



**Equipment Data Acquisition (EDA) Change Summary: November  
2005 to July 2010**

**International SEMATECH Manufacturing Initiative  
Technology Transfer #09055011B-ENG**

**Advanced Materials Research Center, AMRC, International SEMATECH Manufacturing Initiative, and ISMI** are servicemarks of SEMATECH, Inc. **SEMATECH** and the **SEMATECH** logo are registered servicemarks of SEMATECH, Inc. All other servicemarks and trademarks are the property of their respective owners.

**Equipment Data Acquisition (EDA) Change Summary: November 2005  
to July 2010  
Technology Transfer #09055011B-ENG  
International SEMATECH Manufacturing Initiative  
June 30, 2010**

**Abstract:** As a result of a “freeze” declared by International SEMATECH Manufacturing Initiative (ISMI) members, all commercial implementations of the Equipment Data Acquisition (EDA) interface (i.e., Interface A) have been based on the 1105 version of the related SEMI standards. In 2010, ISMI’s members declared a new EDA freeze that includes improvements made to the SEMI standards in the intervening time. This new freeze is based on the standards current as of the release date of the 0710 version. This document from the MFGM065M project lists and explains those changes from the 1105 (2005) to the 0710 (2010) release.

**Keywords:** Data Management Standards, e-Diagnostics, e-Manufacturing, Equipment Data Acquisition, Factory Automation

**Authors:** Lance Rist

**Approvals:** Lance Rist, Author  
Harvey Wohlwend, Project Manager  
Joe Draina, Director  
Laurie Modrey, Technology Transfer Team Leader



## Table of Contents

1	INTRODUCTION .....	1
1.1	Background .....	1
1.2	Purpose .....	1
1.3	Scope .....	1
1.4	References .....	1
1.4.1	ISMI Documents .....	1
1.4.2	SEMI Standards .....	1
1.4.3	Additional Documents .....	2
1.5	Abbreviations .....	2
1.6	Definitions .....	2
2	OVERVIEW .....	2
3	GENERAL MODIFICATIONS .....	3
3.1	XML Schema Updates .....	3
3.2	SOAP 1.2 Support .....	3
4	SEMI E125 EQUIPMENT SELF DESCRIPTION .....	3
4.1	Conceptual Changes .....	4
4.1.1	StateMachines & SimpleEvents .....	4
4.1.2	SEMI Objects .....	4
4.1.3	Exceptions .....	4
4.1.4	SECS ID Mapping .....	5
4.1.5	Restricted Parameters .....	5
4.1.6	Data Types .....	5
4.1.7	MetadataRevised Construct Optimized .....	5
4.1.8	Nested State Machines .....	5
4.2	Simplifications, Clarifications, and Editorial Changes .....	6
4.2.1	Provisional Status Removed .....	6
4.2.2	Initial/Final States .....	6
4.2.3	State Naming Convention .....	6
4.2.4	SEMI Objects .....	6
4.2.5	Exceptions .....	6
5	SEMI E134 DATA COLLECTION MANAGEMENT .....	6
5.1	Conceptual Changes .....	7
5.1.1	Trace Data Collection .....	7
5.1.2	Condition-based Trace .....	8
5.1.3	DCPDefined/DCPDeleted Notifications .....	8
5.1.4	GetCurrentDateTime Service .....	9
5.1.5	Timestamp for Ad Hoc Data Request .....	9
5.1.6	DCP StateMachine .....	9
5.1.7	Description and Name Attributes .....	10
5.1.8	Data Types .....	11
5.1.9	Privileges .....	11
5.2	Clarifications and Editorial Changes .....	11
5.2.1	Provisional Status Removed .....	11
5.2.2	DeactivatePlan Behavior .....	11

5.2.3	TimeActivated.....	11
5.2.4	NoValue and Null Usage .....	11
6	SEMI E120 COMMON EQUIPMENT MODEL .....	12
6.1	Conceptual Changes.....	12
6.1.1	MaterialLocations for Consumables and Others.....	12
6.1.2	Nameable uid Value.....	12
6.1.3	EquipmentElement.....	12
6.1.4	LogicalElement .....	13
6.2	Clarifications and Editorial Changes .....	13
6.2.1	AbstractModule.....	13
6.2.2	Locator .....	13
6.2.3	Regular Expressions.....	13
7	SEMI E132 EQUIPMENT CLIENT AUTHENTICATION AND AUTHORIZATION.....	13
7.1	Conceptual Changes.....	13
7.1.1	EnhancedEstablishSession .....	13
7.1.2	InterfaceDiscovery Interface.....	14
7.2	All Sessions Persistent .....	14
7.2.1	Disabling Session Pings .....	14
7.2.2	Number of Sessions per ACL Principal.....	14
7.2.3	SessionClosed Notification .....	14
7.3	Clarifications and Editorial Changes .....	15
7.3.1	E128 Messaging Support .....	15
7.3.2	GetDefinedPrivileges .....	15
8	SEMI E128 XML MESSAGE STRUCTURES .....	15
9	SEMI E138 XML SEMICONDUCTOR COMMON COMPONENTS .....	15
9.1	Conceptual Changes.....	15
9.2	Clarifications and Editorial Changes .....	15
10	SEMI E145 UNIT SYMBOLS IN XML.....	16
	APPENDIX A – SEMI BALLOTS FOR EDA SINCE THE 1105 RELEASE .....	17
	APPENDIX B – SEMI E125 CHANGES FROM E125-1105 TO E125-0710 .....	21
	APPENDIX C – SEMI E134 CHANGES FROM E134-1105 TO E134-0710 .....	26
	APPENDIX D – SEMI E120 CHANGES FROM E120-1104 TO E120-0310.....	30
	APPENDIX E – SEMI E132 CHANGES FROM E132-1105 TO E132-0310 .....	34
	APPENDIX F – SEMI E128 CHANGES FROM CREATION TO E128-0310.....	37
	APPENDIX G – SEMI E138 CHANGES FROM CREATION TO E138-0709 .....	38
	APPENDIX H – SEMI E145 CHANGES FROM CREATION TO E145-0306.....	40

### List of Figures

Figure 1	Exception from SEMI E125-0710.....	4
Figure 2	Trace Report from SEMI E134-0710.....	7
Figure 3	Data Collection Plan from SEMI E134.....	8
Figure 4	DCP Consumer Interface from SEMI E134-0710 .....	9
Figure 5	DCP State Model from SEMI E134-0710.....	10

### List of Tables

Table A-1	EDA Ballots Since 1105 Release .....	17
-----------	--------------------------------------	----



## 1 INTRODUCTION

### 1.1 Background

Equipment Data Acquisition (EDA) is a data collection interface for semiconductor manufacturing equipment. It moves data collection from the SECS link to a new connection using mainstream technologies such as XML, SOAP, and HTTP.

The SEMI standards for EDA were created in 2002–2005. International SEMATECH Manufacturing Initiative (ISMI) members identified the 1105 SEMI release of the standards as mature and declared that all implementations of EDA should implement the 1105 release until further notice. The equipment suppliers were required to limit their EDA implementations to this version through 2010.

In the interim, changes to the EDA standards continued. As more implementations of EDA were created, the industry uncovered various shortcomings and corresponding improvements. In 2010, ISMI members declared a new EDA freeze.

### 1.2 Purpose

This document provides a technical summary of the changes made to the EDA standards between the 1105 (November 2005) and 0710 (July 2010) releases. It is intended to help users and implementers of EDA understand the differences, learn of the benefits, and gauge the amount of work needed to update existing implementations.

### 1.3 Scope

This document focuses on modifications made to the EDA standards as they are represented in the 1105 SEMI publication through the versions current as of the 0710 version.

Conceptual changes to the standards are summarized. Some basic maintenance of the standards is also included, but it is not within the scope of this document to provide a complete, line-by-line explanation of all changes.

SEMI owns the copyright to all SEMI documents, both the standards and the ballot proposals. ISMI cannot provide copies of either.

### 1.4 References

#### 1.4.1 ISMI Documents

*Equipment Data Acquisition Guidance*, ISMI Technology Transfer #0603478C-ENG, ([ismi.semiatech.org/docubase/abstracts/4784ceng.htm](http://ismi.semiatech.org/docubase/abstracts/4784ceng.htm))

#### 1.4.2 SEMI Standards

The following standards can be purchased from SEMI. The technical specifications can be purchased at <http://www.semi.org/en/Standards/StandardsPublications/index.htm>.

Schema and WSDL files are also available at:

<http://dom.semi.org/web/wstandards.nsf/supmaterials>

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*

SEMI E134 – *Specification for Data Collection Management (DCM)*

SEMI E138 – *XML Semiconductor Common Components*

SEMI E145 – *Classification for Measurement Unit Symbols in XML*

SEMI E147 – *Guide for Equipment Data Acquisition (EDA)*

SEMI E151 – *Guide for Understanding Data Quality*

### 1.4.3 Additional Documents

*SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)* – W3C, 27 April 2007;

<http://www.w3.org/TR/2007/REC-soap12-part1-20070427/>

*SOAP Version 1.2 Part 2: Adjuncts (Second Edition)* – W3C, 27 April 2007;

<http://www.w3.org/TR/2007/REC-soap12-part2-20070427/>

## 1.5 Abbreviations

CEM – Common Equipment Model

DCM – Data Collection Management

DCP – Data Collection Plan

SOAP – Simple Object Access Protocol

uid – Unique Identifier – SEMI E120 identifier based on uuid

uuid – Universally Unique Identifier – Defined in ISO/IEC 11578

## 1.6 Definitions

EDA (Equipment Data Acquisition) – The set of SEMI standard specifications that defines the Interface A data communication interface. EDA is defined by SEMI E125 and E134 and their related specifications.

Interface A – The network interface for data communication from equipment to the factory system. Refers to the interface, independent of implementation details.

Restricted Parameter – A non-transient parameter whose value is valid only during specific time periods and not valid at other times.

## 2 OVERVIEW

This document assumes the reader has detailed knowledge of the EDA standards. For an overview of EDA and the EDA standards, see SEMI E147, *Guide for Equipment Data Acquisition (EDA)* and ISMI Technology Transfer 0603478C-ENG, <sup>1</sup> *Equipment Data Acquisition Guidance*.

---

<sup>1</sup> ISMI documents can be found at <http://ismi.sematech.org> by entering the document number or title into the search field.

This document addresses the changes at two levels. Sections 3 through 10 provide a summary of the most notable changes made to each individual EDA standard from the 1105 release to the 0710 release date. Appendix A through Appendix H provides a ballot-by-ballot history and explanation of the changes made by each.

### **3 GENERAL MODIFICATIONS**

#### **3.1 XML Schema Updates**

The most visible change to the EDA standards is a reworking of the XML representation of the messages and data. The 0710 release has reduced XML of the messages in general and of the metadata specifically.

In the original EDA freeze, most elements in the XML identified as “not required” are represented as zero length strings or as specific values, such as “unknown.” In the 0710 release, most of these are omitted when not needed. This can represent a significant decrease in the bulk of the XML.

The original XML mapping also included many unneeded container structures. These containers added both size and complexity to the XML. The 0710 release has simplified the XML by removing several of these container structures and flattening the hierarchical data structures.

Several inconsistencies were found in the mapping of the EDA standards to XML. The cardinalities of some relationships were incorrect. New items were sometimes added into the XML without a corresponding definition in the main specification. Naming was inconsistent in some places. All such known issues have been resolved in the 0710 release.

These changes/corrections to the XML mapping are too numerous and too detailed to effectively convey in this document.

#### **3.2 SOAP 1.2 Support**

The move from SOAP 1.1 to SOAP 1.2, made by SEMI Document #4596, adds another significant change that impacts all of the EDA messaging. While it does not affect the application data in the EDA messages, it does affect the infrastructure.

SOAP 1.2 is the most commonly used version of SOAP today. Since there are no known plans to further update it, it should be a stable version for this important underlying technology. Software development tools provide the best support for this version. Implementations in SOAP 1.2 should be supportable into the foreseeable future.

SOAP 1.2 also opens the door for related web services that may be useful in automating equipment. For example, WS-Security provides message-level security that is easier to use than the SSL component built into EDA today. Other available web services include WS-Addressing, WS-ReliableMessaging, WS-Eventing, and more.

Note that SOAP 1.1 and SOAP 1.2 implementations are not interoperable. However, a computer can offer both SOAP 1.1 and SOAP 1.2 services simultaneously.

### **4 SEMI E125 EQUIPMENT SELF DESCRIPTION**

For more detail, see Appendix B.

## 4.1 Conceptual Changes

### 4.1.1 StateMachines & SimpleEvents

SEMI documents #4151A, line item #4, and #4366, line item #2, created the concept of SimpleEvents. SimpleEvents are an alternative to StateMachines as sources of EventReports for EquipmentNodeDescriptions and SEMI Objects. Use of SimpleEvents allows the implementer to skip defining the underlying state model.

No guidance was included about when to use SimpleEvents versus StateMachines. ISMI recommends using SimpleEvents only for simple state models, such as those with only two states. It is intended that state models defined in other SEMI standards (E40, E87, E90, E94, etc.) be represented as StateMachines.

### 4.1.2 SEMI Objects

In SEMI document #4366, line item #2, the SEMI Object definition was changed to allow it to contain multiple StateMachine instances.

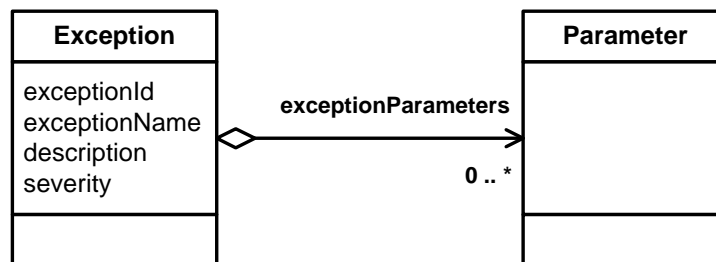
Just as the SEMI E87 Load Port has multiple state models defined for it (Transfer, Access Mode, Reservation, Carrier Association, Clamped/Unclamped, Open/Closed), SEMI Objects may need to hold instances of multiple state models.

### 4.1.3 Exceptions

In SEMI document #4729, the severity attribute of the Exception class changed from an optional string field to a required enumeration. The four values defined include Fatal, Error, Warning, and Informational.

Exceptions were further changed in the 4729 ballot to replace the setData/clearData attributes with a single exceptionData field. With setData/clearData, each Exception had separately defined Parameters. Typically, thousands of exceptions are defined for a piece of equipment. The new exceptionData instead references existing Parameters defined within the equipment. This compresses the size of the metadata and provides more context for the meaning of the Parameter values, since the source is known.

Note in Figure 1 that the relationship from Exception to Parameter has an unfilled diamond, which represents aggregation. Aggregations translate to references to the associated object.



Reprinted with permission from Semiconductor Equipment and Materials International, Inc. (SEMI) © 2010. www.semi.org

**Figure 1 Exception from SEMI E125-0710**

#### 4.1.4 SECS ID Mapping

SEMI document 4822, line item #1, added elements in the metadata to hold the IDs of corresponding constructs from the SECS interface, including SECS collection events (CEIDs), alarms (ALIDs), and variables (VIDs). This mapping makes the transition to EDA simpler by allowing the user to identify quickly the EDA elements that correspond to the familiar SECS data collection items.

Note that the design of the SECS ID mapping allows for multiple SECS IDs to map to a single EDA construct. This is necessary in several instances. For example, if the same state model is applied to multiple equipment components, the events will typically take on different CEIDs in SECS. In EDA, the EventIDs will be the same in each case because the “source” (component identity) is part of the data collection plan (DCP) definition and data report.

#### 4.1.5 Restricted Parameters

The `isTransient` attribute of the `Parameter` class was changed in SEMI Document #4729 from a Boolean to a three-value enumeration. The original definition of a `Transient Parameter` and a `non-Transient Parameter` (now called “`Unrestricted`”) remains essentially unchanged. The new choice, “`Restricted`,” was defined to describe a `Parameter` that does not always report valid values, but is still traceable.

For example, enabling a trace for an end-point detection sensor may gather critical process information. However, when no product is in the chamber, that sensor might be switched off to save power and extend its life. At that time, no valid reading is available. In the past, such a `Parameter` was often classified as “`Transient`,” making it available only for event reporting. This change provides a better option for such `Parameters`.

#### 4.1.6 Data Types

To promote reuse of many of the data types and the tools for defining these data types, this portion was moved from SEMI E125 and SEMI E134 to SEMI E138 (XML Semiconductor Common Components). SEMI E125 references SEMI E138 for the data types.

The data types were improved and streamlined, and the units were separated from the format definition to give greater flexibility in their use. The result is fewer data types are defined. See Section 9 for more detail.

#### 4.1.7 MetadataRevised Construct Optimized

The `MetadataRevised` message from the equipment has been simplified by a change in SEMI document #4729. The `MetadataRevised` message contained redundant information in the form of the `EquipmentElement` (always equal to the `Equipment` instance). By removing `EquipmentElement` and the `RevisionNotice` container, the message content is simplified to contain only the `MetadataRevision` element with the `revisionDate`.

#### 4.1.8 Nested State Machines

In SEMI document #4151A, line item #2, a change was made to allow for state transitions from one “`nested stateMachine`” to another. Harel state models allow for “`concurrent`” states, in which one state is subdivided into multiple concurrent (“`AND`”) states. EDA models these concurrent states as `StateMachines`. The 1105 version of SEMI E125 inadvertently prohibited transition into a `StateMachine` from a `State`, as is required with these concurrent states.

For an example, see the SEMI E87 CarrierID Read/Write state model.

## 4.2 Simplifications, Clarifications, and Editorial Changes

### 4.2.1 Provisional Status Removed

SEMI E125 was elevated from a provisional specification to a full SEMI specification by SEMI document #4366, line item #1. This reflects the acknowledged maturity of the standard.

### 4.2.2 Initial/Final States

In SEMI document #4151, line item #3, a clarification was made to refer to all initial and final states in stateMachine definitions as “NoState.” These are pseudo-states before the state model is created or after the state model is destroyed. They are needed to complete certain transition definitions.

### 4.2.3 State Naming Convention

SEMI document #4151A, line item #5, simplifies the naming convention for states:

~~“urn:semi-org:state:<standard id>:<state machine>:<state path name>”~~

This change significantly compresses StateMachine definitions.

SEMI document #4822, line item #2, defines the simple name for the initial and final states of a StateMachine to be “NoState.” It further defines the metadata representation for default entry transitions to States.

### 4.2.4 SEMI Objects

SEMI Objects have two elements (attributes and eventData), each containing a list of Parameters associated with the SEMI Object. SEMI document #4729 clarified that the two lists cannot overlap.

This was the original intent of these elements, but it had not been documented in the standard. The “attributes” list is intended to contain Parameters that map to the object’s defined attributes. The “eventData” list is intended to contain additional Parameters that provide needed information in associated event reports. Some eventData Parameters may be Transient Parameters, tied directly to Transitions of StateMachineInstances associated with the SEMI Object.

### 4.2.5 Exceptions

A new requirement was added by SEMI document #4729 that any alarms reported to the operator must be modeled as Exceptions. This was the original intent of Exceptions, but it was not documented in the earlier versions of SEMI E125.

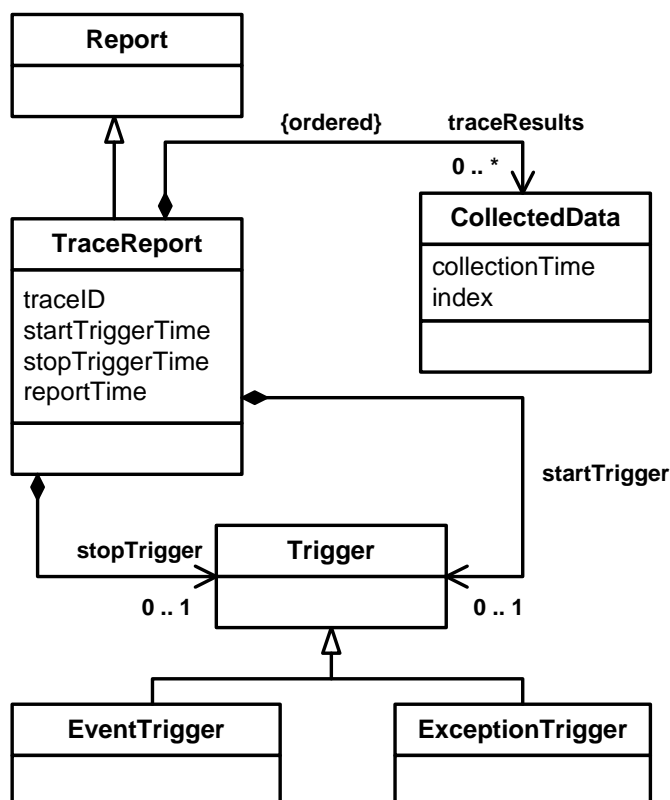
## 5 SEMI E134 DATA COLLECTION MANAGEMENT

For more detail, see Appendix C.

## 5.1 Conceptual Changes

### 5.1.1 Trace Data Collection

Trace data collection has added an “index” to each traceResult (see Figure 2). This allows the client to confirm that all trace reports are received and to order them correctly. At data rates above their capacity, some equipment have dropped messages and/or sent other messages out of order. This will help the client detect such problems, if they occur.



Reprinted with permission from Semiconductor Equipment and Materials International, Inc. (SEMI) © 2010. www.semi.org

**Figure 2 Trace Report from SEMI E134-0710**

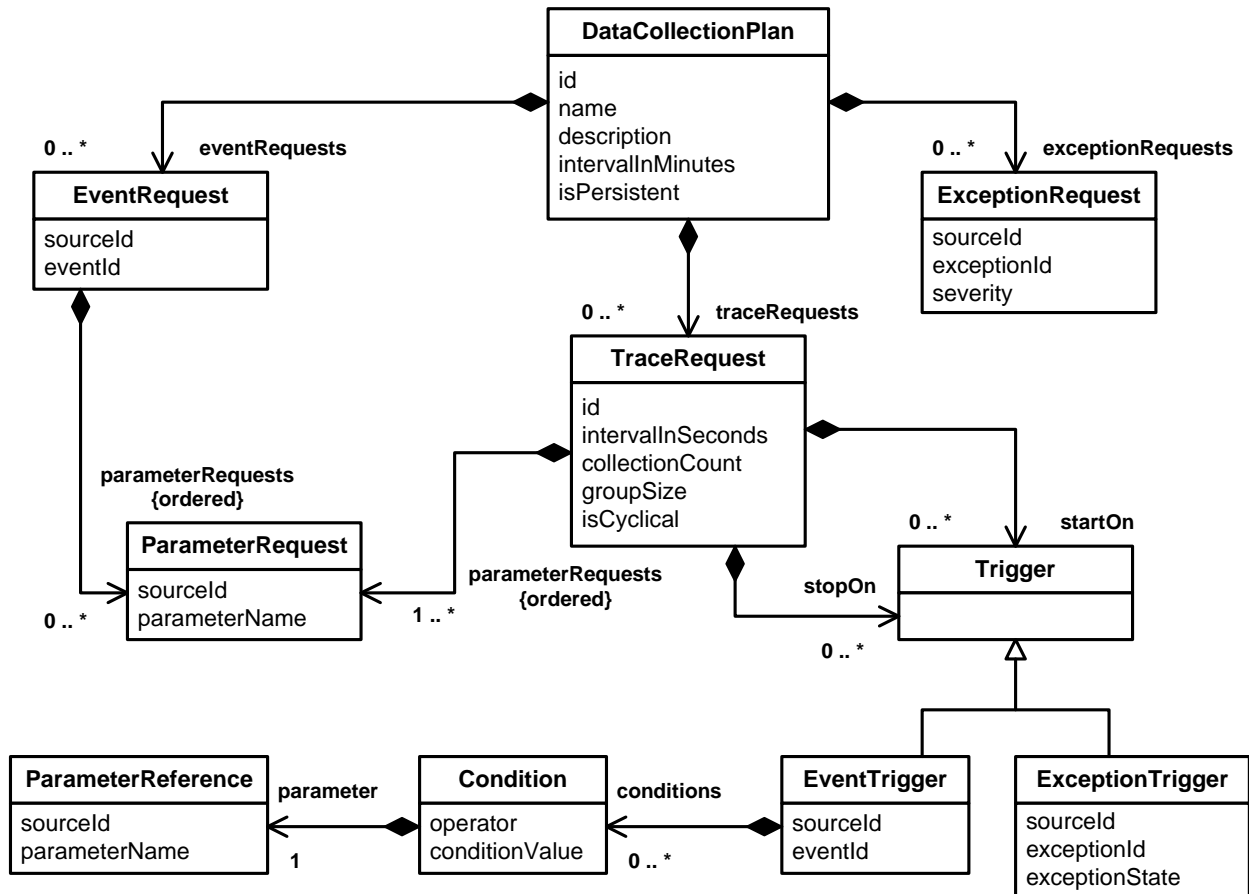
The Trace definition has been changed to clearly define the behavior of an event-triggered trace when an event occurs. When an event initiates the Trace, the first trace report will include the `startTrigger` and `startTriggerTime`. This first trace report has been clarified to occur immediately upon Trace start.

When the Trace is stopped by an event, the last trace report will include the `stopTrigger` and the `stopTriggerTime`. Note that this last trace report may not include any `traceResults` (i.e., `CollectedData`) because the Trace stops immediately upon the event. `TraceReport` cardinality for a trace has been adjusted to a minimum of zero to reflect this.

Both changes were made by SEMI document #4729.

### 5.1.2 Condition-based Trace

SEMI document #4823, line item #1, expands the event-triggered trace capability by adding conditions to the event trigger. Figure 3 (Figure 19 in SEMI E134) shows the new Condition construct in the DataCollectionPlan. Any EventTrigger can now have zero or more Conditions that must be satisfied before the Trigger can occur. Each Condition compares an equipment Parameter to a user-supplied value. If multiple Conditions are supplied, all must be satisfied (logical AND). A logical OR can be achieved by defining a separate Trigger for the same event, but with different Conditions.

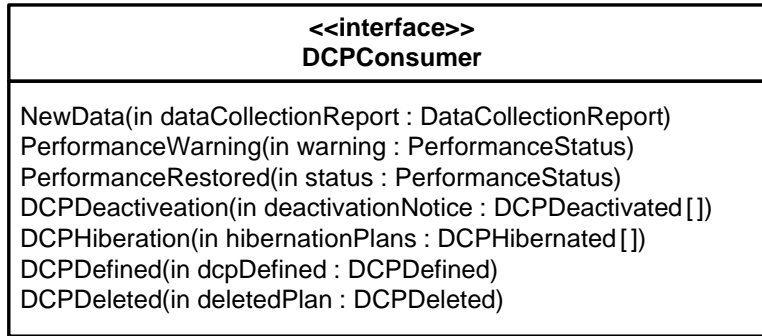


Reprinted with permission from Semiconductor Equipment and Materials International, Inc. (SEMI) © 2010. www.semi.org

**Figure 3 Data Collection Plan from SEMI E134**

### 5.1.3 DCPDefined/DCPDeleted Notifications

SEMI document #4729 added two new services to the DCPConsumer interface: DCPDefined() and DCPDeleted(). See Figure 4. These services allow the equipment to notify an EDA client when a different client adds or deletes DCPs to which this client has access.



Reprinted with permission from Semiconductor Equipment and Materials International, Inc. (SEMI) © 2010. www.semi.org

**Figure 4 DCPConsumer Interface from SEMI E134-0710**

#### 5.1.4 GetCurrentDateTime Service

SEMI document #4729 added the new service GetCurrentDateTime(), which returns the current date and time in the same ISO 8601 format as specified for SEMI E134 timestamps. This allows the client to determine the difference in equipment time and client time. Note that the accuracy of this approach is limited due to the communication delay, but will show where significant offsets exist.

Future use of time synchronization techniques such as NTP will eliminate most significant time offsets.

#### 5.1.5 Timestamp for Ad Hoc Data Request

SEMI document #4729 modified the GetParameterValues to add a “reportTime” field. This new field records the time at which a response to the ad hoc data request was created.

Including the timestamp in the response enables the client to better compare/merge this ad hoc data with other data that has been sampled at about the same time (e.g., trace data or event data).

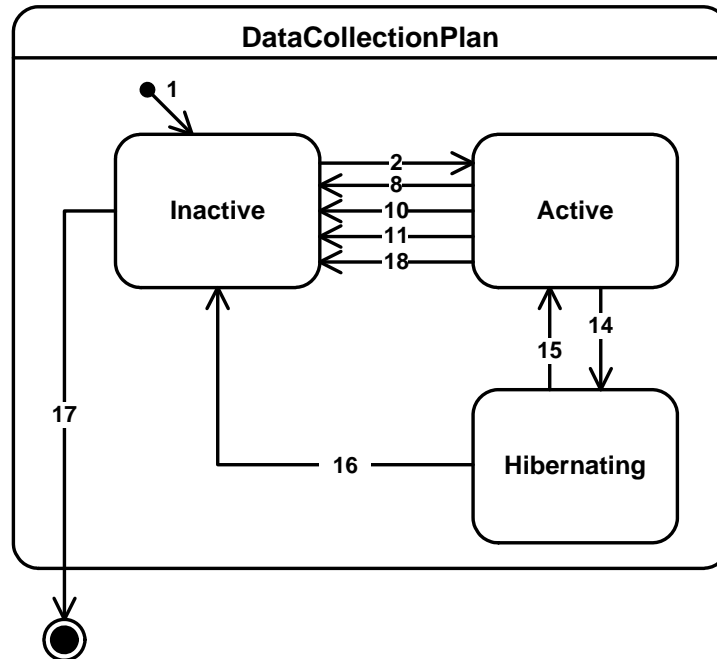
#### 5.1.6 DCP StateMachine

SEMI document #4729 changed the application of the DCP state model (see Figure 5).

Previously, a single instance of the state model represented the DCP’s state for all clients. This added significant complexity to the model and yet left some cases in which the model did not yield reasonable results. For example, if a DCP with a Trace had run to completion for a first client, then when a second client activated the DCP, no Trace data would be returned.

The new approach allows the DCP state model to be applied independently for each client of a DCP, thus assuring the client that the DCP will provide all expected data upon activation.

This change also simplified the DCP state model. Eight of the original 17 transitions were eliminated.



Reprinted with permission from Semiconductor Equipment and Materials International, Inc. (SEMI) © 2010. www.semi.org

**Figure 5 DCP State Model from SEMI E134-0710**

The DCP state model was modified to add one new transition from the Active to Inactive state. This transition is taken when “the equipment has determined that the DCP is invalid” and at the equipment’s discretion, which may be not at all. The change does not dictate when or if the equipment should check whether the DCP is invalid. It also does not give any criteria for declaring the DCP invalid. However, if a DCP no longer contains any valid parameters, the equipment now has the option of deactivating this DCP.

### 5.1.7 Description and Name Attributes

The “name” and “description” attributes have been added by SEMI document #4729 to certain messages to make interpretation more efficient.

The name and description attributes were added to the GetDefinedPlanIds() and GetActivePlanIds() responses to help the client understand the nature of the DCPs listed. Note that some DCPs in the list may not have been created by this client. This includes pre-defined (supplier-defined) DCPs and those created by other clients.

The old version of the service returned the planID, but no descriptive information. The client was forced to upload each unknown DCP to understand its purpose. Receiving name and description may eliminate the need to upload and examine each DCP.

Because it shares a data structure with GetActivePlanIds(), the ActivatePlan() response was also affected. It now contains the name and description of the DCP that was activated.

The Exception description was added to the ExceptionReport structure. This makes the exception report more immediately understandable to a user. It is possible to get that information from the metadata using the exceptionID, but this assumes that the consumer of the exception message has instant access to the metadata.

### 5.1.8 Data Types

To promote the reuse of many of the data types and the tools for defining these data types, this portion was moved from SEMI E125 and SEMI E134 to SEMI E138 (XML Semiconductor Common Components). SEMI E134 references SEMI E138 for the data types.

The data types were improved and streamlined. See Section 9 for more detail.

### 5.1.9 Privileges

SEMI document #4823, line item #4, corrects the ManageOnlyAuthoredDCPs privilege as it applies to the GetCurrentPerformanceStatus operation. A client with this privilege will receive DCPActivated information only for plans activated by that client. It will not receive information related to other clients, even for DCPs that client has activated. Previously, all active DCPs for all clients were returned.

## 5.2 Clarifications and Editorial Changes

### 5.2.1 Provisional Status Removed

SEMI E134 was elevated from a provisional specification to a full SEMI specification by SEMI document #4362, line item #1. This reflects the acknowledged maturity of the standard.

### 5.2.2 DeactivatePlan Behavior

The behavior of the DeactivatePlan() request is covered more completely. Several cases are based on combinations of the following:

- Whether planID = a single ID or “allDCPs”
- Whether terminate = true or false
- Whether the specified DCP was active or not (or whether any were active in the case of allDCPs)
- Whether the requesting client has the ManageAnyDCP privilege

Behavior for each case is clarified.

### 5.2.3 TimeActivated

SEMI document #4823, line item #3, clarifies that the DCP “timeActivated” value is not affected by transition to/from the Hibernating state (e.g., when the equipment is shut down). The purpose of timeActivated is to record the time at which the DCP was first activated for evaluation in the historical context of that time. It is not guaranteed, or intended, to accurately represent the total time the equipment has been actively sending the data reports.

### 5.2.4 NoValue and Null Usage

The implementer can choose whether to use NoValue or a null value as needed. Note that because XML specifies how a null value is represented, it is not given a separate class. The definition of NoValue is now part of SEMI E138.

## **6 SEMI E120 COMMON EQUIPMENT MODEL**

For more detail, see Appendix D.

### **6.1 Conceptual Changes**

#### **6.1.1 MaterialLocations for Consumables and Others**

The MaterialLocation class was expanded to cover consumable and other types of locations in SEMI document #4160A, line item #1. This was done by adding the new values “Consumable” and “Other” to the materialType enumeration definition.

In addition, a new materialSubType attribute was added to the MaterialLocation class. This is a free form text field that allows the implementer to give more specific information. For instance a Carrier type of location might have a materialSubType = “FOUP.” Substrate location might have a materialSubType = “300mm wafer.”

#### **6.1.2 Nameable uid Value**

SEMI document #4160A, line item #3 removed the requirement that the Nameable uid attribute always contain a properly generated ISO/IEC 11578:1996 uuid value. Suppliers reported that creating and maintaining these universally unique values is too difficult. This is especially true in the field as parts are periodically replaced. The uid of the Equipment class still must be a valid uuid value. All other elements of the equipment structure no longer require a uuid.

The change to SEMI E120 still allows the uuid value to be used. When a uuid is assigned, the uuid form is used (e.g., 9F5FBC24-EFE2-4f90-B498-EC0FB7D47D15). However, if a properly generated uuid value is not used for the uid, then the free form string must not take the uuid format. In this way, the client can identify which is provided.

#### **6.1.3 EquipmentElement**

The EquipmentElement has an attributed called elementType. This attribute was intended to identify the type of each equipment component. The attribute is a free form string format. User feedback requested a more limited set of choices for elementType that could be recognized in an automated fashion.

SEMI document #4597, line item #2, establishes a set of predefined strings to be used for elementType. The set includes values from many common instances of Equipment, Modules, and Subsystems.

It was not possible to create an exhaustive set of values. Retaining the string format for elementType allows the implementer to extend the list by creating additional elementType values, but only if the component does not fit any of the predefined values.

A second change to EquipmentElement was made in SEMI document #4597, line item #3. The attributes supplier, make, model, modelRevision, function, and immutableId were changed from required to conditional. They may be excluded when not needed. According to the new text,

“Each attribute is required when a value corresponding to that attribute has been associated with the EquipmentElement. For example, immutableId is a required attribute when the supplier has assigned a serial number (or equivalent) to that component of the equipment.”

The older version required all attributes be included in the metadata, but contain “unknown” when no value is available.

#### **6.1.4 LogicalElement**

SEMI document #4776 added the new construct “LogicalElement” to the Common Equipment Model. This change was made to provide an approved method to model non-structural items that are static. For example, SEMI E116 defines “EPTTracker” objects that track structural elements within the equipment. These EPTTrackers are always present. Previously, the conformant approach was to model these as SEMI Objects. However, the ObjTypes were optimized for dynamic objects, such as Substrates, that appear and disappear from the equipment over time. Some implementations defined EPTTrackers as Subsystems. However, this approach confuses concrete elements of the equipment with software constructs. The LogicalElement is expected to be used to model EPTTracker, MaterialManager (to hold material-related objects and parameters), and JobManager (to hold job-related objects and parameters).

### **6.2 Clarifications and Editorial Changes**

#### **6.2.1 AbstractModule**

The AbstractModule class was renamed to ExecutionElement by SEMI document #4081, line item #2 to help implementers better understand its purpose. Its definition was expanded from “capable of processing material” to include measuring and testing of material as well.

#### **6.2.2 Locator**

The Locator definition and XML data type are used by SEMI E134. However, the Locator was defined in a Related Information section of E120. Thus, it could not be considered a required element. In SEMI document #4081, line item #5, the Locator definition was moved to the main part of the specification.

#### **6.2.3 Regular Expressions**

The formats of the LocatorType and the NameType are controlled by regular expressions in the XML schema file. Because they were found to be faulty, they were corrected in SEMI document #4729.

## **7 SEMI E132 EQUIPMENT CLIENT AUTHENTICATION AND AUTHORIZATION**

For more detail, see Appendix E.

### **7.1 Conceptual Changes**

#### **7.1.1 EnhancedEstablishSession**

As other new standards (e.g., E139 and E142) began to adopt EDA-style messaging based on SEMI E132, it was clear that an EDA client might gain unwanted services (e.g., Recipe Management) when it connects to the equipment to collect data. SEMI document #4409 created a way to request specific services for a session, while excluding others.

The new EnhancedEstablishSession() service allows the requestor to choose the services for the session. The naming of capabilities is specified and includes both the standard number and version. Thus, equipment could, in theory, offer connection to either E134-1105 or E134-0710 on the same equipment.

For backward compatibility, the original EstablishSession() service has been retained.

### **7.1.2 InterfaceDiscovery Interface**

The initial connection from an EDA client to equipment requires entering multiple long internet address strings (URNs), usually by hand, along with some other information. Data entry errors in the URNs are difficult to diagnose. An incorrectly addressed message never arrives at its intended destination. This might occur for several reasons, but no error message indicates the reason.

SEMI document #4533, line item #1 adds the new InterfaceDiscovery interface. This interface will respond to queries with all the information needed to establish a session with a piece of equipment. The returned information includes the list of capabilities that can be specified with the EnhancedEstablishCommunication() service.

To connect, a potential client need know only the Ethernet address or name of the equipment's EDA computer.

## **7.2 All Sessions Persistent**

Session persistence was an added feature to EDA before the 1105 freeze. Subsequently, it was decided that all sessions should be persistent by default. SEMI Document #4219 made this change in SEMI E132. It also deleted the PersistSession() service and removed the now unneeded "isPersistent" attribute from the ActiveSession class.

### **7.2.1 Disabling Session Pings**

The original SEMI E132 suggests a way to disable periodic pings by setting the ping interval to zero. SEMI document #4533, line item #5, made this the standard method for disabling session pings.

### **7.2.2 Number of Sessions per ACL Principal**

The SEMI E132 access control list (ACL) defines identities for the EDA clients called "principals." In the past, a client could connect to the equipment multiple times while identifying itself as the same principal for each connection. This can cause problems, since the connections look the same. SEMI document #4533, line item #6, limits a client to one active session per ACL principal.

### **7.2.3 SessionClosed Notification**

This line item removes the requirement for the equipment to send the SessionClosed notification to a client if that client initiated the session close.

## **7.3 Clarifications and Editorial Changes**

### **7.3.1 E128 Messaging Support**

SEMI E128 was originally created to define how asynchronous messaging should be done using SOAP. SEMI document #4034A added synchronous messaging support to E128. EDA uses synchronous messaging.

SEMI document #4409 changed SEMI E132 to remove the portion that duplicates the SEMI E128 definitions and to reference the E128. While some of the terminology changed (e.g., E32Header became SynchHeader), no conceptual change is involved.

### **7.3.2 GetDefinedPrivileges**

SEMI document #4533, line item #2, clarifies that when a client with “SecurityAdmin” rights requests a list of all defined privileges from the equipment, the GetDefinedPrivileges response should include all the defined privileges, including the “allPrivileges” and “SecurityAdmin” privileges.

## **8 SEMI E128 XML MESSAGE STRUCTURES**

For more detail, see Appendix F.

SEMI E128 was not originally part of the EDA standards set. Rather, it was created to support asynchronous messaging using SOAP. SEMI document #4034A added support for synchronous SOAP messaging as used for EDA. Changes were subsequently made to SEMI E125, E132, and E134 to remove the now duplicate definitions and reference those in SEMI E128.

## **9 SEMI E138 XML SEMICONDUCTOR COMMON COMPONENTS**

For more detail, see Appendix G.

The Error construct used by the 1105 EDA release is the same as that defined in SEMI E138. However, E138 was not included in the original EDA standards set because it was not ready in time. Subsequent changes to the EDA standards added references to SEMI E138 while removing the now duplicate definitions of the Error class.

### **9.1 Conceptual Changes**

SEMI document #4595A added the equivalent of SEMI E125 Parameter Types and SEMI E134 Parameter Values. These are intended to provide a common definition of those data format concepts that could be reused by other standards in the future.

Note that several detailed improvements to the definitions were made when transferring them to SEMI E138. See Appendix G for a list of the more notable of these improvements. While the main concepts have not changed, many of the details are affected. Implementers should study the new definitions closely.

### **9.2 Clarifications and Editorial Changes**

SEMI document #4729 corrects a fatal XML error that occurred with SEMI document #4595A changes.

## **10 SEMI E145 UNIT SYMBOLS IN XML**

For more detail, see Appendix H.

SEMI E145 did not exist at the time the 1105 version of EDA was released. SEMI E125-1105 included the Unit class, which had a text attributed called “symbol.” It did not place any limitations on the content of the symbol string.

SEMI E145 specifies symbols for most common base units and defines how to build more complex units from the base units. Some implementations of the 1105 EDA release conform to SEMI E145. The 0710 release requires it.

SEMI E145 has not been modified since its initial approval by SEMI.

## Appendix A – SEMI Ballots for EDA Since the 1105 Release

All SEMI ballots that were proposed between the 1105 and 0710 SEMI releases are listed in Table A-1, including those that were approved and those that failed.

The table displays the following columns:

- **Cycle** – During the year, SEMI provides seven voting cycles during which ballots can be proposed. In North America, three cycles are generally used: Cycle 1 (January–March), Cycle 4 (May–July), and Cycle 6 (August–October). This column contains text in the form YYYY-C, where YYYY is the year and C is the cycle number.
- **Doc#** – Each ballot submitted to SEMI is assigned a unique four-digit SEMI document number. If the document is submitted for balloting more than once, a letter is appended to the new submission. Therefore, SEMI document 4151 is labeled “4151” for its first submission, then “4151A” if it is resubmitted, then “4151B” for the third submittal, etc. The document number for a ballot that is approved is retired. If even one line item fails, the ballot may be resubmitted. A document that was completely approved with no changes is shown in **green**. A document that failed all line items is shown in **red**. When some line items failed or editorial changes were necessary, the document number is shown in **blue italics**.
- **Stds** – The standards affected by the ballot proposal are listed. When all proposals for a standard were approved with no changes, the standard is shown in **green**. When all proposed changes failed, it is shown in **red with strikeouts**. When some changes were approved or editorial changes were necessary, the standard is shown in **blue italics**.
- **Status** – This column shows whether the ballot proposal was approved or failed. Ballot proposals may be broken into Line Items, which are voted on separately. These are coded similarly to the other columns with **green for approved**, **red for failed**, and **blue italics for approved with editorial changes**.
- **Content** – Each ballot is briefly explained. Typically, this column shows the title of the line item from the ballot proposal.

Further details on these ballots are in the following appendices.

**Table A-1 EDA Ballots Since 1105 Release**

Cycle	Doc#	Stds	Status	Content
2005-1	4081	E120 E120.1	LI# 1 - Failed LI# 2 - 0705 LI# 3 - Failed LI# 4 - Failed LI# 5 – 0705	LI#1 Add Consumable Location LI#2 Rename AbstractModule LI#3 Remove uid Persistence Requirement LI#4 Relax uid Uniqueness Requirement LI#5 Locator Definition
2005-4	4160	<del>E420</del> <del>E420.4</del>	Failed	o Resubmit a corrected version of 4081 Line Item 1 - “Add a Consumable Location,” including the omitted schema change o Make changes to the equipment component identifiers “locator” and “uid” that have evolved from the original 4081 Line Items 3 & 4 o Update the E120.1 XML schema for CEM according to XML TF advice

**Table A-2 EDA Ballots Since 1105 Release (continued)**

2005-6	4151 A	E125 E125.1	LI# 1 - 0306 LI# 2 - 0306 LI# 3 - 0306 LI# 4 - 0306 LI# 5 - 0306	LI#1 Errata and Clarifications LI#2 Permit Transitions between Nested State Machines LI#3 Clarify Initial/Final states LI#4 Events without state machines LI#5 Simplify SEMI state naming convention
2005-6	4219	E132 E132.1	0306 with editorial changes	Removed support for non-persistent sessions
2005-7	3851 C	E145	0306 with editorial changes	New standard (Units)
2006-1	4034 A	E128	LI#1 - 0706 LI#2 - 0706 with editorial changes LI#3 - 0706 with editorial changes LI#4 - 0706	LI#1 Correct Namespace Description LI#2 Add Synchronous Header Elements  LI#3 Correct description and usage of Fault header  LI#4 Clarify usage of Action Header
2006-1	4160 A	E120 E120.1	LI#1 - 0706 LI#2 - 0706 LI#3 - 0706 LI#4 - Failed	LI#1 Changes to MaterialLocation class LI#2 Errata on published 0705 version LI#3 Relaxing requirement for "uid" attribute LI#4 Adding "Locators" attribute to Nameable Class
2006-4	3571 B	<del>E447</del>	Failed	EDA Guide
2006-6	3571 C	E147	0307 with editorial changes	EDA Guide
2006-6	4362	E134	LI#1 - 0307 with editorial changes LI#2 - 0307 LI#3 - 0307 LI#4 - 0307	LI#1 Removal of provisional status  LI#2 Correction of reference to SEMI E120 LI#3 Correction of Namespace references LI#4 Remove Redundant References to Dated Information
2006-6	4363	E132	0307	Remove Redundant References to Dated Information
2006-6	4366	E125	LI#1 - 0307 LI#2 - 0307 LI#3 - 0307 LI#4 - Failed LI#5 - 0307	LI#1 Removal of provisional status LI#2 Changes to NodeEvent and SEMIObjType Classes LI#3 Remove Redundant References to Dated Information LI#4 Transition Naming Convention Clarification LI#5 Schema file errata correction
2007-1	4409	E132	1107 with editorial changes	Adding E128 XML Messaging Support and the new EnhancedEstablishSession Request Service

**Table A-3 EDA Ballots Since 1105 Release (continued)**

2007-4	4452	E125 E125.1 <del>E134</del> <del>E134.1</del>	LI#1 - Failed LI#2 - Failed LI#3 - 1107 LI#4 - Failed LI#5 - Failed	LI#1 E125.1 - Correct File Names, Headers, and Versions for all Files LI#2 Changes to Correct Problem with Choice Construct LI#3 E125 - Editorial Correction LI#4 Correct File Names, Headers, and Versions for all E134.1 Files LI#5 Changes to Correct Problem with Choice Construct
2007-6	4452 A	E125 E125.1 E134 E134.1	LI#1 - 0308 LI#2 - 0308 with editorial changes	LI#1 E125.1 Correct File Names and Versions for all Files LI#2 Correct File Names and Versions for all E134.1 Files
2008-4	4533	E132 E132.1	LI#1 - 1108 LI#2 - 1108 LI#3 - 1108 with editorial changes LI#4 - 1108 LI#5 - 1108 LI#6 - 1108 with editorial changes LI#7 - 1108	LI#1 Interface Discovery LI#2 Equipment Defined Privilege Usage Clarification LI#3 Multiple Addition of Error Text and Error Codes LI#4 Editorial Corrections LI#5 Disabling Session Pings LI#6 Clarification for ACLEntry Session Allowance LI#7 CloseSession Notification Requirement
2008-4	4534	<del>E134</del> <del>E134.1</del>	LI#1 - Failed LI#2 - Failed LI#3 - Failed LI#4 - Failed	LI#1 DCPDefined and DCPDeleted operations added to DCPConsumer interface LI#2 Addition of index in CollectedData for TraceResults LI#3 Addition of timestamp in GetParameterValues LI#4 Changes to Clarify the Acceptance of Transient Parameters in Data Collection Plans
2008-4	4595	<del>E138</del> <del>E138.1</del>	Failed	Revision to SEMI E138, XML Semiconductor Common Components
2008-4	4597	E120 E120.1	LI#1 - Failed LI#2 - 1108 LI#3 - 1108 LI#4 - 1108	LI#1 Simplification of ExecutionElement attributes LI#2 Equipment component type attributes LI#3 Equipment component identity attributes LI#4 Regular Expression changes in the E120 schema
2008-6	4595 A	E138 E138.1	0309 with editorial changes	Revision to SEMI E138, XML Semiconductor Common Components
2009-2	4729	E125 E125.1 E134 E134.1 E138	0709 with editorial changes	Addresses many outstanding issues. Highlights: o Transient Parameters – added "Restricted" category o DCPDefined/DCPDeleted messages to privileged clients o Revised approach to multi-client Trace o Index attribute for Trace reports o Timestamp for ad hoc requests o New service: GetCurrentDateTime o Use of common formats from E138

**Table A-4 EDA Ballots Since 1105 Release (continued)**

2009-6	4596	E120.1 E125.1 E128 E132.1 E134.1	0310	Update EDA standards to SOAP Version 1.2, replacing SOAP 1.1
2009-6	4775	E120 E120.1	LI#1 - 0310 LI#2 - 0310	LI #1 Allow exclusion of ExecutionElement attributes LI #2 Clarify requirements when excluding EquipmentElement attributes
2009-6	4776	E120 E120.1 E125 E125.1	0310	Addition of LogicalElement to equipment structure to represent static non-structural objects.
2010-2	4822	E125 E125.1	LI#1 - 0710 LI#2 - 0710	LI #1 Mapping SECS Events/Alarms/Variables to EDA LI #2 StateMachine Default Entry Transitions (NoState)
2010-2	4823	E134 E134.1	LI#1 - 0710 LI#2 - 0710 LI#3 - 0710 LI#4 - 0710	LI #1 Condition-Based Trace Data Collection LI #2 Error Handling LI #3 Clarification of timeActivated LI #4 Privilege adjustment

## Appendix B – SEMI E125 Changes from E125-1105 to E125-0710

### SEMI E125-1105 was included in the original 1105 EDA freeze

Following are each E125-related ballot document, the resulting standard version, and a summary of the conceptual changes made by that ballot document. No technical details or a line-by-line compilation of the changes are included here. The ballot proposals are SEMI copyright documents and are not publicly available from SEMI after the voting period.

### Ballot Doc# 4151A → E125-0306 (LI#2-4 only, all schema/WSDL files)

- LI#1 – Errata and Clarifications
  - As the title implies, this corrected several conflicts – data item content, association cardinality, typographical errors, etc. There is no conceptual change in this line item.
- LI#2 – Permit Transitions between Nested State Machines
  - The title above is an unfortunate reuse of the term “Nested State Machines.” A “nestedStateMachine” is a StateMachineInstance contained within another StateMachineInstance. This line item adds the term “nested state machine” (note the spaces) to refer to a StateMachine that refers to another StateMachine. Do not be confused.
  - Within a Harel state model a state can contain substates. These substates may be “And” substates or “Or” substates. With “Or” substates, only one substate is occupied at any one time. With “And” substates, each substate is occupied in parallel with the others.
  - With E125 StateMachines, “Or” substates are modeled as a State containing other States. When a State contains “And” substates, it is modeled as the State containing StateMachines – one for each substate in the model for that State.
  - To move into a State with “And” substates, the transition must be to a State within a StateMachine contained within the target State. However, E125-1105 did not allow a transition from a State to a State within another StateMachine. This line item allowed that transition.
- LI#3 – Clarify Initial/Final States
  - This line item described and named the initial states (black dots on a Harel diagram) and final states (black dots with a circle around them in Harel) in the StateMachines.
  - Both initial and final states are to be described within a StateMachine with a single instance of the State object named “NoState.”
  - Transitions to/from “NoState” are treated like any other transition and are identified (numbered). Note that some related standards that must be represented in the metadata do not number these transitions (e.g., see the SEMI E30 Processing State Diagram).

- LI#4 – Events Without State Machines
  - This line item provided for a collection of “NodeEvents” (later called “SimpleEvents”) to be contained within the EquipmentNodeDescription. This allows the equipment to report events without defining the underlying StateMachine that initiates the events.
    - Note that this change allows even defined standard state models to be represented as simple events. There are some two-state models where this is appropriate.
- LI#5 – SEMI State Naming Convention
  - This line item simplified State names by removing the urn qualifier:
    - ~~“urn:semi-org:state:<standard id>:<state machine>:<state path name>”~~

**Ballot Doc# 4366 → E125-0307 (LI#1-3 & 5 only, all new schema/WSDL files)**

- LI#1 – Removal of provisional status
  - Simple removal of all references to “Provisional” for E125.1-0306. The marked sections were originally marked as provisional pending approval of E132.1 and E138.
- LI#2 – Changes to NodeEvent and SEMIObjType Classes
  - Adds SimpleEvents to SemiObjType class
    - Both E87 (Carrier Management) and E40 (Process Job Management) define non-state machine events as part of the standard.
  - Changed NodeEvent → SimpleEvent (see 4151A, LI#4)
    - The concept now applies to SEMI Objects as well as equipment nodes
  - SEMI Objects changed to allow multiple state models for a SEMI Object instance
- LI#3 – Remove Redundant References to Dated Information
  - This line item makes no conceptual changes.
  - It removes some unneeded references and converts other references to exclude version numbers that had to be changed any time the document they referred to was updated.
- **Failed** LI#4 – Transition Naming Convention Clarification
  - This line item attempted to require that all transitions be named by concatenating the “From” state and the “To” state names: <FromState>-<ToState>.
- LI#5 – Schema file errata correction
  - This line item made no conceptual changes. The schema was modified to better match the main document.

**Ballot Doc# 4452 → E125-1107 (LI#3 only)**

- **Failed** LI#1 – E125.1 – Correct File Names, Headers, and Versions for all Files
  - LI#1&2 reballoted and approved as 4452A LI#1
  - This line item attempted to reference updated versions of E120 and E132 and reflect the changes in them.

- **Failed** LI#2 – Changes to Correct Problem with Choice Construct
  - LI#1&2 reballoted and approved as 4452A LI#1
  - This line item addressed XML mapping issues in the schema and corrects the application of the choice construct.
  - There were no proposed conceptual changes in this line item.
- LI#3 – E125 – Editorial Correction
  - This line item made no conceptual changes. It is a minor editorial correction.
- **Failed** LI#4 – Correct File Names, Headers, and Versions for all E134.1 Files
  - **This was an E134.1 line item**
- **Failed** LI#5 – Changes to Correct Problem with Choice Construct
  - **This was an E134.1 line item**

**Ballot Doc# 4452A → E125-0308 (two line items, schema change)**

- Line item #1 – E125.1 Correct File Names and Versions for all Files
  - Re-ballot of 4452 LI#1&2 combined
    - This line item references updated versions of E120 and E132 and reflects the changes in them. This included movement of portions of E132 into E128.
    - This line item addresses XML mapping issues in the schema and corrects the application of the choice construct.
  - There were no proposed conceptual changes in this line item.
- Line item #2 – Correct File Names and Versions for all E134.1 Files
  - **This was an E134.1 line item**

**Ballot Doc# 4729 → E125-0709 (single line item, schema change)**

- Line item #1 – This was a large proposal that addressed several issues. Below is a summary of the most important of these.
  - Removed the data type definitions and replaced them with references to SEMI E138
    - Note that some improvements were made to the data type definitions. The primary change was the separation of data type and units.
  - Changed Exception severity to a required enumeration
    - A list of four (Fatal, Error, Warning, Informational) values are defined.
  - Changed isTransient Parameter attribute to enumeration with “Restricted” value
    - The enumeration has three choices – Transient, Restricted, and Unrestricted (the later two considered Non-transient). Restricted means “valid under certain conditions” to be documented by the implementer.

- Restricted Parameters are not always valid, but can be traced. For example, this is useful for a parameter that is valid only during processing (beam intensity) and is turned off at other times.
- Any alarms reported to the operator must be modeled as Exceptions
- Exception setData/clearData is replaced by exceptionData
  - The exceptionData field is a list of Parameter references. The setData and clearData constructs defined a unique Parameter for each Exception. This allows for reuse of existing parameters and less overall bulk.
- Simplified naming conventions (removing bulky urn text)
  - For example:  
urn:semi-org:transition:<standard id>:<state machine>:<transitionId>:<transition name>  
becomes simply <transition name>
- Required no overlap of SEMIObj attributes and eventData parameter lists
  - Misunderstanding has resulted in duplicate Parameters being defined in these two lists of Parameters within a SEMI Object. This clarifies that “attributes” should hold the object attributes (as defined in the referenced standard) and the “eventData” should hold Parameters that correspond to data items defined for reporting in events related to the object.
- MetadataRevised construct changed/optimized
  - The MetadataRevised contained redundant information (EquipmentElement) and unneeded structure (RevisionNotice class). Removing these items has simplified the construct.

#### **Ballot Doc# 4596 → E125-0310 (single line item)**

- Line item #1 – This ballot modified all of the XML (schema and wsdl) files to transition to support of SOAP Version 1.2 from SOAP 1.1. It made no changes to the application data contained in the EDA messages. Changes to the E125.1 document were limited to updating SOAP terminology and XML file version references.

#### **Ballot Doc# 4776 → E125-0310 (single line item)**

- Line item #1 – This ballot added a new construct "LogicalElement" to the Common Equipment Model (see the section on E120). The change to E125 is to the wording of a background section that references E120. There is no functional change to E125.

#### **Ballot Doc# 4822 → E125-0710 (2 line items, schema change)**

- Line item #1 – Adds a new feature to document the mapping of SECS data items, events, and alarms to their EDA equivalents. This touches several areas of the XML schema for E125:
  - SECS collection events (CEIDs) are mapped in the StateMachineInstance eventMaps and also in the SimpleEvent definitions. Note that StateMachineInstances appear in EquipmentNodeDescriptions and in SEMI Object definitions.

- SECS alarms (ALIDs) are mapped in the EquipmentNodeDescription exceptions references.
  - SECS collection events that correspond to EDA exceptions can also be mapped in the exception references.
- SECS variables (VIDs) are mapped in the Parameter definitions, which can appear in EquipmentNodeDescriptions and in SEMI Objects.
- Line item #2 – Clarifies representation of initial and final states. This line item affects the content of StateMachine definitions, but does not affect the base definitions and schemas. It makes clear that the initial and final state of a StateMachine is to be a single State named "NoState." In Harel statecharts, the default entry of a State (black dot) is not to be represented as "NoState." Such a transition is to be defined as coming from the containing state and going to one of its sub-states. If the default entry transition terminates in a "history" or "conditional" construct, this is to be represented in the metadata as a separate transition for each of the possible destination states.

## Appendix C – SEMI E134 Changes from E134-1105 to E134-0710

### SEMI E134-1105 was included in the original 1105 EDA freeze

Following are each E134-related ballot document, the resulting standard version, and a summary of the conceptual changes made by that ballot document. No technical details or a line-by-line compilation of the changes are included here. The ballot proposals are SEMI copyright documents and are not publicly available from SEMI after the voting period.

#### Ballot Doc#4362 → E134-0307 (four line items)

- LI#1 – Removal of provisional status
  - Simple removal of all references to “Provisional” for E134.1-1105. The marked sections were originally marked as provisional pending approval of E120.1, SEMI E125.1, and SEMI E132.1.
  - This line item makes no conceptual changes.
- LI#2 – Correction of reference to SEMI E120
  - Corrected the reference to the SEMI E120 Locator from Related Information to the main E120 document where it had been moved.
  - This line item makes no conceptual changes.
- LI#3 – Correction of Namespace references
  - Corrected references to point to the latest E120, E132, and E138
  - This line item makes no conceptual changes.
- LI#4 – Remove Redundant References to Dated Information
  - It removes some unneeded references and converts other references to exclude version numbers that had to be changed any time the document they referred to was updated.
  - This line item makes no conceptual changes.

#### Ballot Doc# 4452 → E134-1107 (LI#3 Only; **no approved changes to E134**)

- **Failed** LI#1 – E125.1 – Correct File Names, Headers, and Versions for all Files
  - This was an E125 line item
- **Failed** LI#2 – Changes to Correct Problem with Choice Construct
  - This was an E125 line item
- LI#3 – E125 – Editorial Correction
  - This was an E125 line item
- **Failed** LI#4 – Correct File Names, Headers, and Versions for all E134.1 Files
  - LI#4&5 reballoted and approved as 4452A LI#2.
  - This line item attempted to reference updated versions of E120 and E132 and reflect the changes in them.

- **Failed** LI#5 – Changes to Correct Problem with Choice Construct
  - LI#4&5 reballoted and approved as 4452A LI#2.
  - This line item attempted to address XML mapping issues in the schema and correct the application of the choice construct.
  - There were no proposed conceptual changes in this line item.

**Ballot Doc#4452A → E134-0308 (two line items, schema change)**

- Line item #1 – E125.1 Correct File Names and Versions for all Files
  - **This was an E125 line item**
- Line item #2 – Correct File Names and Versions for all E134.1 Files
  - Re-ballot of 4452 LI#4&5 combined
    - This line item referenced updated versions of E120 and E132 and reflects the changes in them. This included movement of portions of E132 into E128.
    - This line item addressed XML mapping issues in the schema and corrects the application of the choice construct.
  - There were no proposed conceptual changes in this line item.

**Ballot Doc# 4729 → E134-0709 (single line item, schema change)**

- Line Item #1 – This was a large proposal that addressed several issues. Below is a summary of the most important of these.
  - Clarified DeactivatePlan behavior in all cases
    - This DeactivatePlan message has four cases that are a combination of values for “Terminate” (true/false), planID (specific plan or “allDCPs”). These are addressed individually in the text.
  - Clarified use of NoValue and null.
    - NoValue and the reporting of null values are now specified in E138. It states that NoValue is for unanticipated situations, leaving null for “anticipated” times a parameter might be invalid, such as a module off-line for maintenance or a Restricted Parameter in its specified time to be invalid. In E134, this proposal allows the implementer to choose whether to use NoValue.
  - Clarified TraceReport cardinality in all situations.
    - The existing version of E134 said that a TraceReport must contain at least two CollectedData. This is true for most cases, but there are exceptions where one or none might be included. This was corrected to allow for these cases.
  - Added DCPDefined/DCPDeleted notifications
    - This new service notifies a client when a DCP is created or deleted by another client.

- Added new GetCurrentDateTime() service
  - This service returns the current date and time. It helps the client determine whether the equipment clock is accurate. This is a temporary substitute until factory-equipment time synchronization using NTP is broadly available.
- Added timestamp for ad hoc parameter requests
  - Adding a timestamp to ad hoc parameter requests allows for proper alignment when comparing the results with those received by event reports or trace.
- Added DCP description to GetDefinedPlanIds and DCPActivated
  - These messages reported on DCP's that might not be known to the client, including built-in DCP's and DCP's created by other clients. The description field helps understand the list of DCP's returned.
- Added TraceReport sequenceID
  - The sequenceID counts from 1 to n for each trace activation. This helps the client know if all trace reports have been received and which is the correct order.
- Added description to the ExceptionReport
  - Including the description allows the ExceptionReport to be understood more quickly by a human by avoiding having to access the metadata to retrieve it.
- Required a separate instance of the DCP StateMachine for each activating client of a DCP
  - The old approach used a single StateMachine instance to represent the DCP state for all clients. Some conflicts were found when multiple clients activated a DCP. It was possible for a client to activate the DCP, but receive no trace reports. This change makes the behavior more predictable.
- Added an optional transition for when equipment detects an invalid DCP
  - The new transition allows the equipment to deactivate a DCP if it finds it to be invalid. The decision is left to the equipment.
- Modified/clarified trace behavior for event-triggered start/stop
  - Several minor corrections are included. The most important is a change that says a stop trigger may be reported in a traceReport that contains no selected data, which is necessary in some normal cases.
- Removed the data type definitions and replaced them with references to SEMI E138
  - Note that some improvements were made to the data type definitions. The primary change was the separation of data type and units.
- ExceptionRequest attributes can be left out (instead of empty)
  - To eliminate some empty fields in messages, this change says that an unneeded field need not be included in the message. XML messaging accommodates this approach, where SECS does not in most cases.

**Ballot Doc# 4596 → E134-0310 (single line item, schema change)**

- Line item #1 – This ballot modified all of the XML (schema and wsdl) files to transition to support of SOAP Version 1.2 from SOAP 1.1. It made no changes to the application data contained in the EDA messages. Changes to the E134.1 document were limited to updating SOAP terminology and XML file version references.

**Ballot Doc# 4823 → E134-0710 (4 line items, schema change)**

- Line item #1 – Adds a new feature to the data trace capability. The user may now add one or more Conditions to the EventTrigger of a TraceRequest. Each Condition references a Parameter defined for the equipment and compares it to a user provided value. This allows finer control of event-triggered trace start and end.
- Line item #2 – Defines a new error response for "invalidGroupSizeForInterval." Some equipment can sample trace data values at a higher rate than it can send TraceReports. This requires a minimum groupSize be used when the trace interval is too small. The new error is used when the groupSize is too small, resulting in a TraceReport interval that is unsupportable.
- Line item #3 – Clarifies that the DCP "timeActivated" value is not affected by transition to/from the Hibernating state (e.g., when the equipment is shut down).
- Line item #4 – Corrects the ManageOnlyAuthoredDCPs privilege as it applies to the GetCurrentPerformanceStatus operation. A client with this privilege will receive DCPActivated information only for plans activated by that client. It will not receive information related to other clients, even for DCPs that the client has activated.

## Appendix D – SEMI E120 Changes From E120-1104 to E120-0310

### SEMI E120-1104 was included in the original 1105 EDA freeze

Following are each E120-related ballot document, the resulting standard version, and a summary of the conceptual changes made by that ballot document. No technical details or a line-by-line compilation of the changes are included here. The ballot proposals are SEMI copyright documents and are not publicly available from SEMI after the voting period.

### Ballot Doc# 4081 → E120-0705 (LI#2&5 only, schema change)

- **Failed** LI#1 – Add Consumable Location
  - Reballoted as 4160, which failed, and then was approved as 4160A, Line Item #1.
- LI#2 – Rename AbstractModule
  - Renames AbstractModule to ExecutionElement in E120.
  - Renames AbstractModuleType to ExecutionElementType in E120.1.
  - Changes the E120.1 schema file to rename AbstractModuleType to ExecutionElementType.
  - Expands definition of ExecutionElement from “capable of processing material” to “capable of processing, measuring, or testing material.”
- **Failed** LI#3 – Remove uid Persistence Requirement
  - Reballoted as 4160, which failed, and then was approved as 4160A, Line Item #3
- **Failed** LI#4 – Relax uid Uniqueness Requirement
  - Reballoted as 4160, which failed, and then was approved as 4160A, Line Item #3
- LI#5 – Locator Definition
  - Moved the Locator definition to the main body of the E120 standard to make it a formal definition. E134 referenced the Locator construct in E120 and needed a formal definition of Locator.
  - Defined “positional reference” as used in SEMI E134.

### **Failed** Ballot Doc# 4160 (remains E120-0705; one line item, schema change)

- The purpose of this ballot was to make the following changes to E120 and E120.1:
  - Resubmit a corrected version of 4081 Line Item #1 - “Add a Consumable Location,” including the omitted schema change.
  - Make changes to the equipment component identifiers “locator” and “uid” that have evolved from the original 4081 Line Items #3 & #4.
    - Proposed adding “locator” attribute to Nameable class.
    - Proposed moving uid attribute to EquipmentElement class (has never been reballoted).

- Update the E120.1 and the XML schema for CEM according to XML TF advice
  - Update the description for the XML diagram notation.
  - Restructuring of the E120 schema to remove containers and match XML TF recommendations.

**Ballot Doc# 4160A → E120-0706 (LI#1-3 only, schema change)**

- LI#1 – Add Consumable Location to MaterialLocation class
  - Expands the materialType enumeration to add “Consumable” and “Other.” Other is to be used only when none of the other choices apply.
  - Adds a new materialSubType, a free form text field so that, for example a Substrate materialType could be clarified with “300mm wafer” in the materialSubType.
  - The schema is updated with these changes.
- LI#2 – Errata list of references to figures and changes not included in the previous ballot
  - The changes are all editorial in nature.
  - Schema changes are included.
- LI#3 – Relaxation of the requirement for uniqueness of the uid value
  - This line item was first proposed as 4081, LI#3&4 and again as part of 4160.
  - Removes the requirement that the uid always contain a properly generated uuid value.
  - Requires uid to be a properly generated ISO/IEC 11578:1996 uuid value **if** a string of uuid form is used (for example: 9F5FBC24-EFE2-4f90-B498-EC0FB7D47D15); otherwise, requires uid to be a string that is unique within the scope of the equipment.
  - E120.1 and the XML schema are changed accordingly. Primarily, the UuidType is replaced with xs:string as the data type for the Uid type and any references using a uid value.
  - Some text was also added to clarify situations when an Equipment may contain another Equipment.
- **Failed** LI#4 – Addition of a “Locator” attribute to the Nameable class
  - Resubmitted from the 4160 ballot.
  - Proposed to add a "locators" attribute to the Nameable class.
  - Corresponding E120.1 and XML schema changes were also proposed.
  - This line item has never been resubmitted.

**Ballot Doc# 4597 → E120-1108 (LI#2-4 only, schema changes)**

- **Failed** LI#1 – Simplification of ExecutionElement attributes
  - Proposed the removal of the ExecutionElement attributes processName and processType.
  - This included changes to E120, E120.1, and the XML schema.

- This line item has not been resubmitted, but it may be resubmitted in the future.
- LI#2 – Equipment component type attributes
  - Clarifies the EquipmentElement elementType attribute definition
    - Adds pre-defined elementType string values for various instances of Equipment, Module, and Subsystem. These values are required to be used wherever their definitions match the nature of the modeled components.
    - This approach also allows for the addition of new elementTypes without changing the schema.
- LI#3 – Equipment component identity attributes
  - Modifies the “Reqd” field for attributes to include “C” for conditional. When conditional, the criteria for being required are defined in the specification.
  - The EquipmentElement attributes supplier, make, model, modelRevision, function, and immutableId are changed to Reqd = “C” for conditional.
    - New text says “Each attribute is required when a value corresponding to that attribute has been associated with the EquipmentElement. For example, immutableId is a required attribute when the supplier has assigned a serial number (or equivalent) to that component of the equipment.”
    - When not required, the attributes can now be left out entirely.
- LI#4 –Regular Expression changes in the E120 schema
  - The regular expressions in the “pattern” field for LocatorType and NameType give inconsistent results in Java and C#. These have been modified to simplify them and give correct and consistent results with these two commonly used programming languages.
  - The “pattern” that controls the contents of the LocatorType string is changed:
    - From: "([a-zA-Z])|([a-zA-Z][a-zA-Z\-. 0-9]\*[a-zA-Z\-.0-9])|([a-zA-Z])|([a-zA-Z][a-zA-Z\-.0-9]\*[a-zA-Z\-.0-9]))\*"
      - To: "[a-zA-Z]([a-zA-Z\-. 0-9]\*[a-zA-Z\-.0-9])?|([a-zA-Z]([a-zA-Z\-. 0-9]\*[a-zA-Z\-.0-9])?)\*/>
  - The “pattern” that controls the contents of the NameType string is changed:
    - From: "([a-zA-Z])|([a-zA-Z][a-zA-Z\-. 0-9]\*[a-zA-Z\-.0-9])"
    - To: "[a-zA-Z]([a-zA-Z\-. 0-9]\*[a-zA-Z\-.0-9])?"

**Ballot Doc# 4596 → E120-0310 (single line item) – See also 4775 and 4776**

- Line item #1 – Changes to SEMI E120 were unrelated to SOAP 1.2. An error in a namespace prefix was found in the E120 XML schema file that had to be corrected for the EDA schema set to be valid. This identical change was also made in ballot proposals 4775 and 4776 to ensure a viable schema set if any of these were approved (all were approved).

**Ballot Doc# 4775 → E120-0310 (single line item) – See also 4596 and 4776**

- Line item #1 – The line item changes the attributes of ExecutionElement from Required to Not Required. This includes processName, processType, and recipeType. This allows the implementer to leave these attributes out if they do not have meaningful contents. This eliminates many zero length strings and random or default values in these fields.
  - An error in a namespace prefix in the E120 XML schema file was also fixed (see the notes for 4596 for more detail).
- Line item #2 – In a previous ballot, four of the attributes for EquipmentElement (supplier, make, model, and modelRevision) were changed from Required to Conditional. However, text in the table where they were defined still required that these take the value "unknown" if no meaningful value is known. This ballot removed that text so they can be completely left off when not needed.

**Ballot Doc# 4776 → E120-0310 (single line item) – See also 4596 and 4775**

- Line item #1 – This ballot added a new construct "LogicalElement" to the Common Equipment Model. This allows the implementer to model non-structural objects and attach them to the equipment structure. For example, common LogicalElements may include JobManager (to hold job-related objects and parameters), MaterialManager (to hold material-related objects and parameters), and EPTTracker (from SEMI E116). Previously, implementers were modeling such constructs as SEMI Objects or as Equipment subsystems. This provides a well-defined solution for this type of object.
  - An error in a namespace prefix in the E120 XML schema file was also fixed (see the notes for 4596 for more detail).

## Appendix E – SEMI E132 Changes From E132-1105 to E132-0310

### SEMI E132-1105 was included in the original 1105 EDA freeze

Following are each E132-related ballot document, the resulting standard version, and a summary of the conceptual changes made by that ballot document. No technical details or a line-by-line compilation of the changes are included here. The ballot proposals are SEMI copyright documents and are not publicly available from SEMI after the voting period.

#### Ballot Doc# 4219 → E132-0306 (one line item, schema change)

- Removes non-persistent sessions
  - Removes the isPersistent attribute from the ActiveSession class
  - Deletes the PersistSession() service from the SessionManager interface
  - Changes affect E132, E132.1, and the XML schema
- Minor editorial corrections
- Changes most schema and wsdl file references in E132.1 to be more indirect. Thus, when the file names change (e.g., version information), the references do not also have to be updated.

#### Ballot Doc# 4363 → E132-0307 (one line item, schema change)

- Editorial changes only
  - Removes all instances of the binding target namespace name used in association with the SOAP Action.
  - Removes the names of the Port Type and Binding files included in each of the defined web service sections.

#### Ballot Doc# 4409 → E132-1107 (one line item, schema change)

- Note that this ballot passed committee review in April of 2007 and would ordinarily have become E132-0707. However, the patent issue with EDA delayed final approval of this document, pushing its release to 1107.
- Adds E128 XML Messaging Support based on E128-0706
  - SessionID attribute for ActiveSession now becomes an association to sessionID from E128.
  - The E132Header is replaced with the E128 xms:SyncHeader (Sync stands for “synchronous messaging”).
    - Various sections defining the old E132Header are removed.
  - This change affects E132, E132.1, and the schema and wsdl files.
- Adds the new EnhancedEstablishSession capability
  - Add the EnhancedEstablishSession service to the SessionManager interface.

- This is an alternate to the existing EstablishSession service.
  - This capability allows the requestor to connect to selected services available from the EDA service provider. For example, the client might choose to connect to Metadata and DataCollection services, but not to a non-EDA capability, such as RecipeManagement.
  - The naming of capabilities is specified and includes both the standard number and version. Thus, equipment could, in theory, offer connection to either E134-1105 or E134-0710 on the same equipment.
- Some editorial corrections are made in this ballot.

### **Ballot Doc# 4533 → E132-1108 (Line Items #1-7, schema change)**

- LI#1 – E132 and E132.1 – InterfaceDiscovery
  - A new InterfaceDiscovery interface is added to E132.
    - This interface allows a potential client to request the interface capabilities supported by the equipment. The requestor needs to know only the computer name or Ethernet address of the equipment. This eliminates the need for error-prone configuration of the client based on the equipment specification.
    - The new service is named “GetInterfaces().”
- LI#2 – E132 – Equipment Defined Privilege Usage Clarification
  - This line item makes it clear that when a client with “SecurityAdmin” rights requests the equipment for a list of all defined privileges, the GetDefinedPrivileges response should include all the defined privileges, i.e., “allPrivileges” and “SecurityAdmin” privileges as well as those defined in another specification or by the implementer.
- LI#3 – E132 and E132.1 – Multiple Addition of Error Text and Error Codes
  - Line Item #3 was approved with editorial changes.
  - This line item addresses the proper error reporting for the following:
    - A client attempts to create a second Administrator Principal
    - A client attempts to delete a Role that is currently assigned to a Principal
    - A client attempts to establish a session when the equipment is shutting down
- LI#4 – E132 – Editorial Corrections
  - These changes are editorial in nature.
  - It includes correcting the data type of the Session Secret and the Client Id Proof of the Client Authenticate parameter definitions.
- LI#5 – E132 – Disabling Session Pings
  - SEMI E132 suggests a way to disable periodic pings by setting the ping interval to zero. This line item turns that suggestion into a standard method for disabling session pings.

- LI#6 – E132 – Clarification for ACLEntry Session
  - Line Item #6 was approved with editorial changes.
  - This line item limits a client to one active session per ACL Principal.
    - A client can still open multiple sessions, but each must be opened as a different Principal (that is, with a different identity).
- LI#7 – E132 – CloseSession Notification Requirement
  - This line item removes the requirement for the equipment to send the SessionClosed Notification to a client closing its own session.

**Ballot Doc# 4596 → E132-0310 (single line item)**

- Line item #1 – This ballot modified all of the XML (schema and wsdl) files to transition to support of SOAP Version 1.2 from SOAP 1.1. It made no changes to the application data contained in the EDA messages. Changes to the E132.1 document were limited to updating SOAP terminology and XML file version references.

## Appendix F – SEMI E128 Changes From Creation to E128-0310

### SEMI E128 was not part of the original 1105 EDA freeze

Note that E128 was originally created to define the XML structures required to encode message header or “envelope” information for asynchronous messages. The corresponding structures for synchronous messaging were originally defined within SEMI E132. The two definitions were merged in E128-0706 and, later, E132 was modified to remove the definition there and reference E128 instead.

Following are each E128-related ballot document, the resulting standard version, and a summary of the conceptual changes made by that ballot document. No technical details or a line-by-line compilation of the changes are included here. The ballot proposals are SEMI copyright documents and are not publicly available from SEMI after the voting period.

### Ballot Doc# 4034A → E128-0706 (Line Items #1-4, schema change)

- LI#1 – Correct Namespace Description
  - As the title suggests, this is a simple change of the XML namespace.
  - This is not a conceptual change.
- LI#2 – Add Synchronous Header Elements
  - Adds “SyncHeader” to replace the “E132Header” formerly in E132
    - Similarly, SynchHashHeader replaces E132HashHeader
  - Rephrases the old header construct as “AsynchHeader.” Note that E128 was previously focused on asynchronous messaging. EDA uses synchronous messaging – this is added in the 4034A ballot.
- LI#3 – Correct description and usage of Fault header
  - Changed mention of “SOAP Faults” to a more general discussion of Faults.
    - The original E128 specified SOAP Faults, but EDA has never used them.
- LI#4 – Clarify usage of Action Header
  - The line items adds explanatory text:
    - “In a simple web service binding, the Action value will be the operation name for the web service (i.e., ActivatePlan or GetSEMIObjType as defined in E134 and E125).”

### Ballot Doc# 4596 → E128-0310 (single line item)

- Line item #1 – This ballot modified all of the XML (schema and wsdl) files to transition to support of SOAP Version 1.2 from SOAP 1.1. E128 contains several XML message examples that had to be updated to conform to SOAP 1.2. Additional changes to the E128 document were limited to updating SOAP terminology and XML file version references.

## Appendix G – SEMI E138 Changes From Creation to E138-0709

### SEMI E138 was not part of the 1105 EDA freeze

SEMI E138 originally contained the definition of the “Error” construct used by EDA. The 1105 release of EDA did not directly reference SEMI E138. Each of the specifications (E125, E132, E134) contained its own copy of the common components schema. The EDA standards were later update to reference SEMI E138 directly.

More recently E138 was updated to include the data type/format definitions from SEMI E125 and E134. These specifications were subsequently updated to remove those definitions and reference them in E138.

Following are each E138-related ballot document, the resulting standard version, and a summary of the conceptual changes made by that ballot document. No technical details or a line-by-line compilation of the changes are included here. The ballot proposals are SEMI copyright documents and are not publicly available from SEMI after the voting period.

### **Failed** Ballot Doc# 4595 (remains E138-0305; one line item, schema change)

- This ballot proposed to add to SEMI E138 data format definitions equivalent to the Parameter value formats from SEMI E125 and from SEMI E134.
  - This was reballoted and approved as 4595A.

### Ballot Doc# 4595A → E138-0309 (one line item, schema change)

- This ballot adds to SEMI E138 a set of data format definitions equivalent to the Parameter types/formats from SEMI E125 and from SEMI E134.
  - The design of the formats changed in this process to correct some shortcomings and to align some of the differences in the two standards.
  - Note that this ballot and the subsequent changes to E125 and E134 decoupled units from the data type to provide a new flexibility and greater reusability of data types. This also reduces the total number of data types that must be defined.
  - Some of the changes to data formats include:
    - For the metadata, the simple data types are described using a single enumeration. Notice that for String type, the description no longer includes attributes for *language* and *maxCharacters*.
    - The VariableType, which was defined only in E125.1, is shown more clearly in E138.
    - The EnumerationType, which could contain either an EnumeratedInteger or EnumeratedString, is replaced with two separate types: EnumeratedInteger and EnumeratedString.
    - The ParameterValue is redesigned (and renamed to PV), adding more detailed simple data types and reflecting the other changes to E138.

**Ballot Doc# 4729 → E125-0709 (single line item)**

- Line Item #1 – This ballot corrects an XML error left in E138-0309. The details of that error are not included here. Some other minor issues are also addressed.
  - The Value Type (PV) was modified to reduce the number of times the E138 namespace would have to be repeated in data reports.
  - The “null” type was removed, because XML already handles null values.
  - The TypeDef class was removed to ensure that Structures use only defined types. This reduces the overall complexity of the metadata.

## **Appendix H – SEMI E145 Changes From Creation to E145-0306**

### **Not Part of the 1105 EDA freeze**

SEMI E145 was created to provide a common way to define units within the context of XML formatted communications. This specification was created after the 1105 version of EDA was released and thus was not a part of the EDA first freeze. Subsequent updates to the EDA standards added references to SEMI E145.

Following are each E145-related ballot document, the resulting standard version, and a summary of the conceptual changes made by that ballot document. No technical details or a line-by-line compilation of the changes are included here. The ballot proposals are SEMI copyright documents and are not publicly available from SEMI after the voting period.

### **Ballot Doc# 3851C → E145-0306 (new standard, includes schema file)**

- New specification was approved that contains the Unit object definition, references to common basic unit definitions and methods for building complex unit definitions from the basic units.



**International SEMATECH Manufacturing Initiative  
Technology Transfer  
2706 Montopolis Drive  
Austin, TX 78741**

**<http://ismi.sematech.org>  
e-mail: [info@sematech.org](mailto:info@sematech.org)**