SEMI E157-0211
SPECIFICATION FOR MODULE PROCESS TRACKING

This Standard was technically approved by the global Information & Control Committee. This edition was approved for publication by the global Audits & Reviews Subcommittee on December 21, 2010. Available at www.semiviews.org and www.semi.org in February 2011; originally published June 2010.

1 Purpose

1.1 The purpose of this specification is to define a standard equipment capability to report process-related data to the factory system. The subject of this specification is the activities of a processing location (i.e., process module) that are related to the execution of a recipe.

1.2 The collection of process data during recipe execution is important to today's semiconductor factories to support various applications that help optimize equipment processes, finished product quality, yield, and overall factory performance.

1.3 This process data reporting capability provides a factory's applications the means of linking sampled time series data (Traces) and summary process data to related context data. Standard events that document recipe execution in the process module guarantee the availability of such context information. The context data may include such items as identification of process job, equipment recipe, and affected substrates. The reporting capability provides a record of start and/or end of processing, with granularity to the recipe step level.

1.4 Reporting of these events must be process module-specific to support the reporting of process module-specific data parameters. Each process module will typically have a different set of context and process data parameter values. A general event, one that applies to all process chambers, would report the same data, independent of which process module the current event report represents. Such general events do not support reporting of different data sets for each process module, as needed to capture the unique data parameters available from each.

2 Scope

2.1 This standard applies to data collection related to the portion of equipment recipe execution that directly affects the processing locations (process modules) within production equipment.

2.2 This standard addresses:

- Equipment recipe execution as it directly affects process modules
- Process module state model for recipe execution
- Event reporting for process data collection
- Process context data parameters
- Equipment recipe components and recipe steps contained in the recipe components as they apply to process modules

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 addresses recipe execution only as it applies to equipment processing locations (process modules). This specification does not cover substrate sequencing among process chambers and substrate movement within the equipment.

3.2 The specification also does not cover processing-related computations that occur outside of the timeframe of activities that directly affect the process module.

3.2.1 For example, equipment might process a substrate in a process module and then move that substrate out of the process module, but continue computing the results of that process activity for later reporting. This specification does not specify the events for the eventual reporting of such data or for tracking the computation process.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E157-0211

4 Referenced Standards and Documents

4.1 *SEMI Standards*

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

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 E120 — Specification for the Common Equipment Model (CEM)

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

SEMI E134 — Specification for Data Collection Management (DCM)

SEMI E139 — Recipe and Parameter Management (RaP)

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

5.2 *Definitions*

5.2.1 *data parameter* — an item of data. In the context of this specification, a data parameter is a value or set of values that can be reported by the equipment to describe current status or describe an event that occurred on the equipment.

5.2.2 *equipment data acquisition (EDA)* — data collection interface based on SEMI E125 and SEMI E134, along with referenced standards including SEMI E120, SEMI E128, SEMI E132, SEMI E134 and SEMI E138.

5.2.3 *equipment* — the combination of hardware and software required to perform an operation or activity (e.g., processing, transporting, storing), including all direct auxiliary support or peripheral equipment (e.g., vacuum pumps, heat exchangers, effluent/exhaust treatment equipment).

5.2.4 *equipment recipe* — an executable specification of an activity or process on an equipment. The recipe is the user-managed, reusable portion of the set of instructions and settings that determine the processing environment seen by the material. Recipes may be subject to change between runs or processing cycles. An equipment recipe consists of one or more recipe components.

5.2.5 *host* — the intelligent system that communicates with the equipment.

5.2.6 *master recipe* — the recipe component that represents the set of recipe components that make up an equipment recipe. Through the master recipe, the other recipe components can be identified. It is the identifier of the master recipe that is supplied to the equipment for processing activities. For example, the RecID provided to a SEMI E40 ProcessJob is the master recipe's name or identifier. If the recipe consists of only one component, that component is the master recipe.

5.2.7 *multi-chamber equipment* — equipment that has more than one process module.

5.2.8 *process module* — a component of the equipment capable of processing material (i.e., adding manufacturing value). A process module is directed in its processing by equipment recipes, which are under the control of the factory user. Process chamber is another term for process module. The SEMI E120 Module class describes the same concept. A process module can contain other process modules (sub-modules).

5.2.9 *production equipment* — equipment used to produce semiconductor devices, including wafer sorting, process, and metrology equipment and excluding material handling equipment.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E157-0211

5.2.10 *recipe component* — an executable specification that is managed by the equipment as a separate entity (e.g., file). A recipe component represents all or part of an equipment recipe. For a multi-part recipe, a recipe component may be referred to as a sub-recipe. A recipe component may contain zero or more recipe steps.

5.2.11 *recipe parameter* — a formally defined variable (setting) defined within a recipe, permitting the actual value to be supplied externally.

5.2.12 *recipe step* — a distinct subset of the recipe's process specification that represents a significant change in the recipe execution conditions or instructions from the steps that precede or follow. A recipe step is completely contained within a single recipe component. Steps within a recipe component proceed one at a time.

5.2.13 *substrate* — the basic unit of material on which work is performed to create a product. Examples include wafers, lead frames, CD's, die, glass panels, circuit boards, and disks.

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.

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

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

- [/RQ]

6.1.2 Requirement IDs apply across the row (left to right) of a table unless otherwise specified. No "[/RQ]" is used to mark the end of the requirements text in this case.

6.1.3 Only text marked with the requirement identifier is a requirement of this specification. Clarification, examples, and related recommendations may be provided near a requirement, but are not part of the requirement.

Table 1 Requirement Identifiers

Format Notation	*Purpose*
Vsss.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.

6.2 *State Model Methodology*

6.2.1 This document uses the Harel state chart convention for describing dynamic operation of defined objects. The outline of this convention is described in an attachment of SEMI E30. The official definition of this convention is described in "State Charts: A Visual Formalism for Complex Systems" written by D. Harel in Science of Computer Programming 8, 1987.

6.2.2 The Harel convention does not have the concept of "creation" and "extinction" of state models for a temporary entity. In this document, a circle with a black circle inside is used for expressing extinction of an entity. A filled, black circle denotes the entry to the state model (the entity creation).

6.2.3 Transition tables are provided in conjunction with the state diagrams to describe explicitly the nature of each state transition. A transition table contains columns for Transition number, Previous State, Trigger, New State, Actions, and Comments. The "trigger" (column 3) for the transition occurs while in the "previous" state. The "actions" (column 5) includes a combination of:

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E157-0211

1) Actions taken upon exit of the previous state.

2) Actions taken upon entry of the new state.

3) Actions taken which are most closely associated with the transition.

6.2.3.1 No differentiation is made between these cases.

6.2.4 A state model consists of a state model diagram, state definitions, and a state transition table.

[E157.00-RQ-00001-00] All state transitions in this standard, unless otherwise specified, shall correspond to collection events. [/RQ]

6.2.5 A state model represents the host's view of the equipment, and does not necessarily describe the internal equipment operation.

[E157.00-RQ-00002-00] All state model transitions shall be mapped into the appropriate internal equipment events that satisfy the requirements of those transitions. [/RQ]

6.2.6 In certain implementations, the equipment may enter a state and have already satisfied all of the conditions required by the state models for transition to another state. In that case, the next state transition may occur immediately.

Num	*Previous State*	*Trigger*	*New State*	*Actions*	*Comments*

7 Implementation Overview

7.1 This section provides an overview of the module process tracking capability. This overview refers to requirements defined later in the document. However, no statement in this overview section defines a requirement. This overview is intended to provide a general understanding of the module process tracking concepts so that the detailed requirements that follow can be more easily understood.

7.1.1 Module process tracking capability provides data collection functionality for tracking the portion of recipe execution that directly affects the process module. This includes necessary context information and crucial data parameters related to that recipe execution.

7.1.2 The specification does not in any way imply or restrict a specific use of the data collected.

7.2 The implementation of recipe execution varies across equipment types. However, the following concepts can be found across many platforms. Figure 1 provides an example illustration of some of these concepts in the form of a class diagram.

7.2.1 This example is intended to be conceptual only and self-describing, so additional definitions of the classes shown are not provided. It recognizes that in the context of ProcessJob, the equipment executes the job at different levels. The highest level, the "MasterProcess" may be instructed by a "MasterRecipe". A subordinate "SubProcess" might get its instructions from a separate "RecipeComponent". The execution that affects a "ProcessModule" might be divided into "RecipeSteps". Recipe parameters (RcpParamValues) may affect all levels of processing and can come from any of the RecipeComponents.

7.2.2 The Module Process State Model reflects the effects of recipe execution on a ProcessModule. It provides events for reporting data related to process execution within a process module. It allows for reporting down to the recipe step level.

7.2.3 The classes in Figure 1 represented in bold lines and text are described in more detail within this specification.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E157-0211

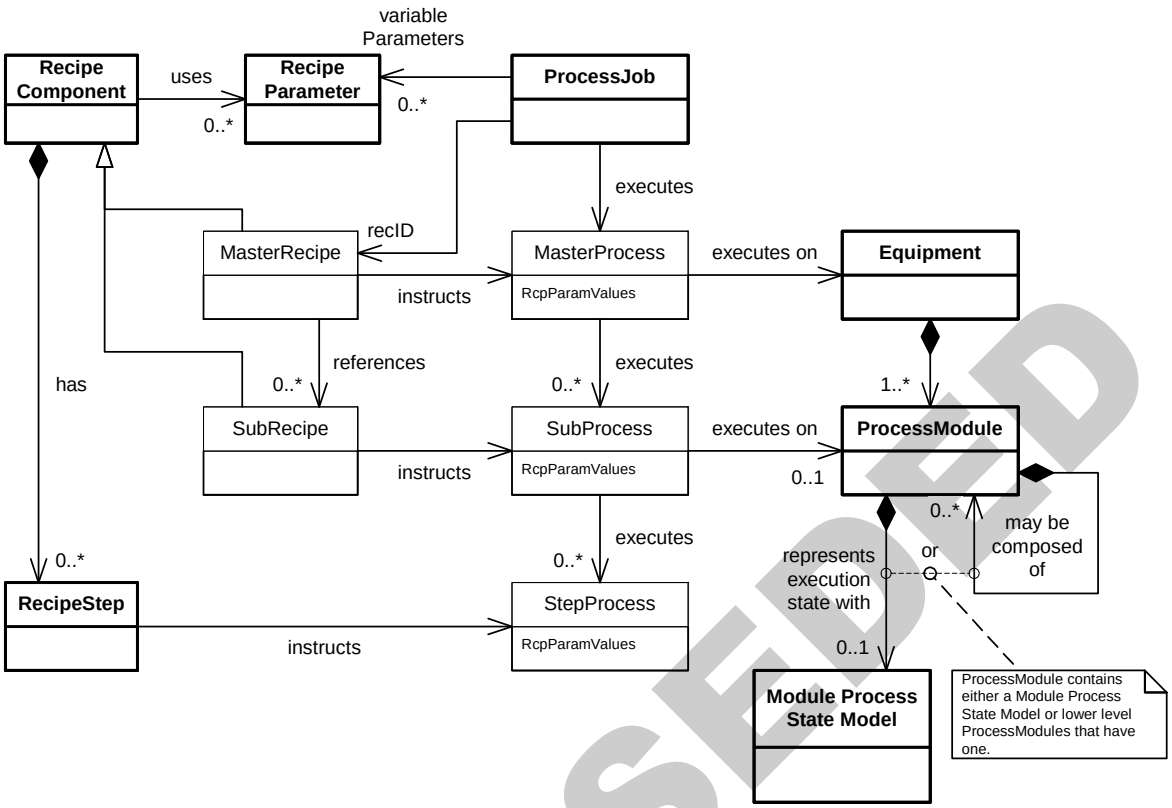

Figure 1
Example Illustration of Recipe Execution Concepts

SUPERSEDED

7.3 *Recipes*[1]

7.3.1 When equipment is instructed to process zero or more substrates, the name of the main or "master" recipe is typically referenced in the setup for the processing. For example, a SEMI E40 ProcessJob contains the RecID.

7.3.2 This master recipe can reference multiple sub-recipes. The master recipe and all sub-recipes are individual recipe components. Together, these recipe components make up the equipment recipe.

7.3.3 Some or all of the recipe components contain the instructions and settings for processing within each process module. The "module process" is the activity within a process module executed according to an equipment recipe. The module process does not include any pre-process and post-process activities that do not affect the process module directly. For example, calculation activity that occurs after all affects on the process module conclude is not part of the module process.

7.3.4 The beginning and end of module process activity do not necessarily correspond to the beginning and end of execution of a particular recipe component.

7.3.5 Where substrates are included in recipe execution, the processing instructions from the equipment recipes are executed separately on each substrate or substrate group. Activities for different substrates within an equipment typically occur in parallel. One substrate may be processing in Chamber A, while another is processing in Chamber B and a third is in transit back to the carrier.

7.3.6 Tracking of the recipe execution affecting a process module is independent of how the execution is initiated. Recipe execution may be initiated remotely, such as through a SEMI E40 ProcessJob. It may also be initiated at the

[1] Note that this specification and the recipe identifiers mentioned within are independent of the style of recipes (or process programs) supported and their corresponding naming conventions. See SEMI E30, SEMI E40 and SEMI E139 for more information.

Documents Document Sharing Hub
Full document details: SEMI E157-0211

user interface. In some cases, recipe execution may be triggered by the equipment itself. For example, maintenance activities might be triggered on a timed basis.

7.3.7 Process execution within a process module normally consists of a variety of activities happening sequentially and/or in parallel. From the standpoint of the object of the process execution (typically a substrate), this can be described as a linear sequence of conditions (or combinations of conditions) encountered.

7.3.7.1 Where a change of conditions is deemed significant to the user, a recipe step boundary should be defined. This provides for reporting of this significant change.

7.3.7.2 Reporting of step-related events can provide important information to the host. The Module Process State Model recognizes recipe steps that begin and end within the scope of module processing and provides transitions to report their start and end.

7.3.7.3 When steps are reported, they must be represented as occurring one at a time. The sequencing of the steps is not restricted. It may be simple or contain loops and branches.

7.3.7.4 It is left to the implementer to define the appropriate recipe steps to be reported (if any).

7.3.7.5 The granularity of the steps defined is an implementation decision and may not represent all changes to recipe execution conditions.

7.3.8 Each module process activity is tracked and reported separately. Recipe steps are recognized and reported if they are defined by the supplier.

7.4 *Recipe Parameters*

7.4.1 The recipe specification can be modified by the use of recipe parameters. Recipe parameter values may be supplied from several sources.

7.4.2 They may come from the factory host, for example as ProcessJob Variable Parameters. However, the recipe parameter values used by a recipe component or recipe step may not correspond to the Variable Parameter values supplied during job setup.

7.4.2.1 Recipe parameters may be passed from higher-level recipes to sub-recipes. For example, a master recipe may contain a fixed value for the process time, which is passed as a recipe parameter to a recipe component at run time.

7.4.2.2 A recipe parameter may also be supplied by the equipment based on an internal tuning algorithm performed during recipe execution (e.g., wafer-to-wafer control).

7.4.3 The source of the recipe parameters is not the topic of this specification. However, reporting of the values used for recipe parameters is addressed in this specification.

7.5 *Module Process State Model*

7.5.1 The primary new concept introduced in this standard is the Module Process State Model, shown in Figure 2 below. This state model defines behavior and events that are common to all process modules. The result is uniform reporting of key processing events across most or all equipment types. Details of the state model are defined in ¶ 9.1.

7.5.2 Each process module in the equipment that executes recipe instructions will have its own instance of the Module Process State Model.

7.5.3 This specification does not prohibit the equipment from providing an additional state model that is equipment specific and provides more detail. Such a state model can co-exist with the Module Process State Model.

7.6 The Module Process State Model provides a means of reporting meaningful events and data related to recipe–driven processing activities within a process module. This is not a state model of the equipment recipe or of any recipe component. Instead, it represents the effects of recipe execution on the process module.

7.7 *Process Modules*

7.7.1 An important aspect of implementing Module Process Tracking is the designation of process modules and the assignment of state model instances.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E157-0211