equipment structure for the specific equipment that is being modeled. Then examine the equipment structure in the Common Metadata for each of the supported Standards. In each case, the provided equipment components are templates that focus on the effect of the particular content Standard on a particular type of node. Match the templates for the supported content Standards to the corresponding nodes in the actual equipment structure, and add the corresponding elements where they apply.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E164-0712

g. GetEquipmentNodeDescriptionsResponse – By association with the equipment structure, the NodeDescriptions are representative of a type of node (Equipment, Module, Subsystem, or IODevice). Each NodeDescription provides the Parameters, StateMachineInstances, etc. needed for nodes of that type for that particular Standard.
   i. Combine the elements for the Equipment node for each of the content Standards with those from the other supported Standards and apply them to the actual Equipment node.
   ii. Repeat the process for Modules, Subsystems, IODevices, and LogicalElements. Note that each content Standard metadata set defines only the node types that it affects.
   iii. Apply the elements for the common MaterialLocation nodes to similar types of MaterialLocation nodes (i.e., for Substrates or Carriers).

5. After combining the various Standards' metadata, add any equipment-specific metadata elements to complete the equipment metadata.

9.3.2.6.6.1 All GetEquipmentStructure and GetEquipmentNodeDescription files include comments that contain additional instructions to help with the merge and edit process.

9.3.2.7 *Example Equipment Metadata Sets*

9.3.2.7.1 Also included with the attached XML files are examples of complete SEMI E125 equipment metadata sets. The TrackSysFixedBuffer example is a fixed buffer equipment. The TrackSysInternalBuffer example is an internal buffer equipment.

9.3.2.7.1.1 These files are found in the zip file under the folder \EDA 0710 Common Metadata\TrackSys.

9.3.2.7.1.2 Neither example is intended to represent any particular commercial process equipment.

9.3.2.7.2 These examples combine the required representations of the content Standards and additional equipment specific detail necessary to complete the metadata set.

9.3.2.7.3 The number of process parameters and exceptions included in these examples is far lower than would be expected for a commercial implementation. This helps make review of the examples easier.

9.3.3 *SEMI E30 Generic Equipment Model*

[E164.00-RQ-00036-00] Equipment which implement SEMI E30 shall conform to the requirements in this section, entitled SEMI E30 Generic Equipment Model. [/RQ]

[E164.00-RQ-00037-00] Implementers shall represent the metadata constructs relating to SEMI E30 with the content of the SEMI E30 Common Metadata files listed in Table 6. The content of the files listed in Table 6 shall be modified:
a) According to the instructions embedded as comments in those files, and
b) According to the additional instructions in Appendix 2 for adapting the files to the specific version of SEMI E30 implemented on the equipment. [/RQ]

Table 6 XML Files For SEMI E30

RequirementID	*File Name*	*File Location Within Collection*
[E164.00-RQ-00038-00]	E30GetEquipmentNodeDescriptions.xml	\EDA 0710 Common Metadata\E30-0307
[E164.00-RQ-00039-00]	E30GetEquipmentStructure.xml	\EDA 0710 Common Metadata\E30-0307
[E164.00-RQ-00040-00]	E30GetStateMachines.xml	\EDA 0710 Common Metadata\E30-0307
[E164.00-RQ-00041-00]	E30GetTypeDefinitions.xml	\EDA 0710 Common Metadata\E30-0307

9.3.3.1 Note that some content Standards do not require specific content for all seven service responses (listed here as files). In the case of SEMI E30, there is no required content for GetExceptions, GetSEMIObjTypes, or GetUnits.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E164-0712

9.3.3.2 The SEMI E30 metadata files need the following entries from the CommonGetTypeDefinitions.xml file: int, String32, String80, String128, and String255.

[E164.00-RQ-00042-00] Where the equipment supports SEMI E30-0307, the Parameters for the EquipmentNodeDescription representing the Equipment shall conform to Table 7. In the case where a different version is supported, the Table 7 contents shall be modified by the appropriate instructions in Appendix 2, if any. [/RQ]

9.3.3.3 Table 7 and other Parameter tables in § 9.3 list the SECS Data Variables from the content Standards that have been included in the Common Metadata XML files.

9.3.3.3.1 In some cases, the content Standards define multiple Data Variables that hold the same information. These Data Variable pairs which have duplicate content may have the same or different names. The required Parameter lists in this Standard specify only one EDA Parameter to represent such duplicate-content Data Variables.

9.3.3.3.2 The implementer of this Specification is not prohibited from providing the excluded Data Variable as an additional Parameter.

Table 7 Equipment Node Parameters from SEMI E30

Requirement ID	*EDA Parameter*	*Req*	*isTransient*	*SECS Variable*	*Type*	*Explanation*
[E164.00-RQ-00043-00]	ProcessState	C	Unrestricted	ProcessState	SV	See SEMI E30-0307, ¶5.2. Required only if the ProcessingStateChange simple event is defined
[E164.00-RQ-00044-00]	PreviousProcess State	C	Unrestricted	PreviousProcessSta te	SV	See SEMI E30-0307, ¶5.2. Required only if the ProcessingStateChange simple event is defined
[E164.00-RQ-00045-00]	ControlState	Y	Unrestricted	ControlState	SV	See SEMI E30-0307, ¶5.2
[E164.00-RQ-00046-00]	Clock	C	Unrestricted	Clock	SV	See SEMI E30-0307, ¶5.2. Required only if equipment supports E30 Clock capability
[E164.00-RQ-00047-00]	PPChangeName	C	Transient	PPChangeName	DVVAL	See SEMI E30-0307, ¶5.2. Required only on equipment supporting process programs
[E164.00-RQ-00048-00]	PPChangeStatus	C	Transient	PPChangeStatus	DVVAL	See SEMI E30-0307, ¶5.2. Required only on equipment supporting process programs
[E164.00-RQ-00049-00]	PPExecName	C	Unrestricted	PPExecName	SV	See SEMI E30-0307, ¶5.2. Required only on equipment supporting process programs
[E164.00-RQ-00050-00]	PPFormat	C	Unrestricted	PPFormat	SV	See SEMI E30-0307, ¶5.2. Required only if equipment supports SEMI E30 Process Recipe Management capability

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E164-0712

Requirement ID	EDA Parameter	Req	isTransient	SECS Variable	Type	Explanation
[E164.00-RQ-00051-00]	PPError	C	Transient or Unrestricted	PPError	DVVAL	See SEMI E30-0307, ¶ 4.6.2.5, ¶ 5.2. Required only on equipment supporting process programs; it shall be defined as Transient if an event is provided for recipe verification or recipe verification failure; otherwise it shall be defined as Unrestricted
[E164.00-RQ-00052-00]	Operator Command	C	Transient	Operator Command	DVVAL	See SEMi E30-0307, Table 8. Required only if the OperatorCommandIssued event is supported
[E164.00-RQ-00053-00]	Equipment Constant	C	Transient	ECID	n/s	See SEMI E30-0307, ¶ 4.5.4 3rd Bullet; ¶ 6.0 Table 8. Required only if Operator EquipmentConstant Change event is supported.
[E164.00-RQ-00054-00]	DataSetName	C	Transient	DataSetName	DVVAL	See SEMI E30-0307, Table 8. Required only on equipment supporting uploading of large process programs or execution recipes
[E164.00-RQ-00055-00]	Establish Communications Timeout	C	Unrestricted	Establish Communications Timeout	ECV	See SEMI E30-0307, ¶ 5.2. Required only if equipment supports SEMI E30 Establish Communications capability
[E164.00-RQ-00056-00]	TimeFormat	C	Unrestricted	TimeFormat	ECV	See SEMI E30-0307, ¶ 5.2. Required only if equipment supports SEMI E30 Clock capability, SEMI E90 (for TimeIn/TimeOut), or SEMI E116 (for TransitionTimestamp)
[E164.00-RQ-00057-00]	ControlCurrState	Y	Unrestricted	n/a	n/a	See ¶ 9.3.2.5.7
[E164.00-RQ-00058-00]	ControlPrevState	N	Unrestricted	n/a	n/a	See ¶ 9.3.2.5.7

[E164.00-RQ-00059-00] The equipment metadata shall include a representation of the SEMI E30 Control state model associated with the Equipment node. [/RQ]

[E164.00-RQ-00060-00] The equipment metadata shall include a representation of a Processing StateMachine in accordance with SEMI E30 and associated with the Equipment node. [/RQ]

9.3.3.4 No Processing StateMachine or StateMachineInstance is provided in the SEMI E30 metadata files. Note that SEMI E30 requires these for the implementation. Examples are provided in the TrackSys metadata sets. The Processing state Parameters (ProcessState and PreviousProcessState) are provided in the SEMI E30 metadata files.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E164-0712

9.3.4 *SEMI E40 Processing Management*

[E164.00-RQ-00061-00] Equipment which implement SEMI E40 shall conform to the requirements in this section, entitled SEMI E40 Processing Management. [/RQ]

[E164.00-RQ-00062-00] Implementers shall represent the metadata constructs relating to SEMI E40 with the content of the SEMI E40 Common Metadata files listed in Table 8. The content of the files listed in Table 8 shall be modified:
a) According to the instructions embedded as comments in those files, and
b) According to the additional instructions in Appendix 2 for adapting the files to the specific version of SEMI E40 implemented on the equipment. [/RQ]

Table 8 XML Files For SEMI E40

RequirementID	*File Name*	*File Location Within Collection*
[E164.00-RQ-00063-00]	E40GetEquipmentNodeDescriptions.xml	\EDA 0710 Common Metadata\E40-0709
[E164.00-RQ-00064-00]	E40GetEquipmentStructure.xml	\EDA 0710 Common Metadata\E40-0709
[E164.00-RQ-00065-00]	E40GetSEMIObjTypes.xml	\EDA 0710 Common Metadata\E40-0709
[E164.00-RQ-00066-00]	E40GetStateMachines.xml	\EDA 0710 Common Metadata\E40-0709
[E164.00-RQ-00067-00]	E40GetTypeDefinitions.xml	\EDA 0710 Common Metadata\E40-0709

9.3.4.1 In the case of SEMI E40, there is no unique required content for GetExceptions or GetUnits.

9.3.4.2 The SEMI E40 metadata files need the following entries from the CommonGetTypeDefinitions.xml file: boolean, long, String80, String128, String255, PauseEventArray, RecipeVariable, RecipeVariableArray, SlotArray, and CarrierSlotList.

9.3.4.3 The SEMI E40 metadata files need the following entries from the CommonGetUnits.xml file: non-dimensional quantity.

[E164.00-RQ-00068-00] The SEMI E40 ProcessJob object shall be modeled in the equipment metadata as a SEMI Object associated with the EquipmentNodeDescription representing the JobManager LogicalElement. [/RQ]

[E164.00-RQ-00069-00] Where the equipment supports SEMI E40-0709, the Parameters for the ProcessJob SEMIObjType shall conform to Table 9 and Table 10. In the case where a different version is supported, the Table 9 and Table 10 contents shall be modified by the appropriate instructions in Appendix 2, if any. [/RQ]

Table 9 ProcessJob Object Attribute Parameters from SEMI E40

Requirement ID	*EDA Parameter*	*Req*	*isTransient*	*SECS Variable*	*Type*	*Explanation*
[E164.00-RQ-00070-00]	ObjID	Y	Unrestricted	ObjID	Attr	See SEMI E40-0709, Table 2, ¶ 8.2.2.4, ¶ 11.2, SEMI E40.1 ¶ 7.1. PRJobID DVVAL is not needed in EDA.
[E164.00-RQ-00071-00]	ObjType	Y	Unrestricted	ObjType	Attr	See SEMI E40-0709, Table 2, ¶ 11.2, SEMI E40.1 ¶ 7.1

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E164-0712

Requirement ID	EDA Parameter	Req	isTransient	SECS Variable	Type	Explanation
[E164.00-RQ-00072-00]	PauseEvent	N	Unrestricted	PauseEvent	Attr	See SEMI E40-0709, Table 2, ¶ 11.2, SEMI E40.1, ¶ 7.1. PauseEvent DVVAL is not needed in EDA.
[E164.00-RQ-00073-00]	PRJobState	Y	Unrestricted	PRJobState	Attr	See SEMI E40-0709, Table 2, ¶ 8.2.2.4, ¶ 11.2, SEMI E40.1 ¶ 7.1. PRJobState DVVAL is not needed in EDA.
[E164.00-RQ-00074-00]	PRMtlNameList	Y	Unrestricted	PRMtlNameList	Attr	See SEMI E40-0709, Table 2, ¶ 11.2, SEMI E40.1, ¶ 7.1. PRMtlNameList DVVAL is not needed in EDA.
[E164.00-RQ-00075-00]	PRMtlType	Y	Unrestricted	PRMtlType	Attr	See SEMI E40-0709, Table 2, ¶ 11.2, SEMI E40.1, ¶ 7.1. PRMtlType DVVAL is not needed in EDA.
[E164.00-RQ-00076-00]	PRProcessStart	N	Unrestricted	PRProcessStart	Attr	See SEMI E40-0709, Table 2, ¶ 11.2, SEMI E40.1, ¶ 7.1. PRProcessStart DVVAL is not needed in EDA.
[E164.00-RQ-00077-00]	PRRecipeMethod	Y	Unrestricted	PRRecipeMethod	Attr	See SEMI E40-0709, Table 2, ¶ 11.2, SEMI E40.1, ¶ 7.1. PRRecipeMethod DVVAL is not needed in EDA.
[E164.00-RQ-00078-00]	RecID	Y	Unrestricted	RecID	Attr	See SEMI E40-0709, Table 2, ¶ 11.2, SEMI E40.1, ¶ 7.1. RecID DVVAL is not needed in EDA.
[E164.00-RQ-00079-00]	RecVariableList	N	Unrestricted	RecVariableList	Attr	See SEMI E40-0709, Table 2, ¶ 11.2, SEMI E40.1, ¶ 7.1. RecVariableList DVVAL is not needed in EDA.

Table 10 ProcessJob Object EventData Parameters from SEMI E40

Requirement ID	EDA Parameter	Req	isTransient	SECS Variable	Type	Explanation
[E164.00-RQ-00080-00]	ProcessJobCurrState	Y	Unrestricted	n/a	n/a	See ¶ 9.3.2.5.7
[E164.00-RQ-00081-00]	ProcessJobPrevState	N	Unrestricted	n/a	n/a	See ¶ 9.3.2.5.7

[E164.00-RQ-00082-00] The ProcessJob SEMI Object shall include a StateMachineInstance of a StateMachine that models the SEMI E40 Process Job State Model. [/RQ]

9.3.5 *SEMI E87 Carrier Management*

[E164.00-RQ-00083-00] Equipment which implement SEMI E87 shall conform to the requirements in this section, entitled SEMI E87 Carrier Management. [/RQ]

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E164-0712

[E164.00-RQ-00084-00] Implementers shall represent the metadata constructs relating to SEMI E87 with the content of the SEMI E87 Common Metadata files listed in Table 11. The content of the files listed in Table 11 shall be modified:
a) According to the instructions embedded as comments in those files, and
b) According to the additional instructions in Appendix 2 for adapting the files to the specific version of SEMI E87 implemented on the equipment. [/RQ]

Table 11 XML Files For SEMI E87

RequirementID	*File Name*	*File Location Within Collection*
[E164.00-RQ-00085-00]	E87GetEquipmentNodeDescriptions.xml	\EDA 0710 Common Metadata\E87-0709
[E164.00-RQ-00086-00]	E87GetEquipmentStructure.xml	\EDA 0710 Common Metadata\E87-0709
[E164.00-RQ-00087-00]	E87GetExceptions.xml	\EDA 0710 Common Metadata\E87-0709
[E164.00-RQ-00088-00]	E87GetSEMIObjTypes.xml	\EDA 0710 Common Metadata\E87-0709
[E164.00-RQ-00089-00]	E87GetStateMachines.xml	\EDA 0710 Common Metadata\E87-0709
[E164.00-RQ-00090-00]	E87GetTypeDefinitions.xml	\EDA 0710 Common Metadata\E87-0709

9.3.5.1 In the case of SEMI E87, there is no unique required content for GetUnits.

9.3.5.2 The SEMI E87 metadata files need the following entries from the CommonGetTypeDefinitions.xml file: boolean, short, String80, String128, and LocationStateEnum.

9.3.5.3 The SEMI E87 metadata files need the following entries from the CommonGetUnits.xml file: Non-dimensional quantity.

9.3.5.4 *SEMI E87 Equipment Node*

[E164.00-RQ-00091-00] Where the equipment supports SEMI E87-0709, the Parameters of the EquipmentNodeDescription representing the Equipment s-hall conform to Table 12. In the case where a different version is supported, the Table 12 contents shall be modified by the appropriate instructions in Appendix 2, if any. [/RQ]

Table 12 Equipment Node Parameters from SEMI E87

Requirement ID	*EDA Parameter*	*Req*	*isTransient*	*SECS Variable*	*Type*	*Explanation*
[E164.00-RQ-00092-00]	BypassReadID	C	Unrestricted	BypassReadID	ECV	See SEMI E87-0709, ¶ 19.2.1 Table 37; Must be included at Equipment or at Load Port level, but not both.
[E164.00-RQ-00093-00]	CarrierHold Trigger	Y	Unrestricted	CarrierHold Trigger	ECV	See SEMI E87-0709, ¶ 15.2
[E164.00-RQ-00094-00]	Unclamp ControlTrigger	Y	Unrestricted	Unclamp ControlTrigger	ECV	See SEMI E87-0709, ¶ 15.4

9.3.5.4.1 The BypassReadID is listed as a conditional parameter for the Equipment node and for each Load Port node. Since SEMI E87-0709 requires that BypassReadID exist in SECS, it must be included at one level or the other in the metadata. See also SEMI E87-0709, ¶ 10.7.7.

9.3.5.5 *SEMI E87 Carrier Object*

[E164.00-RQ-00095-00] The SEMI E87 Carrier object shall be modeled in the equipment metadata as a SEMI Object associated with the MaterialManager LogicalElement. [/RQ]

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E164-0712