Service	*SECS-II Field*	*Values*	*Req.*	*Description*
	$CPVAL_4$	L, c 1. L, 3 1. U4 2. U4 3. U4 … c. L, 3 1. U4 2. U4 3. U4	N	Specifies datalog options for specific test ids. L, c (c = # of (TestId, TestStatus) pairs being assigned reports.) 1. L, 3 1. $TestID_1$ 2. $TestStatus_1$ 3. $ReportID_1$ … c. L, 3 1. $TestID_c$ 2. $TestStatus_c$ 3. $ReportID_c$
	$CPNAME_5$	"DATALOGOPTION"	N	DATALOGOPTION parameter.
	$CPVAL_5$	U4	N	Turns datalog on/off. (0:No Log, 1:Log Pass, 2:Log Failures, 3: Log All)
	$CPNAME_6$	"DATALOGPLANNAME"	N	DATALOGPLANNAME parameter.
	$CPVAL_6$	A[80]	N	Turns on datalogging according to a predefined plan. This parameter overrides all other individual datalog setting options in a START command.
START-EXEC	RCMD	"START-EXEC"	Y	
STOP	RCMD	"STOP"	Y	
STOP-EXEC	RCMD	"STOP-EXEC"	Y	

6 Variable Data Item Mapping

6.1 The purpose of this section is to define the list of variable items required by the TSEM. Values of these variables will be available to the host through collection event reports and host status queries.

6.2 *Requirements*

6.2.1 All generic variable items defined in SEMI E30 are required by all TSEM equipment.

6.2.2 Any supplier-defined variables shall be documented in the same format used by this document. The following minimum information is required:

<variable name> **Class:** <ECV, SV, or DVVAL> **Format:** <SML>

Description: <If class = DVVAL, description must contain statement of when data is valid.>

<If format = ASCII, then a length is required. It is assumed to be left-justified unless otherwise noted.>

6.3 *Data Types*

6.3.1 Equipment Constants (ECVs) can be changed by the host using S2,F15. The operator may be able to change some values, but the equipment does not change the values on its own. The value of an equipment constant may be queried by the host at any time, using the S2,F13/14 transaction. They reside in non-volatile memory of the equipment. Equipment constants remain in effect until they are overwritten either by manual entry or by a S2,F15 (NEW EQUIPMENT CONSTANT SEND).

6.3.2 Equipment constants have various uses in TSEM, including the following:

- Equipment offsets that match the performance of several pieces of equipment that would otherwise perform differently due to inherent manufacturing differences. Examples are home values and motion axis scaling factors.

- Setting the configuration of the equipment to allow for different material specifications, equipment options, material flows, frequency of automatic functions, etc. An example is yield check frequency.
- Managing optional machine features. Examples are constants that indicate whether optional features such as automated media stackers are present and control the configuration and function of these optional subsystems when they are present.

6.3.3 Status Variables (SVs) are valid at all times. An SV may not be changed by the host but may be changed by the equipment or operator. The value of status variables may be queried by the host at anytime using the S1,F3/4 or S6,F19/20 transactions.

6.3.4 DVVALs are variables that are valid only upon the occurrence of specific collection events. An attempt to read a variable item at the wrong time does not generate an error, but the data reported may not have relevant meaning.

6.3.5 *Data Item Requirements for Multi-Head, Multi-Site Equipment* — The identification for multi-head and multi-site data (variable items, status variables, events, etc.) is addressed in this specification through the use of status variables. In Table 5, the subscript "v" is used to denote the number of virtual testers, "h" is used to denote the number of tester heads, "s" to denote the number of tester head sites, and "b" to denote the number of bins or classes.

6.3.6 *class, hard-bin, and soft-bin* — Equipment is to maintain DVVALs which provide three levels of granularity for test results: class, hard-bin, and soft-bin. Classes, hard-bins, and soft-bins are expected to be defined within a process program, and their names are made available as DVVALs. When device testing has completed, the process program is to determine the class, hard-bin, and soft-bin with which the device is to be associated, based on the results of the testing. Finally, a summation of the number of devices associated with each class, hard-bin, and soft-bin is also maintained throughout the execution of a process program, and these are also made available as DVVALs.

6.4 For multiple sites scenarios, if a variable is described as List by number of sites, then a list is expected, one value for each site.

Table 5 Variable Item Mapping Table

Variable Name	*Description*	*SECS-II Type*	*Class*	*Comments*
ActiveSites	List of test-sites that are to be tested.	L, n 1. U4 2. U4 : n. U4	SV	n = # of active sites
Address	Address of the reported vector.	U4	DVVAL	Valid only in Datalog reports.
Channel	Hardware channel or resource identifier.	A[80]	DVVAL	List by PinID.
ClassID	Test result class number.	U4	DVVAL	List by test-site.
ClassName	Test result class name.	A[80]	DVVAL	List by ClassID.
ConfigInfo	Physical configuration information.	Format is determined by vendor.	SV	Implementation specific variable.
Cycle	Cycle count for this report.	U4	DVVAL	Valid only in Datalog reports.
DatalogConfig	Datalog Configuration.	A[80]	SV	
DatalogPlanName	Datalog report plan name.	A[80]	SV	
DockingStatus	Information on handler/prober docking status.	BOOLEAN	SV	

Variable Name	*Description*	*SECS-II Type*	*Class*	*Comments*
EnabledSites	List of test-sites initially enabled at process program download.	L, n 1. U4 2. U4 : n. U4	SV	n = # of enabled sites
EquipMake	Tool Manufacturer.	A[80]	SV	
EquipSerialID	Unique Equipment identifier.	A[80]	SV	
FunctionalResult	Vector of bits indicating pass or fail for each pin.	B	DVVAL	
HardBinID	Test result hard-bin number.	U4	DVVAL	List by test-site.
HardBinName	Test result hard-bin name.	A[80]	DVVAL	List by HardBbinID.
HighLimit	Higher limit for measurement. Multiple instances of this variable exist. See TEST-LIST report.	F4	DVVAL	List by TestID.
LowLimit	Lower limit for measurement. Multiple instances of this variable exist. See TEST-LIST report.	F4	DVVAL	List by TestID.
NumberOfHeads	The number of test-heads on the system.	U4	SV	
NumberOfPins	The number of pins for each test-head.	U4	SV	
OperatorID	Current Operator ID.	A[80]	ECV	
Pass	True if the test passed.	BOOLEAN	DVVAL	Valid only in Datalog reports.
PatternName	This must identify the location to which the vector address and cycle count are relative.	A[80]	DVVAL	Valid only in Datalog reports.
PinID	Pin identifier used to identify the pin results in pin result reports.	U4	DVVAL	List by PinID.
PPExecVersion	Test program version.	A[80]	SV	
ProcessProgramID	Process program identifier.	A[80]	SV	
Range	Range used by the measurement unit.	F4	DVVAL	Valid only in Datalog reports.
RealResult	Measured value.	F4	DVVAL	Valid only in Datalog reports.
Signal	Signal name.	A[80]	DVVAL	List by PinID.
SiteID	Test-board test-site number for multi-site (parallel) testing.	U4	DVVAL	
SoftBinID	Test result soft-bin number.	U4	DVVAL	List by test-site.
SoftBinName	Test result soft-bin name.	A[80]	DVVAL	List by SoftBinID.
SoftwareBuildID	The build ID of this software list element. Multiple instances of this variable exist. See SOFTWARE-LIST report.	A[80]	DVVAL	
SoftwareName	The name of this software list element (e.g. Solaris, IG9000, etc.) Multiple instances of this variable exist. See SOFTWARE-LIST report.	A[80]	DVVAL	
SoftwareType	The type of this software list element (e.g. Operating System, Tool Control System, etc.) Multiple instances of this variable exist. See SOFTWARE-LIST report.	A[80]	DVVAL	

Variable Name	*Description*	*SECS-II Type*	*Class*	*Comments*
SoftwareVersion	The version of this software list element. Multiple instances of this variable exist. See SOFTWARE-LIST report.	A[80]	DVVAL	
StartTestPortID	Start Test Source (i.e. hand, keyboard, host).	A[80]	DVVAL	
TestBoardCalStatus	Test-board calibration status.	BOOLEAN	SV	
TestBoardID	ID of current test-board.	A[80]	SV	
TestBoardPosition	A relative location of a test-site on a test-board.	Size 2 array of U4	SV	X and Y position. Site 1 is 0,0.
TestBoardSiteID	A mapping of a test-site to a physical location on the tester.	L, 3 1. SiteID 2. TestBoardID 3. TestBoardPosition	SV	
TestBoardStatus	Test-board availability (1 = enabled, 0 = disabled).	U4	SV	List by test-site.
TestBoardType	Type of fixture (test-board, probecard, cable, contactor, etc.).	A[80]	SV	
TestHeadID	The ID of the test-head.	U4	SV	
TestHeadStatus	(2 = Not Available, 1 = enabled, 0 = disabled)	U4	SV	
TestID	Test identifier used to identify test results. Multiple instances of this variable exist. See TEST-LIST report.	U4	DVVAL	
TestInstance	Test instance identifier. The TestID and TestInstance combined should be unique for one execution of the test program. They can be the same for all results from units tested in parallel by the same instance of a single test (possibly different for a serially applied test in a multi-site test program).	U4	DVVAL	Valid only in Datalog reports.
TestName	Test Name. Multiple instances of this variable exist. See TEST-LIST report.	A[80]	DVVAL	List by TestID.
TestSetup	Setup conditions used for the test. Multiple instances of this variable exist. See TEST-LIST report.	A[80]	DVVAL	List by TestID.
TestStatus	Test status of a specific test in the test program: 1 = Pass 2 = Fail 3 = Both	U4	DVVAL	Valid only in Datalog reports.
TestTime	Test execution time in seconds.	F4	DVVAL	Valid only in Datalog reports.
TestType	1 = Parametric Test (real value result). 2 = Functional test (vector of pass/fail results). 4 = Text test (arbitrary text string as result). Multiple instances of this variable exist. See TEST-LIST report.	U4	DVVAL	List by TestID.

Variable Name	*Description*	*SECS-II Type*	*Class*	*Comments*
TextResult	Arbitrary string of text.	A[80]	DVVAL	Valid only in Datalog reports.
UnitID	Unit Serial Number.	A[80]	DVVAL	List by test-site.
Units	Electrical units to be measured. Multiple instances of this variable exist. See TEST-LIST report.	A[80]	DVVAL	List by TestID.
VirtualConfig	Current Virtual Configuration listing all test-sites used.	L, 2 1. TestBoardSiteID 2. TestHeadId	SV	List by test-site.

RELATED INFORMATION 1

NOTICE: This related information is not an official part of SEMI E122.1 and was derived from work by the originating committee. This related information was approved for publication by full letter ballot procedures.

R1-1 Scenarios

R1-1.1 The purpose of this section is to document possible TSEM-specific operational scenarios to provide further context to the specifications.

R1-1.2 *Normal Processing Scenario*

R1-1.2.1 This scenario shows typical automated processing of materials using a materials handler and TSEM equipment. This scenario assumes no direct link between Test Equipment and Handler. Host directs all run-to-run processing between the two equipments. Host and Station Controller represent the same entity here. Under this intended usage, job processing with TSEM is simplified from perspective of Host as shown in Figure R1-1.

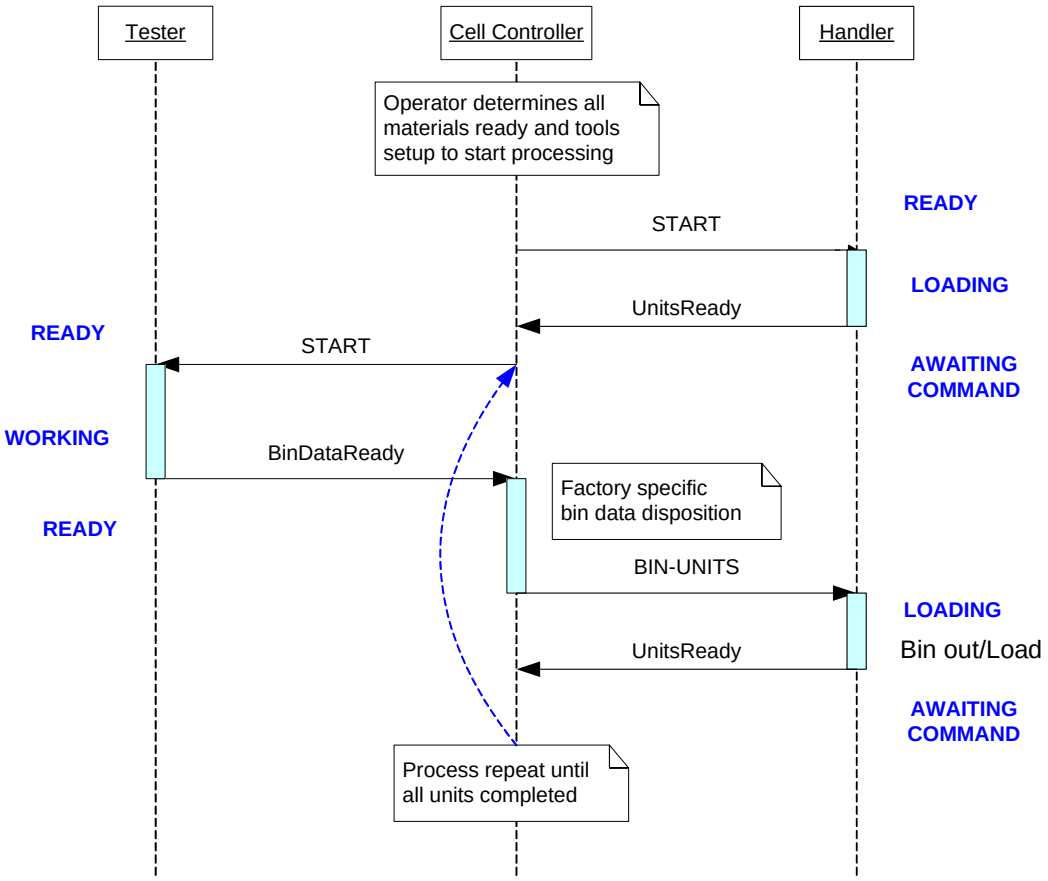

Figure R1-1
Normal Processing Scenario

R1-1.3 *Unbuffered Test Data Acquisition*

R1-1.3.1 This scenario shows acquisition of test results data from the test equipment while test is being performed on the device(s). This scenario is useful for online process analysis. Note that all test data for current device(s) under test needs to be read and acknowledged before next unit can proceed since data will be unbuffered.

Table R1-1 Unbuffered Test Data Acquisition Scenario

Host			*Test Equipment*	
SC introduces lot to equipment and other external systems.				
				Tester completes PP-SELECT and in READY state.
SC configures Test Data Collection plan on Tester and issues START command. See Test Result Data Collection section for details. *Assume Data Set A is assigned to all Testheads.*	S2,F49	⇨		
		⇦	S2,F50	
		⇦	S13,F1	*Tester starts Testing and as Test Result data becomes available, notifies SC of data set …*
DSSA ACK = 0	S13,F2	⇨		
Open Data Set Request (DSOR) L,2 1. HANDLE = H-A (arbitrary but unique) 2. DSNAME = A	S13,F3	⇨		
		⇦	S13,F4	Open Data Set Data (DSOD) L,5 1. HANDLE = H-A 2. DSNAME = A 3. ACKC13 = 0 4. TYPE = STREAM 5. RECLEN = 0 (N/A for streams)
Read Data Set Request (DSRR) L,2 1. HANDLE = H-A 2. READLN = 0 (read all available)	S13,F5	⇨		
SC processes incoming data … i.e. SC can spawn an agent to handle all incoming test data and route to DSS system, freeing core SC to process other events…		⇦	S13,F6	Read Data Set Data (DSRD) L,4 1. HANDLE = H-A 2. ACKC13 = 0 3. CKPNT =… 4. L,n 1. <FILDAT> 2. … n. <FILDAT>
Repeat until Test Completes …				
		⇦	S6,F11	Event Report Send (ERS) *Test Complete Event or State Transition Event associated with Test Complete, along with attached BinData Report.*
Event ACK	S6,F12	⇨		
Since test is complete, SC can now close the data set (is using agent, notify it to close upon read complete).				
Close Data Set Send (DSCS) L,1 1. HANDLE = H-A	S13,F7	⇨		
		⇦	S13,F8	

Perform dispositioning…				
Repeat until Lot is Complete …				

R1-1.4 *Buffered Test Data Acquisition*

R1-1.4.1 This scenario shows acquisition of test results data using the buffered approach. The buffered test data log is read at the end of entire lot processing. How the buffered data is being stored on tester is implementation specific.

Table R1-2 Buffered Test Data Acquisition

Host			*Test Equipment*	
SC introduces lot to equipment and other external systems.				
				Tester completes PP-SELECT and in READY state.
SC configures Test Data Collection plan on Tester and issues START command. See Test Result Data Collection section for details. *Assume all Testheads are assigned to file/bufferred Data Set "LOT12345.DLG".* DATALOG-DATASETS L,3 1. L,2 1. DSNAME=file:LOT12345.DLG 2. L,0 (all sites)	S2,F49	⇨		
		⇦	S2,F50	
				*Tester starts Testing and as Test Result data becomes available**, is buffered to assigned dataset name.***
Repeat until Test Completes …				
		⇦	S6,F11	Event Report Send (ERS) *Test Complete Event or State Transition Event associated with Test Complete, along with attached Bin Data Report.*
Event ACK	S6,F12	⇨		
Perform dispositioning on results from Bin Data report…				
Repeat until Lot is Complete …				
When all units for lot have been processed, before ending the lot, SC requests all buffered data from tester … ***Note:** This transfer step can be skipped if file is shared on network. In this case, just copy the file and reset the dataset.*				
Open Data Set Request (DSOR) L,2 1. HANDLE = H-LOT12345 (arbitrary but unique) 2. DSNAME = file:LOT12345.DLG	S13,F3	⇨		
		⇦	S13,F4	Open Data Set Data (DSOD) L,5 1. HANDLE = H-LOT12345 2. DSNAME = file:LOT12345.DLG 3. ACKC13 = 0 4. TYPE = STREAM 5. RECLEN = 0 (N/A for streams)

Host			*Test Equipment*	
Read Data Set Request (DSRR) L,2 1. HANDLE = H-LOT12345 2. READLN = 0 (read all available)	S13,F5	⇨		
SC processes incoming data …		⇦	S13,F6	Read Data Set Data (DSRD) L,4 1. HANDLE = H-LOT12345 2. ACKC13 = 0 3. CKPNT = … 4. L,n 1. <FILDAT> 2. … n. <FILDAT>
Upon completion of transfer, SC ends the lot.				

SEMI E123-0703
STANDARD FOR HANDLER EQUIPMENT SPECIFIC EQUIPMENT MODEL (HSEM)

This standard was technically approved by the Global Information and Control Committee and is the direct responsibility of the North American Information and Control Committee. Current edition approved by the North American Regional Standards Committee on November 22, 2002. Initially available at www.semi.org January 2003; to be published March 2003.

NOTICE: The designation of SEMI E123 was updated during the 0703 publishing cycle to reflect the addition of SEMI E123.1.

NOTICE: This standard replaces SEMI E30.2, which has been removed from publication as of the March 2003 (0303) publication cycle.

1 Purpose

1.1 This document establishes a Specific Equipment Model (SEM) for Handling equipment (HSEM). The SEM consists of equipment characteristics and behaviors that are applicable to this class of equipment. The intent of this document is to facilitate the integration of Handling equipment into an automated (semiconductor) factory. This document accomplishes this by defining an operational model for Handling equipment as viewed by a factory automation controller. This definition provides a standard host interface and equipment operational behavior.

2 Scope

2.1 The scope of this document is limited to the definition of Handling equipment behavior as perceived by a host. It defines the view of the equipment through the equipment communications interface. It does not define the internal operation of the equipment. It includes a specific processing state model as the basis for the behavior of all equipment of this class.

2.2 This document expands the handler equipment requirements and capabilities in the areas of the processing state model, collection event, alarm documentation, remote commands, variable item, and process program management.

NOTICE: This standard does not purport to address safety issues, if any, associated with its use. It is the responsibility of the users of this standard to establish appropriate safety and health practices and determine the applicability of regulatory or other limitations prior to use.

3 Limitations

3.1 *Communications* — HSEM-compliant equipment shall support the High Speed Messaging Service (HSMS) communication standard sending SEMI E5 messages over TCP/IP. The reason behind this requirement is the amount of data available for monitoring from this class of equipment. This specification deals only with the behavior of the handler in communicating with the host.

3.2 *Multi-Process-Site HSEM Implementations* — This SEM makes some demands and assumptions about the Handler with multiple process-sites in a configuration. These requirements are as follows:

3.2.1 Handling equipment in a multiple process-site configuration (i.e., lead conditioning site, electrical test-site) provides identification and status information (see Variable Item) at both the site level and the subsite level. An example could be a handler with both a lead conditioning site and an electrical test-site, with the electrical test-site containing multiple subsites (i.e., test heads).

4 Referenced Standards

4.1 *SEMI Standards*

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

SEMI E37 — High-Speed SECS Message Services (HSMS) Generic Services

SEMI E37.1 — High-Speed SECS Message Services Single Selected-Session Mode (HSMS-SS)

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

5 Terminology

5.1 *Definitions*

5.1.1 *alignment location* — Location that individual packaged units are placed at the process-site (e.g., electrical test).

5.1.2 *chaining* — The process of execution over multiple lots or runs with the same Process Program and the same handler operating conditions.

5.1.3 *electrical test-site* — A process-site on the equipment which is coupled with electrical testing

equipment for purposes of performing package electrical testing.

5.1.4 *execution area* — The area from which a current copy of the process program instructions is executed.

5.1.5 *handling equipment* — An equipment class generally consisting of integrated mechanisms and controls for the purpose of manipulating packaged devices, trays, and tubes during the manufacturing process.

5.1.6 *indexing* — The controlled stepped movement of material through the handler.

5.1.7 *kit* — Specific items of hardware and software as specified by the equipment manufacturer that adapt the equipment for a specific unit or unit package.

5.1.8 *leadconditioning site* — A process-site on the handler where some form of conditioning occurs on the package leadfingers (i.e., warming).

5.1.9 *leadfinger (or substrate connector lead)* — (1) In ceramic packages, an area of refractory metal that has been plated and is designated for the attachment to a process-site. (2) The area of the unit designated for attachment to a process-site.

5.1.10 *leadframe* — A sheet metal framework upon which a chip (sometimes chips) is attached, wire-bonded, and then either molded with plastic epoxy or with ceramic and/or metal.

5.1.11 *media* — A temporary material carrier used to hold and transport units/devices (tubes, trays, etc.).

5.1.12 *media map* — Formatted data used to map functionally good and bad units/devices to an X, Y, Z location in the media. Maps can be requested by the handler for use prior to processing and then updated after processing.

5.1.13 *off-line programming (OLP) utility* — Utility to create, edit, and format process programs on a computer, as opposed to creating process programs at the equipment.

5.1.14 *process-site* — A location on the equipment where work is performed on a packaged device (i.e, electrical test-site, lead conditioning site).

5.1.15 *process subsite* — An addressable portion of a process-site.

5.1.16 *reset* — The action of changing the value of a variable, such as wafer count (usually to zero).

5.1.17 *safe state* — A state in which the equipment presents no danger to the product or user. This implies that safety interlocks are in place such that the equipment can be serviced without harm to the operator and that the material being processed has been removed from the processing station into an accessible location.

5.1.18 *slot* — A position in a carrier where a leadframe, tray, tube, or other media element may reside.

5.1.19 *sort category* — Handler specific sorting category related to a physical buffer within a handler where parts are stored after processing based on processing results. Typically there is some software mapping between process results (e.g., Test Result Bins) and a Sort Category.

5.1.20 *tray* — A flat rectangular form of media for holding singulated packaged units. Also referred to as waffle packs or matrix trays. A tray is generally molded plastic with a defined matrix of cells or pockets tailored for specific packages.

5.1.21 *tube* — A hollow form of media for holding packaged units. Also referred to as rails or sticks. A tube is generally composed of extruded polymer with internal section dimensions and features tailored for a specific package.

5.1.22 *unit* — Discreetly addressable element, such as an integrated circuit or chip, handled individually or in groups by a handler for a process tool.

6 State Model

6.1 *State Model* — The purpose is to define the equipment-specific processing state model and other state models necessary to portray the expected operational states of the equipment to enable host tracking and control in place of a local operator.

6.2 *State Model Requirements*

6.2.1 The processing state models in this document are required for implementing an HSEM-compliant handler. A state model consists of the following state model diagram, state definitions, and state transition tables.

6.2.2 A state model represents the host' s view of the handler, not necessarily the actual handler' s internal operations.

6.2.3 All HSEM state model transitions shall be mapped sequentially into the actual equipment events that satisfy the requirements of those transitions. In certain implementations, the handler may enter a state and have already satisfied all of the conditions required by the HSEM state model for transition to another state. The handler makes the required transition without any additional actions in this situation.

6.2.4 Some equipment may need to include additional states. However, any additional states must not change

the HSEM-defined state transitions. All expected transitions between HSEM states must occur.

6.3 *HSEM State Model*

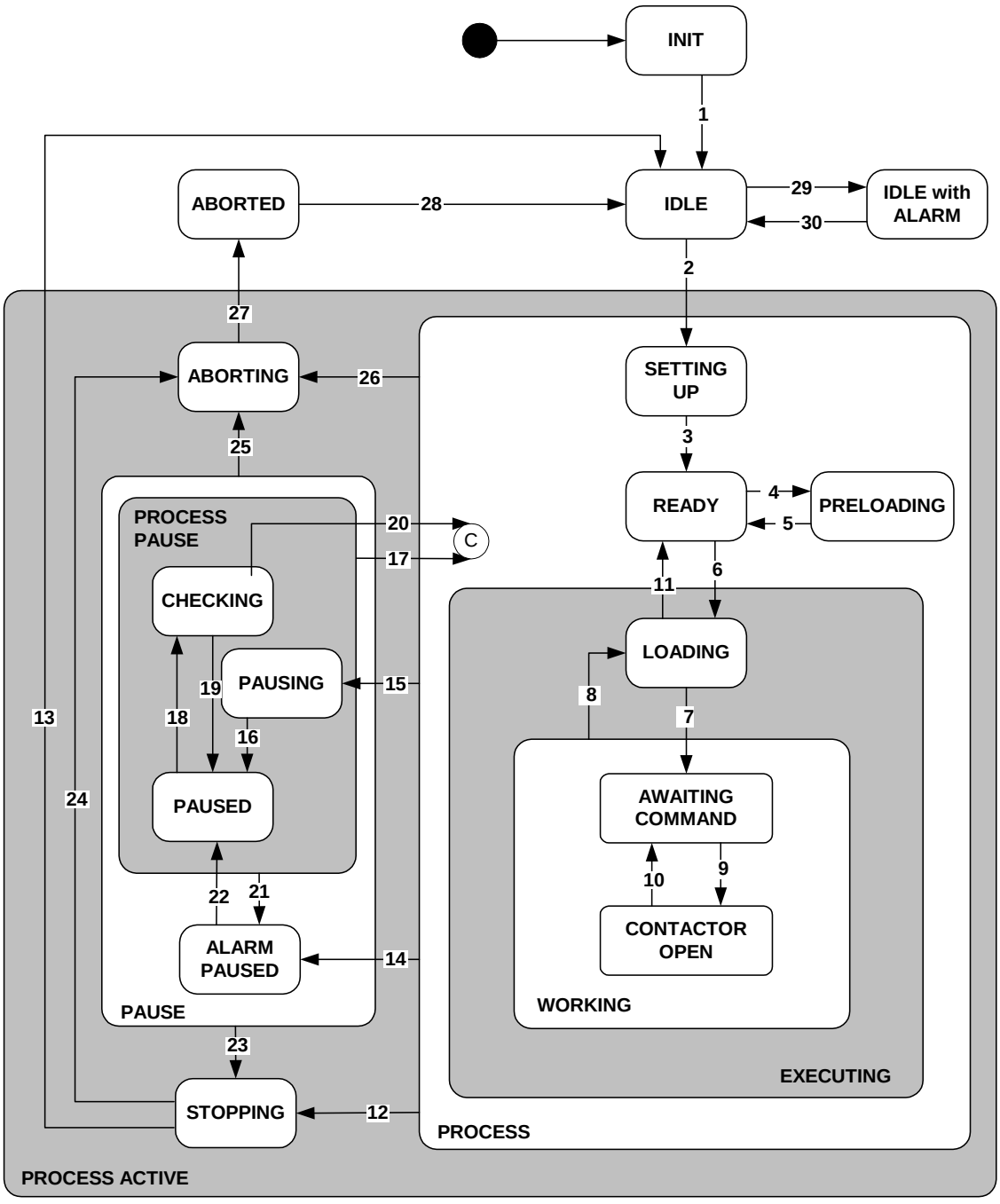

Figure 1
HSEM Processing State Model

6.4 *Description of Handler Processing States*

6.4.1 *ABORTED* — All activity is suspended as a result of an ABORT command. Any alarm and abort conditions must be cleared and verified by an operator before exit from this state.

6.4.2 *ABORTING (PROCESSING ACTIVE Sub-State)* — The handler has received an ABORT command. All activity is suspended. The handler is taking appropriate action to bring itself and material to a "safe" state where possible. Unit or Lot data may be invalid or not available.

6.4.3 *ALARM PAUSED (PAUSE Sub-State)* — An alarm has occurred in the PROCESS or PROCESS PAUSE states, and the handler is waiting for the alarm to be cleared.

6.4.4 *CHECKING (PROCESS PAUSED Sub-State)* — The handler verifies that updates made to the process program are valid (i.e., possible errors induced via an operator during the pause).

6.4.5 *EXECUTING (PROCESS Sub-State*) — The handler is processing material automatically and can continue to do so without external intervention. This state may include interaction with the host or operator.

6.4.6 *IDLE* — Awaiting a command. IDLE is free of ALARMS and error conditions.

6.4.7 *IDLE with ALARMS* — An alarm has occurred in the IDLE state, and the handler is waiting for all alarms to be cleared.

6.4.8 *INIT* — Handler initialization is occurring.

6.4.9 *LOADING (EXECUTING Sub-State)* — This is the state the next unit or units are transferred from the input media or input buffers to the process site(s) and/or removed from the process site(s) to sort categories.

6.4.10 *PAUSE (PROCESSING ACTIVE Sub-State)* — The PROCESS state will be suspended at the completion of the current unit or next opportunity. Actions to put the handler in a safe state are performed. The handler is awaiting a command (RESUME, STOP, or ABORT) or for alarm(s) to be cleared.

6.4.11 *PAUSED (PROCESS PAUSE Sub-State)* — The PROCESS state has been suspended, and the handler is waiting for a command (RESUME, STOP, or ABORT). In this state, the operator may correct error conditions that do not affect the current Process Program selection. One of the possible corrective actions is for the operator to manually align the units being processed.

6.4.12 *PAUSING (PROCESS PAUSE Sub-State)* — The PROCESS state will be suspended at the completion of the current unit or next opportunity. The handler cannot transition to PAUSED state until the current unit is completed and the handler is in a "safe state".

6.4.13 *AWAITING COMMAND (WORKING Sub-State)* — Unit is moved to process-site (e.g., for electrical test, to insert into contactor). Actuator piston(s) is extended to the alignment test site(s) if applicable. If contactors must be "closed" before processing can occur, then contactors are in the closed condition. In this state, handler has completed all loading and is awaiting the next command from Host to signal process complete.

6.4.14 *CONTACTOR OPEN (WORKING Sub-State)* — Unit process is complete and host issued a BREAK-CONTACT or RECONTACT command. (e.g. Actuator piston(s) is retracted from the alignment test site(s), contactors are open.) If RECONTACT command was issued, unit will return to alignment for a step and repeat. Otherwise Handler will remain in this state until a MAKE-CONTACT is received to resume processing. Commands such as STOP, ABORT and PAUSE are still valid and follow the respective required transitions.

6.4.15 *PRELOADING (PROCESS Sub-State)* — Handler has received a PRELOAD-UNITS command to begin preloading units to soak chambers or some equivalent process buffers. Units should not be transferred to process sites in this state. During this time, test equipment may also go through its internal setup process in parallel, such as testhead calibrations. If handler does not implement any process buffers, the remote command from Host must still be accepted and state transition followed, in which case handler will transition back to READY state immediately with PreloadComplete event sent to Host.

6.4.16 *PROCESS (PROCESS Sub-State)* — This state is the parent of those sub-states that refer to the preparation and execution of a process program.

6.4.17 *PROCESS PAUSE (PAUSE Sub-State)* — The handler is free of alarm conditions in the PAUSE state.

6.4.18 *READY (PROCESS Sub-State)* — The handler is ready to begin processing and is awaiting a START or PRELOAD-UNITS command from the operator or host.

6.4.19 *SETTING UP (PROCESS Sub-State)* — The handler is satisfying conditions so that processing can begin. This includes the receipt of any process programs, the material to be processed, and machine-specific calibration. While in this state, the handler can be single-stepped through each process in order for the operator to ensure that the handler is moving the unit correctly.

6.4.20 *STOPPING (PROCESSING ACTIVE Sub-State)* — The handler has completed a Process Program or has been instructed to stop processing and shall do so at the next opportunity. All necessary cleanup is completed within this state with regard to material, data, control system, etc. Data is preserved. Any error condition is cleared before exiting from this state.

6.4.21 *WORKING (EXECUTING Sub-State)* — The handler is processing a specific unit.

6.5 *HSEM Processing State Transitions Table*

6.5.1 Unless specified, if a command is stated as a possible trigger for a state transition, it is implied that the command or its equivalent can come from the host or operator or any other command interface provided by the equipment, for example a GPIB command interface.

Table 1 Processing State Transitions Table

#	*Current State*	*Trigger*	*New State*	*Actions*	*Comments*
1	INIT	All handler initialization is complete with no alarms or error conditions.	IDLE	None	None
2	IDLE	A Process Program is selected.	SETTING UP	Handler-dependent	Commit has been made to setup.
3	SETTING UP	All setup activity has completed, and the handler is ready to receive a START command.	READY	The handler is waiting for a START command.	The selected Process Program is available for execution, and material is present at the input port.
4	READY	Handler receives a PRELOAD-UNITS command.	PRELOADING	Handler starts preloading units to temperature soak chambers or some process buffers, where some type of pre-work can be applied to units before processing. No units should be transferred to process sites at this point.	For handlers that do not support process buffers of any type, this will just be a pass-thru state. Transition #5 is taken immediately.
5	PRELOADING	Handler completes preloading of units to its process buffers, e.g. unit(s) is ready to be transferred to process site(s) from buffers.	READY	PreloadComplete collection event reported to Host.	
6	READY	The handler receives a START command.	LOADING	Transfers the next unit to the process-site.	LOAD is an EXECUTING sub-state.
7	LOADING	A unit(s) is loading to the process –site(s) or been removed from process site(s).	AWAITING COMMAND	The actuator(s) is extended to the alignment process site(s) if applicable. If unit(s) is loaded to process site(s), UnitsReady event is sent to Host with loaded status of each process site.	None
8	WORKING	The processing of the current unit(s) completes normally and handler received a BIN-UNITS command.	LOADING	This unit(s) is transferred from the process sites specified in the remote command and next unit(s) is loaded to the now empty process site(s).	None
9	AWAITING COMMAND	The processing of the current unit(s) completed. The handler has received a RECONTACT or BREAK-CONTACT command.	CONTACTOR OPEN	The actuator(s) is retracted from the alignment test site(s). If RECONTACT command is received, handler will attempt to contact unit(s) again.	None

#	*Current State*	*Trigger*	*New State*	*Actions*	*Comments*
10	CONTACTOR OPEN	The handler has received a MAKE-CONTACT command or continuation of RECONTACT and handler has completed contacting of the unit(s).	AWAITING COMMAND	The actuator(s) is extended to the alignment process site(s). UnitsReady event is sent to Host with loaded status of each site.	None
11	LOADING	The handler completes unload of last unit and there are no more unit(s) available for loading.	READY	The handler completes sorting the last unit(s). LastUnitCompleted collection event reported to Host.	None
12	PROCESS	The handler has received a STOP command.	STOPPING	The handler completes the current unit in the WORKING state and unloads it.	The handler begins its cleanup procedure.
13	STOPPING	The handler cleanup is complete, and the handler is free of alarms.	IDLE	None	None
14	PROCESS	An alarm occurs.	ALARM PAUSED	PROCESS activity is suspended, and the handler is waiting for all alarms to be cleared.	ALARM PAUSED is a PAUSE sub-state.
15	PROCESS	The handler has received a PAUSE command.	PAUSING	The PROCESS state shall be suspended at the completion of the current unit. Any necessary actions to put the handler in a safe state are performed.	PAUSING is a PAUSE sub-state.
16	PAUSING	The handler has completed Processing the Current unit in the WORKING state and achieved a safe condition.	PAUSED	The handler is waiting for a command (RESUME, STOP, or ABORT).	None
17	PROCESS PAUSE	The handler has received a RESUME command.	STATE based on conditional table.	Return to previous state or state per process resume conditions.	None
18	PAUSED	A RESUME command was received.	CHECKING	Validation of the Process Program Parameters.	None
19	CHECKING	Error was detected in new parameter setting being validated in the CHECKING state.	PAUSED	None	None
20	CHECKING	Parameter checking completes successfully.	STATE based on conditional table.	Return to previous state or state per process resume conditions.	None
21	PROCESS PAUSE	An alarm is set by the handler.	ALARM PAUSED	The handler waits for all alarms to be cleared or for a STOP or ABORT command.	None
22	ALARM PAUSED	All alarms are cleared.	PAUSED	The handler is waiting for a command (RESUME, STOP, or ABORT).	None
23	PAUSE	The handler has received a STOP command.	STOPPING	The handler proceeds with cleanup.	Data is preserved and is valid.
24	STOPPING	The handler has received an ABORT command.	ABORTING	Any unsafe condition is resolved, if possible.	Data may be invalid or unavailable.
25	PAUSE	The handler has received an ABORT command.	ABORTING	Any unsafe condition is resolved, if possible.	Data may be invalid or unavailable.
26	PROCESS	The handler has received an ABORT command from operator, host, or self-generated.	ABORTING	The handler is put in a “safe” state.	Unit or lot data may be invalid or not available.

#	Current State	Trigger	New State	Actions	Comments
27	ABORTING	Unsafe conditions have been resolved where possible.	ABORTED	The handler is waiting for alarm and ABORT conditions to be cleared.	The only state change allowed is to IDLE.
28	ABORTED	An operator has verified that all alarms and abort conditions have been cleared.	IDLE	None	None
29	IDLE	An alarm is set.	IDLE with ALARMS	The handler waits for all alarms to be cleared.	None
30	IDLE with ALARMS	All alarms have been cleared.	IDLE	None	The IDLE state is free of alarms.

6.6 *Process Model Conditions Table*

Table 2 Process Resume Conditions

Condition	Next State
Checking determines that process program conditions were changed.	SETTING UP
Previous State PRELOADING.	PRELOADING
Previous State LOADING.	LOADING
Previous State AWAITING COMMAND.	AWAITING COMMAND
Previous State CONTACTOR OPEN.	CONTACTOR OPEN
Previous State READY.	READY
Previous State was SETTING UP.	SETTING UP

7 Collection Event List

7.1 *Requirements*

7.1.1 ALL SEMI E30-required Events are required by the HSEM. Since a Processing State Model is required by the HSEM, all state transitions are required Events.

7.1.2 This section of the HSEM lists only those collection events that are not associated with a change of state or those requiring specific data variables or Reports defined in the HSEM.

7.2 *Collection Event Tables* — The first table contains required events and associated reports. The second table contains required events and associated data variables.

Table 3 Processing State Transitions Requiring Report Levels

Transition	From State	To State	Required variables or Report
SETUP COMPLETE (3)	SETTING UP	READY	Setup Report

Table 4 Other Required Collection Events

Collection Event Name	Description	Typical Variable Data	Reference / Comments
CarrierEmpty	Last item from carrier was removed.	MediaID	Configuration specific event (CSE). EXECUTING event.
CarrierFull	Output carrier capacity is full.	MediaID	CSE. Valid in EXECUTING event.
ReaderFailed	Any type of failure related to reading of units.	ReaderType, ReaderErrorType	CSE. PROCESSING event
UnitCntInterval		UnitCountInterval, Clock	EXECUTING event.

Collection Event Name	*Description*	*Typical Variable Data*	*Reference / Comments*
MediaCntInterval		MediaCountInterval	EXECUTING event.
SkipCntInterval		SkipCountInterval	CSE. EXECUTING event.
MediaChange		MediaID, MediaType	EXECUTING event.
SortComplete	Sorting of unit(s) from last BIN-UNITS command to sort categories has completed.	CategoryCount	EXECUTING event.
InputsEmpty	No materials available on direct inputs. There may still be unit(s) in process buffers or under processing.	UnitCount, CategoryCount	EXECUTING event.
LastUnitCompleted	Handler completed unloading last unit, i.e. no materials available in all inputs or buffers.	UnitCount, CategoryCount	EXECUTING event.
UnitsReady	At load complete, this event notifies Host that unit(s) is ready to be processed and indicates loaded status of process site(s).	See Units Ready Report	EXECUTING event.
PreloadComplete	Handler completes preloading of units to its process buffers, e.g. unit(s) is ready to be transferred to process site(s) from buffers.	BufferID, BufferType	CSE. PROCESSING event.
BufferEmpty	One of process buffer emptied. ID of buffer identifies which buffer.	BufferID, BufferType	CSE. EXECUTING event.

8 Variable Items

8.1 The purpose of this section is to define the list of variable items required by the HSEM. Values of these variables are available to the host through collection event reports and host status queries.

8.2 *Requirements*

8.2.1 All generic variable items defined in SEMI E30 are required by all HSEM equipment.

8.2.2 Variable items required by HSEM are categorized as follows:

- *Common Variables (CVs)* — Variables common to all handlers.
- *Configuration-Specific Variables (CSVs)* — Variables associated with a specific configuration of the above equipment class.

8.3 *Data Types*

8.3.1 Equipment constants have various uses in HSEM, including the following:

- Equipment offsets that match the performance of several pieces of equipment that would otherwise perform differently due to inherent manufacturing differences. Examples are home values and motion axis scaling factors.
- Setting the configuration of the equipment to allow for different material specifications, equipment options, material flows, frequency of automatic functions, etc. Examples are media and yield check frequency.
- Managing optional machine features. Examples are constants that indicate whether optional features, such as automated media stackers, are present and control the configuration and function of these optional subsystems when they are present.

8.3.2 Status Variables are valid at all times. A status variable may not be changed by the host but may be changed by the equipment or operator. The value of status variables may be queried by the host at anytime.

8.3.3 Variables that are valid upon the occurrence of specific collection event can be queried by the host. An attempt to read a data variable at the wrong time shall not generate an error, but the data reported may not have relevant meaning.

8.3.4 *Data Item Requirements for Multi-Head, Multi-Site Equipment* — The identification for multi-head and multi-site data (data variable, status variables, events, etc.) is addressed in this specification through the use of status variables with list structures. In the table below, the subheading “Process-Site Group” contains variables that must be available for all process-sites on the handler equipment. When multiple process-sites exist, either a list structure or table structure may be used to show multiple occurrences of a specific variable.

8.4 *Variable Item Table*

Table 5 Variable Item Table

Variable Name	*Category*	*Description*	*Format*	*Comments*
Physical Handler Group				
ProcessBuffers	CSV	ID of all implemented input and output buffers and their type.	Structure (List of) BufferID BufferType	This is a configuration specific variable item (CSV). List by number of buffers.
BufferID	CSV	Input or Output Buffer ID, valid in reports only.	Text	
BufferType	CSV	Input or Output Type, valid in reports only.	Integer	
CategoryID	CV	List of sort category IDs currently configured in handler.	(List of) sort category ID	Valid in all states. List by category.
CategoryCount	CSV	Count of units in each sort category.	Structure (List of) sort category ID UnitCount	Valid in PROCESS states. List by category.
EquipSerialID	CV	Identification of Equipment.	Text	Valid in all states.
KitID	CSV	ID of unique tooling unit.	Text	Valid in all states.
LightPoleStatus	CSV	Color/status (i.e., Red/flash)	Text	Valid in all states.
LinkPortStatus	CSV	(3 = Input/Output linked, 2 = Input linked, 1 = Output linked, 0 = HANDLER not Linked)	Enumerated	Valid in all sub-states.
MediaID	CV	Media Serial Number	Text	Valid in Executing state.
MediaType	CV	Media Type	Text	Valid in Executing state.
OperationType	CSV	Current Operation Mode (i.e., maintenance, production).	Text	Valid in all states.
OperatorID	CSV	Current Operator ID.	Text	Valid in all states.
QueueStatus	CV	PPID Queued to be run.	Integer	Valid in all states.
ReaderType	CSV	Type of reader installed on handler.	Text	Supplier defined.
ReaderErrorType	CSV	Type of error detected by the material reader.	Text	Supplier defined.
Process-Site Components				
AlignmentCount	CSV	Number of units since last alignment (i.e., Homing/Adjustment).	Integer	Valid in all states.
InsertionForce	CSV	Insertion-Force energy	Float	Valid in PROCESSING state.
InsertionForceSetpoint	CSV	Insertion-Force set point (setpoint).	Float	Valid in all states.
MediaChangeTime	CV	Elapsed time to replace media and send ready.	Text	Valid in all states.
MediaCount	CV	Number of media since last reset.	Integer	Valid in all states.
MediaCountInterval	CV	Event generated when number of media is completed.	Integer	Valid in all states.
PresentPositionActual	CSV	Present position actual.	Integer	Valid in all states.
PresentPositionSetpoint	CSV	Present position set points (setpoint).	Integer	Valid in all states.
ProcessSiteTemp	CSV	Process-site temperature (setpoint).	Float	Celsius - Set point.
ProcessSiteID	CV	ID of process-site in configuration.	Integer	Valid in all states.

Variable Name	*Category*	*Description*	*Format*	*Comments*
ProcessSiteStatus	CV	$Site_n$ Availability(1 = enabled, 0 = disabled)	Enumerated	Valid in all states. List by sites.
ProcessSiteCount	CV	Total count of process sites.	Integer	Valid in all states.
ProcessSiteLoaded	CV	$Site_n$ loaded status, normally associated with UnitsReady event.	Integer	Valid in EXECUTING states. List by sites.
SkipCount	CSV	Number of units skipped since last reset (Skip + Process = Total).	Integer	Valid in all states.
SkipCountInterval	CSV	Event generated when number of units is skipped.	Integer	Valid in all states.
StartProcessPortID	CSV	Start process source (i.e., hand, keyboard, host)	Integer	Valid in all states.
UnitCount	CV	Number of units since last reset.	Integer	Valid in all states.
UnitCountInterval	CV	Event generated when number of units is completed.	Integer	Valid in all states.
UnitPosition	CSV	X, Y, Z media location of a unit.	Integer	Valid in all states.
UnitStatus	CSV	(1 = Processed, 0 = Skipped)	Integer	Valid in all states.

8.5 *HSEM Required Reports* — The reports below are required as "canned" or preconfigured reports by HSEM. HSEM does not require the equipment to guarantee the accuracy of data identified in these reports outside the PROCESSING ACTIVE state defined in the HSEM process state model.

8.5.1 *Setup Report* — Table 6 contains variables that are required to be available at the setup complete event.

Table 6 Required Variables at Setup Complete Event

Variable Name	*Notes*
KitID	Configuration Kit
MediaID	Serial # of Media
PPExecName	Current value of the selected PPID
EquipID	
InsertionForceSetPoint	

8.5.2 *Process Report* — Table 7 contains variables that must be available when the equipment is in the PROCESSING state.

Table 7 Required Variables for PROCESSING State

Variable Name	*Notes*
AlignmentCount	
ProcessSiteTemp	
OperatorID	
OperationType	

8.5.3 *SPC Report* — The table below contains variables that must be available and reported at the completion of a unit.

Table 8 Required Variables at Completion of Unit

Variable Name	*Notes*
UnitCount	
SkipCount	
SkipCountInt	

Variable Name	*Notes*
UnitCountInt	
MediaCount	
MediaCountInterval	
MediaID	
OperatorID	

8.5.4 *Units Ready Report* — The table below contains variables that must be available and reported when units are ready to be processed (i.e. at load complete).

Table 9 Required Variables at Load Complete

Variable Name	*Notes*
ProcessSiteLoaded	Site loaded status
ProcessSiteStatus	List of sites that are available
ProcessSiteCount	Total process site count

9 Process Program Management

9.1 *Process Program Requirements*

9.1.1 The HSEM requires that the capability of process program management be fully supported for this class of equipment. The HSEM also requires that the process program have a structure that enables the user to build process programs with default conditions that can be overridden for a run. The concepts of process program structure and process program variable parameters are discussed in the following sections. The HSEM also requires the following:

- Minimum, maximum, and default parameter values must be defined for all process programs.
- Verification will occur when a process program is downloaded to the equipment; the program syntax must be verified by the equipment manufacturer.
- Parameter validation will occur when a process program is downloaded. Parameters must be type and range checked.
- Equipment should provide the functionality to manually or interactively modify the parameters set in the process program.
- An error message must be generated from the handler if the process program parameters are outside the range of the machine calibration.

9.2 *Process Program Structure*

9.2.1 A handler process program must contain the following information:

- Machine-specific configuration parameters,
- Process-Site-specific information section,
- Media-Type-specific information section, and
- Unit (Unit/Package) information section.

9.2.2 When combined, this information constitutes a complete process program. It is emphasized that the HSEM does not enforce the exact format and data type of each section. However, it does provide direction as to what each section should consist of.

9.2.3 *Machine-Specific Configuration Parameters* — Each brand or type of handler may have one or more machine-specific configuration parameters. Examples of such parameters would be input configuration, number of process-sites, and output configuration. Even though they are supplier-specific, these parameters nevertheless play a vital role in the overall generation or creation of a process program. Since the machine-specific parameters can differ from one equipment manufacturer to another, the HSEM does not specify the exact number, data types, and format of these parameters. These details are left to the sole discretion of the equipment manufacturer.

9.2.3.1 In addition, handler process program must support user defined mapping of logical test bin categories that are specific to test recipes and test equipment to handler sort categories. The mapping must be easy to define and flexible allowing ranges to be specified. Logical bin categories can range from 1–9999.

9.2.4 *Process-Site-Specific Information* — This process-site-specific section contains information necessary for the configuration and execution of the various process-sites configured on the equipment. Each equipment may contain different process-site configurations. Since these configurations will differ, the HSEM does not specify the exact number, data

types, and format of these parameters. HSEM does recommend a minimum list of data items for the common handler process-sites. These include:

9.2.4.1 *Thermal Conditioning Site Parameters*

- Temperature Set Point
- Upper-Temp Guardband Set Point
- Lower-Temp Guardband Set Point
- Soak Time
- Test-Site Temp

9.2.4.2 *Test-Site Parameters*

- Device Pick Up/Place
- Speed of Device Pick Up/Place
- Device Insertion/Retraction
- Speed of Device Insertion/Retraction
- Speed of Index Mechanism
- Insertion/Place Force/Stroke

9.2.4.3 *Lead Condition Site Parameters*

- Device Pick Up/Place
- Speed of Device Pick Up/Place
- Speed of Insertion
- Insertion Force/Stroke

9.2.4.4 *Sort Sites Parameters*

- Device Pick Up/Place
- Speed of Device Pick Up/Place
- Device Index
- Insertion Force/Stroke
- Sort Category Set
- Sort Category – Full/Empty/Partial
- Sort Media – In place or empty.

9.2.5 *Unit/Device-Specific Information* — The unit/device-specific section contains information necessary for the configuration and execution of the specific units to be handled by the equipment. HSEM requires a minimum list of data items be available to determine package dimensions, terminal dimensions, package height, and coplanarity.

9.2.6 *Media-Type-Specific Information* — The media-type-specific information section contains information necessary for the configuration and execution of the specific media type in use on the equipment. HSEM requires a minimum list of data items be available to determine row/column count, X/Y distance to a cell, device height in tray, media height, and X/Y pitch.

9.3 *Methods of Process Program Creation* — The method by which an equipment manufacturer creates a process program may be unique to that manufacturer. However, it is required that the customer at least be given both of the following options for the creation of a process program.

9.3.1 *Off-Line Development* — Using this method, the customer is given a set of software tools (process program compilers, decompilers, and debuggers) that enables him/her to generate or create a process program using the above mentioned information (flow, parameter, functional test, etc.). The newly generated process program then is downloaded onto a specific handler, verified, and is now ready to be selected and executed locally by the operator or remotely by the host computer. If this process is used, the supplied software tools should closely mimic or simulate a handler so that a user can create a complete process program. In many situations, minor adjustments may be needed to the process program on the equipment before it is completely ready for execution.

9.3.2 *On-Line Development* — The second option is to enable the user to download the above-mentioned information (tables or files) onto the equipment and create the actual process program on the equipment itself.

10 Remote Commands

10.1 The purpose of this section is to identify remote commands, command parameters, and valid commands versus states in the processing state models.

10.2 *Requirements*

- All the remote commands defined by HSEM are required.
- The alphanumeric strings defined by HSEM for remote commands and command parameters are required.
- If additional remote commands are supported, then the “Remote Command Versus Valid States” matrix must be generated for these additional commands. Place an “X” in the table for each state in which a given command is valid.

10.3 *Remote Command Descriptions*

10.3.1 *ABORT* — This command terminates the current processing. ABORT makes no guarantee about completion of the current unit. Levels of ABORT may be specified by selecting a particular process site or the equipment itself. "Cleanup" removes all material that belongs to the lot and then the equipment enters the aborting state.

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
RemoteCommand	M		"ABORT"
ProcessSiteID	C	-	ID of handler process site.
Cleanup	C	-	
Ack	-	M	Response acknowledge.
Status	-	C	Error code and Error text if any

10.3.2 *BIN-UNITS* — This command instructs the handler to start unloading the units from process sites and transfer to output locations. The required bin list parameter combined with the currently loaded mapping of bin values to output locations will tell handler how to disposition the units.

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
RemoteCommand	M	-	"BIN-UNITS"
Bins	M	-	Bin results of current processed units, one bin value per test site. Bin values for disabled or empty sites are ignored.
Ack	-	M	Response acknowledge.
Status	-	C	Error code and Error text if any.

10.3.3 *BREAK-CONTACT* — This command instructs the handler to break contact on the unit(s) at the specified site(s). This allows user to remove parts from socket(s), replace socket(s) if applicable, perform testhead calibration etc.

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
RemoteCommand	M	-	"BREAK-CONTACT"
Sites	M	-	List of sites to break contact. Empty list implies all sites.
Ack	-	M	Response acknowledge.
Status	-	C	Error code and Error text if any.

10.3.4 *DISABLE-SITE* — This command instructs handler to disable sites from been used for processing.

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
RemoteCommand	M	-	"DISABLE_SITE"
Sites	M	-	List of sites to disable. Empty list implies all sites.
Ack	-	M	Response acknowledge.
Status	-	C	Error code and Error text if any.

10.3.5 *ENABLE-SITE* — This command instructs the handler to enable sites for processing.

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
RemoteCommand	M	-	"ENABLE-SITE"
Sites	M	-	List of sites to enable. Empty list implies all sites.
Ack	-	M	Response acknowledge.
Status	-	C	Error code and Error text if any.

10.3.6 *MAKE-CONTACT* — This command instructs the handler to recontact the unit(s) at the specified site(s) and resume processing. This command is only available to the Host after a BREAK-CONTACT command has been issued or selected sites have units that have broken contact.

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
RemoteCommand	M	-	"MAKE-CONTACT"
Sites	M	-	List of sites to make contact after a break command is issued. Empty list implies all sites.
Ack	-	M	Response acknowledge.
Status	-	C	Error code and Error text if any.

10.3.7 *PAUSE* — This command transitions the handler to the PAUSING process state when the current unit/media completes processing.

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
RemoteCommand	M	-	"PAUSE"
ProcessSiteId	C	-	ID of handler process site.
Ack	-	M	Response acknowledge.
Status	-	C	Error code and Error text if any.

10.3.8 *PP-SELECT* — This command instructs the handler to copy the indicated Process Program from non-volatile storage to the handler's Process Program execution area. Process Program Variable Parameters can be specified in this command which modify the default values for these Variable Parameters in the Process Program.

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
RemoteCommand	M	-	"PP-SELECT"
ProcessSiteID	C	-	ID of handler process site.
LotID	C	-	Lot to be processed.
ProcessProgramID	M	-	Id of the processes program to be used.
MediaList	C	-	One or more media to be processed with this program.
Ack	-	M	Response acknowledge.
Status	-	C	Error code and Error text if any.

10.3.9 *PRELOAD-UNITS* — This command instructs the handler to begin preloading unit(s) to temperature soak chambers or some equivalent process buffers. Handler transitions to PRELOADING state while action is performed and returns to READY upon completion, reporting PreloadComplete collection event to Host. Command must still be accepted even if handler does not support process buffers or preloading, in which case, it transitions back to READY state immediately.

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
RemoteCommand	M	-	"PRELOAD-UNITS"
Ack	-	M	Response acknowledge.
Status	-	C	Error code and Error text if any.

10.3.10 *RECONTACT* — This command instructs the handler to re-contact the unit(s) at the specified sites and is only available in the WORKING state. Handler transitions to CONTACTOR OPEN state while action is performed. Once unit(s) is contacted again, UnitsReady event is sent to Host and handler transitions back to AWAITING COMMAND or LOADING state if there are more process site(s) available for loading.

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
RemoteCommand	M	-	"RECONTACT"
Sites	M	-	List of sites to re-contact. Empty list implies all sites.
Ack	-	M	Response acknowledge.
Status	-	C	Error code and Error text if any.

10.3.11 *RESUME* — This command resumes processing from the point where the process was PAUSED. This command is only recognized if the handler is in the PAUSED state.

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
RemoteCommand	M	-	"RESUME"
ProcessSiteID	C	-	ID of handler process site.
Ack	-	M	Response acknowledge.
Status	-	C	Error code and Error text if any.

10.3.12 *START* — This command is only available to the host or operator when a process program has been selected and the handler is in the READY processing state. The START command instructs the handler to initiate processing.

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
RemoteCommand	M	-	"START"
ProcessSiteID	C	-	ID of handler process site.
Ack	-	M	Response acknowledge.
Status	-	C	Error code and Error text if any.

10.3.13 *STOP* — This command completes the current unit, stops in a safe condition, and returns to the IDLE processing state. STOP has the intent of bringing about a normal termination after completion of the current unit.

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
RemoteCommand	M	-	"STOP"
ProcessSiteID	C	-	ID of handler process site.
CloseLot	C	-	Automatically close the lot.
Ack	-	M	Response acknowledge.
Status	-	C	Error code and Error text if any.

10.3.14 *RESET-TOOL-COUNTS* — This command will initialize equipment tool counts. The minimum set are those contained in the Variable Items section.

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
RemoteCommand	M	-	"RESET-TOOL-COUNTS"
ProcessSiteID	C	-	ID of handler process site.
StatusVariableList	M	-	List of Status variables to reset.
Ack	-	M	Response acknowledge.
Status	-	C	Error code and Error text if any.

10.3.15 *PURGE* — Purge flush or clean the equipment of process material.

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
RemoteCommand	M	-	"RESET-TOOL-COUNTS"
Ack	-	M	Response acknowledge.
Status	-	C	Error code and Error text if any.

10.4 *Remote Commands and HSEM Process Model Mapping* — Tables 10 and 11 illustrates the relationship between remote commands and states of the HSEM processing state model. An "X" indicates that a command is valid for use in this state. If a remote command is attempted during a non-valid state, the equipment would reject the remote command.

Table 10 Remote Commands vs. Process States

COMMAND							
ABORT							
BIN-UNITS							
PAUSE							
PP-SELECT							
RESUME							
START							
STOP							
PROCESSING STATE							
IDLE				X			
PROCESSING ACTIVE							
PROCESS							
SETTING UP					X		X
READY	X	X			X		X
PRELOADING	X				X		X
EXECUTING							
LOADING	X						X
WORKING							
AWAITING COMMAND	X				X	X	X
CONTACTOR OPEN	X				X		X
PROCESS PAUSE							
PAUSING							X
PAUSED	X		X				X
CHECKING				X			X
ALARM PAUSED			X				X
ABORTED							

Table 11 Remote Commands vs. Process States (cont.)

COMMAND	RESET-TOOL-COUNTS	BREAK-CONTACT	MAKE-CONTACT	PURGE	RECONTACT	ENABLE-SITE / DISABLE-SITE	PRELOAD-UNITS
PROCESSING STATE							
IDLE		X					X
PROCESSING ACTIVE							
PROCESS							
SETTING UP							
READY	X	X		X			X
PRELOADING							
EXECUTING							
LOADING							
WORKING							
AWAITING COMMAND			X			X	
CONTACTOR OPEN					X		
PROCESS PAUSE							
PAUSING							
PAUSED		X		X			X
CHECKING							
ALARM PAUSED				X			
ABORTED				X			

11 Additional SEMI E30 Requirements

11.1 The purpose of this section is to specify any GEM additional capabilities that are required to be supported by this class of equipment.

11.2 *Requirements* — The following GEM additional capabilities required by HSEM are:

- Establish Communications,
- Dynamic Event Report Configuration,
- Variable Data Collection,
- Status Data Collection,
- Alarm Management,
- Remote Control,
- Equipment Constants,
- Process Program Management,
- Equipment Terminal Services,
- Clock,
- Spooling, and
- Control (Host-Initiated).

12 Alarm Definition

12.1 SEMI E30 has a very restrictive definition of alarms. This standard requires a broader definition.

12.2 Exceptions are classified into two categories: errors and alarms. Alarms may be divided into categories as follows:

1. *Personal Safety* — Condition may be dangerous to people.
2. *Equipment Safety* — Condition may harm equipment.
3. *Parameter Control Warning* — Parameter variation outside of preset limits — may harm product.

4. *Parameter Control Error* — Parameter variation outside of reasonable control limits — may indicate an equipment malfunction.

5. *Irrecoverable Error* — Intervention required before normal use of equipment can resume.

6. *Equipment Status Warning* — An unexpected condition has occurred, but operation can continue.

7. *Attention Flags* — A signal from a process program indicating that a particular step has been reached.

8. *Data Integrity* — A condition which may cause loss of data; usually related to Stream 6.

12.3 Some alarm conditions may cause more than one type of alarm to be issued. For example, a parameter control error on over temperature may also trip a protective device that makes the alarm irrecoverable without some intervention. SEMI E30 constrains the definition of an alarm to be only categories 1–3. However this definition does not account for the modular and parallel nature of modern processing equipment.

12.4 *Relationship to ALARM_PAUSED Process State*

12.4.1 The equipment manufacturer should define the relationship between an exception condition and the ALARM_PAUSED process state on a case-by-case basis along these guidelines.

1. An exception is not always an alarm.

2. An exception does not necessarily require a process state change to ALARM_PAUSED even though it may ultimately prove to be the root cause of that transition at a later time.

3. Exceptions that force immediate processing stoppage for the primary function of the tool should trigger a transition to ALARM_PAUSED.

4. Exceptions reported with an alarm code with a category of 1–3 and 5 as defined above should cause a transition to ALARM_PAUSED.

5. A machine with currently set alarms may PAUSE instead of transitioning to ALARM_PAUSED if none of the currently set exception conditions were the immediate trigger for the transition.

SEMI E123.1-0703
SPECIFICATION FOR SECS-II PROTOCOL FOR HANDLER SPECIFIC EQUIPMENT MODEL (HSEM)

This specification was technically approved by the Global Information & Control Committee and is the direct responsibility of the North American Information & Control Committee. Current edition approved by the North American Regional Standards Committee on April 11, 2003. Initially available at www.semi.org May 2003; to be published July 2003.

1 Purpose

1.1 This document maps the services and data of SEMI E123 to SECS-II streams and functions, and data definitions.

2 Scope

2.1 This is a specification covering equipment supporting automated communication of the handler equipment.

2.2 This document applies to all implementations of SEMI E123 that use the SECS-II message protocol (SEMI E5). Compliance to this standard requires compliance to both SEMI E123 and SEMI E5.

NOTICE: This standard does not purport to address safety issues, if any, associated with its use. It is the responsibility of the users of this standard 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 applies to semiconductor equipment that also use SEMI E30 GEM standard.

4 Referenced Standards

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 E37 — High-Speed SECS Message Services (HSMS) Generic Services

SEMI E123 — Standard for Handler Equipment Specific Equipment Model (HSEM)

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

5 Service Message Mapping

5.1 This section shows the specific SECS-II streams and functions that shall be used for SECS-II implementation of the services or remote commands defined in SEMI E123, as well as the parameter mapping for data attached to services.

5.2 *Services Message Mapping*

5.2.1 Table 1 defines the relationships between SEMI E123 services and SECS-II messages. Mapping of service parameters to the SECS-II data items is provided in a separate table. Conventions and definitions of table fields are as described below.

5.2.2 *Service Name* — Name of the service or remote command defined in SEMI E123.

5.2.3 *Stream, Function* — Specifies the SECS-II stream and function (SxFx) mapped to the service messages. Following convention of SECS-II, request and notification messages are mapped to the odd-numbered functions and response or acknowledgement messages are mapped to the corresponding even-numbered functions.

5.2.4 *SECS-II Message Name* — Name of the SECS-II message.

Table 1 Services Message Mapping Table

Service Name	*Stream, Function*	*SECS-II Message Name*
ABORT	S2,F41/F42	Host Command Send/Acknowledge
BIN-UNITS	S2,F49/F50	Enhanced Remote Command Send/Acknowledge
BREAK-CONTACT	S2,F49/F50	Enhanced Remote Command Send/Acknowledge
DISABLE-SITE	S2,F49/F50	Enhanced Remote Command Send/Acknowledge
ENABLE-SITE	S2,F49/F50	Enhanced Remote Command Send/Acknowledge
MAKE-CONTACT	S2,F49/F50	Enhanced Remote Command Send/Acknowledge
PAUSE	S2,F41/F42	Host Command Send/Acknowledge

Service Name	*Stream, Function*	*SECS-II Message Name*
PP-SELECT	S2,F41/F42	Host Command Send/Acknowledge
PRELOAD-UNITS	S2,F41/F42	Host Command Send/Acknowledge
PURGE	S2,F41/F42	Host Command Send/Acknowledge
RECONTACT	S2,F49/F50	Enhanced Remote Command Send/Acknowledge
RESET-TOOL-COUNTS	S2,F49/F50	Enhanced Remote Command Send/Acknowledge
RESUME	S2,F41/F42	Host Command Send/Acknowledge
START	S2,F41/F42	Host Command Send/Acknowledge
STOP	S2,F41/F42	Host Command Send/Acknowledge

5.3 *Event Message Mapping*

5.3.1 Table 2 defines the relationships between SEMI E123 collection events and SECS-II messages. Conventions and table field definitions similar to mapping table for the services, here Event Name specifies the event defined in SEMI E123.

Table 2 Event Message Mapping Table

Event Name	*Stream, Function*	*SECS-II Message Name*
All events defined in SEMI E123	S6F11/12	Event Report Send/Acknowledge

5.4 *Service Parameter Mapping*

5.4.1 Table 3 defines the relationships between SEMI E123 service parameters and SECS-II data definitions or parameter fields in the mapped streams and functions. Parameters for acknowledgements or responses will follow SECS-II specification for that stream and function, no service specific error codes are defined in this specification. Descriptions of each table column are described below.

5.4.2 *Service* — Specifies the service whose parameters are described.

5.4.3 *SECS-II Field* — Specifies the SECS-II data item or message parameter used by the service. In this specification, the DATAID and OBJSPEC fields are unspecified for services using the S2F49 SECS-II message. Their values are to be ignored or used for implementation specific purposes.

5.4.4 *Values* — Value or SECS II format for the specified field. Formats are specified using the SML notation as defined in appendices of SEMI E30 (GEM).

5.4.5 *Req.* — Indicates whether the specified field or parameter is required or not. If an optional CPNAME parameter is used, the corresponding CPVAL must also be given.

5.4.6 *Description* — Provides a brief description of the SECS-II field in relation to the service parameters defined in SEMI E123.

Table 3 Service Parameter to SECS-II Data Items Mapping

Service	*SECS-II Field*	*Values*	*Req.*	*Description*
ABORT	RCMD	"ABORT"	Y	
	$CPNAME_1$	"PROCESSSITEID"	N	ProcessSiteID parameter.
	$CPVAL_1$	U4	N	ProcessSiteID value. ID of handler process site.
	$CPNAME_2$	"CLEANUP"	N	Cleanup parameter.
	$CPVAL_2$	BOOLEAN	N	Cleanup value. If TRUE, remove all materials to outputs.

Service	*SECS-II Field*	*Values*	*Req.*	*Description*
BIN-UNITS	RCMD	"BIN-UNITS"	Y	
	$CPNAME_1$	"BINS"	Y	Bins parameter.
	$CPVAL_1$	L, n 1. U4 2. U4 : n. U4	Y	Bins value. List of bin results for current processed units, one bin value per process site beginning with site 1. Bin values for disabled or empty sites are to be ignored. L, n (n = **total number** of sites) 1. Site 1 Bin Value 2. Site 2 Bin Value 3. … n. Site n Bin Value
BREAK-CONTACT	RCMD	"BREAK-CONTACT"	Y	
	$CPNAME_1$	"SITES"	Y	Sites parameter.
	$CPVAL_1$	L, n 1. U4 2. U4 : n. U4	Y	Sites value. List of sites to break contact. Zero-length list indicates all sites. Ex. L, n (n = number of sites selected) 1. ID of 1st site 2. ID of 2nd site 3. … n. ID of nth site
DISABLE-SITE	RCMD	"DISABLE-SITE"	Y	
	$CPNAME_1$	"SITES"	Y	Sites parameter.
	$CPVAL_1$	L, n 1. U4 2. U4 : n. U4	Y	Sites value. List of sites to disable. Zero-length list indicates all sites.
ENABLE-SITE	RMCD	"ENABLE-SITE"	Y	
	$CPNAME_1$	"SITES"	Y	Sites parameter.
	$CPVAL_1$	L, n 1. U4 2. U4 : n. U4	Y	Sites value. List of sites to enable. Zero-length list indicates all sites.
MAKE-CONTACT	RCMD	"MAKE-CONTACT"	Y	
	$CPNAME_1$	"SITES"	Y	Sites parameter.
	$CPVAL_1$	L, n 1. U4 2. U4 : n. U4	Y	Sites value. List of sites to make contact after a break contact command. Zero-length list indicates all sites.
PAUSE	RCMD	"PAUSE"	Y	
	$CPNAME_1$	"PROCESSSITEID"	N	ProcessSiteID parameter.
	$CPVAL_1$	U4	N	ProcessSiteID value. ID of handler process site.

Service	*SECS-II Field*	*Values*	*Req.*	*Description*
PP-SELECT	RCMD	"PP-SELECT"	Y	
	$CPNAME_1$	"PPID"	Y	ProcessProgramID parameter.
	$CPVAL_1$	A[80]	Y	The ID of the program to be loaded.
	$CPNAME_2$	"PROCESSSITEID"	N	ProcessSiteID parameter.
	$CPVAL_2$	U4	N	ProcessSiteID value. ID of handler process site.
	$CPNAME_3$	"LOTID"	N	LotID parameter.
	$CPVAL_3$	A[1..40]	N	LotID value. ID of lot to be processed with this program. This is an optional parameter that has implementation specific meanings.
PRELOAD-UNITS	RCMD	"PRELOAD-UNITS"	Y	
PURGE	RCMD	"PURGE"	Y	
RECONTACT	RCMD	"RECONTACT"	Y	
	$CPNAME_1$	"SITES"	Y	Sites parameter.
	$CPVAL_1$	L, n 1. U4 2. U4 : n. U4	Y	Sites value. List of sites to re-contact. Zero-length list indicates all sites.
RESET-TOOL-COUNTS	RCMD	"RESET-TOOL-COUNTS"	Y	
	$CPNAME_1$	"SVIDLIST"	Y	StatusVariableList parameter.
	$CPVAL_1$	L, n 1. U4 2. U4 : n. U4	Y	StatusVariableList value. List of status variables to reset, list by SVID.
	$CPNAME_2$	"PROCESSSITEID"	N	ProcessSiteID parameter.
	$CPVAL_2$	U4	N	ProcessSiteID value. ID of handler process site.
RESUME	RCMD	"RESUME"	Y	
	$CPNAME_1$	"PROCESSSITEID"	N	ProcessSiteID parameter.
	$CPVAL_1$	U4	N	ProcessSiteID value. ID of handler process site.
START	RCMD	"START"	Y	
	$CPNAME_1$	"PROCESSSITEID"	N	ProcessSiteID parameter.
	$CPVAL_1$	U4	N	ProcessSiteID value. ID of handler process site.
STOP	RCMD	"STOP"	Y	
	$CPNAME_1$	"PROCESSSITEID"	N	ProcessSiteID parameter.
	$CPVAL_1$	U4	N	ProcessSiteID value. ID of handler process site.
	$CPNAME_2$	"CLOSELOT"	N	CloseLot parameter.
	$CPVAL_2$	BOOLEAN	N	CloseLot value. This is an optional parameter that has implementation specific meanings.

6 Variable Data Item Mapping

The purpose of this section is to define the list of variable items required by the HSEM. Values of these variables will be available to the host through collection event reports and host status queries.

6.1 *Requirements*

6.1.1 Any supplier-defined variables shall be documented in the same format used by this document. The following minimum information is required:

<variable name> **Class:** <ECV, SV, or DVVAL> **Format:** <SML>

Description: <If class = DVVAL, description must contain statement of when data is valid.>

<If format = ASCII, then a length is required. It is assumed to be left-justified unless otherwise noted.>

6.2 *Data Types*

6.2.1 Equipment Constants (ECVs) can be changed by the host using S2,F15 (New Equipment Constant Send). The operator may be able to change some values, but the equipment does not change the values on its own. The value of an equipment constant may be queried by the host at any time, using the S2,F13/14 transaction. They reside in non-volatile memory of the equipment. Equipment constants remain in effect until they are overwritten either by manual entry or by a S2,F15.

6.2.2 Status Variables (SVs) are valid at all times. An SV may not be changed by the host but may be changed by the equipment or operator. The value of status variables may be queried by the host at anytime using the S1,F3/4 or S6,F19/20 transactions.

6.2.3 DVVALs are variables that are valid only upon the occurrence of specific collection events. An attempt to read a variable item at the wrong time does not generate an error, but the data reported may not have relevant meaning.

6.3 For multiple sites scenarios, if a variable is described as List by number of sites, then a list is expected, one value for each site.

Table 4 Variable Item Mapping Table

Variable Name	*Description*	*Class*	*SECS-II Type*
	Physical Handler Group		
BufferID	Identifier of the buffer. This is implementation specific depending if handler supports any input, sort or output buffers.	DVVAL	A[1..16]
BufferType	Supplier defined buffer type. This is implementation specific, and may include input, output or sort buffers.	DVVAL	Enumerated value for each buffer type.
CategoryID	List of sort category IDs currently configured in handler.	ECV	L, n 1. A[1..16] 2. A[1..16] : n. A[1..16]
CategoryCount	Count of units in each sort category. List by number of categories. L,n # n=number of categories L, 2 1. CategoryID 2. Count of units in this category	SV	L, n L,2 1. A[1..16] 2. U4
EquipSerialID	Identification of Equipment.	SV	A[1..16]
KitID	ID of unique tooling unit.	SV	A[1..24]
LightPoleStatus	Color/status (i.e., Red/flash).	SV	A[1..16]

Variable Name	*Description*	*Class*	*SECS-II Type*
LinkPortStatus	(3 = Input/Output linked, 2 = Input linked, 1 = Output linked, 0 = HANDLER not Linked)	SV	U4, enumerated values as indicated.
MediaID	Media Serial Number.	SV	A[1..24]
MediaType	Name of Media Type.	SV	A[1..24]
OperationType	Current Operation Mode (i.e., maintenance, production).	ECV	A[1..24]
OperatorID	Current Operator ID.	ECV	A[1..24]
QueueStatus	PPID Queued to be run.	SV	U4
ReaderType	Type of reader installed on handler.	DVVAL	A[1..24]
ReaderErrorType	Type of error detected by the material reader.	DVVAL	A[1..24]
Process-Site Components			
AlignmentCount	Number of units since last alignment (i.e., Homing/Adjustment).	SV	U4
InsertionForce	Insertion-Force energy.	DVVAL	F8
InsertionForceSetpoint	Insertion-Force set point (setpoint).	ECV	F8
MediaChangeTime	Elapsed time to replace media and send ready.	SV	A[16]
MediaCount	Number of media since last reset.	SV	U4
MediaCountInterval	Event generated when number of media is completed.	SV	U4
PresentPositionActual	Present position actual.	ECV	U4
PresentPositionSetpoint	Present position set points (setpoint).	ECV	U4
ProcessSiteTemp	Process-site temperature (setpoint).	SV	F8
ProcessSiteID	ID of process-site in configuration.	SV	U4
ProcessSiteStatus	$Site_n$ Availability (1 = enabled, 0 = disabled). List by number of sites.	DVVAL	L, n # n=number of sites 1. U4 2. U4 : n. U4
ProcessSiteCount	Total count of process sites.	SV	U4
ProcessSiteLoaded	$Site_n$ loaded status, normally associated with UnitsReady event. 0=No unit, 1=Loaded. List by number of sites. L, n # n=number of sites 1. Loaded status of Site 1 2. Loaded status of Site 2 : n. Loaded status of Site n	SV	L, n # n=number of sites 1. U4 2. U4 : n. U4
SkipCount	Number of units skipped since last reset (Skip + Process = Total).	SV	U4
SkipCountInterval	Event generated when number of units is skipped.	SV	U4
StartProcessPortID	Start process source (i.e., hand, keyboard, host).	SV	U4
UnitCount	Number of units since last reset.	SV	U4
UnitCountInterval	Event generated when number of units is completed.	SV	U4
UnitPosition	X, Y, Z media location of a unit.	SV	U4
UnitStatus	(1 = Processed, 0 = Skipped)	SV	U4, enumerated values as indicated.

7 Additional Alarm Management Requirements

7.1 *Alarm Definition*

7.1.1 SEMI E30 has a very restrictive definition of alarms conflicting with the SEMI E5. This standard requires the broader SEMI E5 definition.

7.1.2 Messages S5,F1 through S5,F8 of this section provide basic alarm messages. The messages S5,F9 through S5,F18 provide extended capabilities for Exception Handling. When using messages F1–F8, alarms may be divided into categories as follows:

Table 5 Alarm Code Mapping

Alarm Name	*ALCD*	*Description*
Personal Safety	1	Condition may be dangerous to people.
Equipment Safety	2	Condition may harm equipment.
Parameter Control Warning	3	Parameter variation outside of preset limits — may harm product.
Parameter Control Error	4	Parameter variation outside of reasonable control limits — may indicate an equipment malfunction.
Irrecoverable Error	5	Intervention required before normal use of equipment can resume.
Equipment Status Warning	6	An unexpected condition has occurred, but operation can continue.
Attention Flags	7	A signal from a process program indicating that a particular step has been reached.
Data Integrity	8	A condition that may cause loss of data; usually related to Stream 6.

7.1.3 For messages F1 through F8, it will be the equipment's responsibility to categorize the alarm. Some alarm conditions may cause more than one type of alarm to be issued. For example, a parameter control error on over temperature may also trip a protective device that makes the alarm irrecoverable without some intervention.

- Exceptions reported with an ALCD with a category of 1–3 and 5 should cause a transition to ALARM_PAUSED.

7.2 *Recoverable Exceptions*

7.2.1 Any exception that may be programmatically recoverable should be reported as an exception using ***S5F9*** Exception Post Notify (EXPN) with a minimum recovery action of RETRY.

7.2.2 If a machine reports recoverable exceptions it should implement the other ***S5*** functions for Exception Clear, Recovery Request, and Recovery Request Abort.

RELATED INFORMATION 1

NOTICE: This related information is not an official part of SEMI E123.1 and was derived from the work of the originating committee. This related information was approved for publication by full letter ballot procedures.

R1-1 Scenarios

R1-1.1 The purpose of this section is to document possible HSEM-specific operational scenarios.

R1-1.2 *Normal Run Scenario*

R1-1.2.1 This is an error-free run of a single lot with a single test-site. All optional SEMI E30 events are turned off by default.

COMMENT	HOST	EQUIPMENT		COMMENT
Host selects process program.	S2,F41-->			
		<--S2,F42	Equipment Ack	
Setting Up				
Setup Complete				
		<--S6,F11	Event : PPLoadOk	
Host Ack	S6,F12-->			
Process				
Process.SettingUp				
Process.Ready				
Host commands start	S2,F41-->			
		<--S2,F42	Equipment Acks Start.	
			Handler cycles devices... ...until empty...	
		<--S6,F11	Event : Empty	
Host Acks Event	S6,F12-->			
Host commands Purge.	S2,F41-->			
		<--S2,F42	Handler acks Rmt.Cmd.	
Stopping				
		<--S6,F11	Event : Lot Completed	
Host Acks Event	S6,F12-->			
Idle				

R1-1.3 *Normal SPC Scenario*

R1-1.3.1 This is a normal SPC run with all optional SEMI E30 events turned off by default.

COMMENT	HOST	EQUIPMENT	COMMENT
Host selects GEM Alarms to enable (list).	S5,F3-->		
		<--S5,F4	Alarms xyz ON
Host selects GEM Events to enable (list).	S2,F37-->		
		<--S2,F38	Events xyz ON
Host selects process program.	S2,F41-->		
		<--S2,F42	Equipment Ack
Setting Up			
Setup Complete			
		<--S6,F11	Event : PPLoadOk
Host Ack	S6,F12-->		
Process			
Process.SettingUp			
Process.Ready			
Host requests start-of-lot report.	S6,F15-->		
		<--S6,F16	Equipment sends report items.
Host commands start	S2,F41-->		
		<--S2,F42	Equipment Acks Start.
		<--S5,F1	Alarm : NoDevicesPresent
Host Acks Alarm	S5,F2-->		
		Time Passes.	
		<--S6,F11	Event : PortLoaded
Host Acks Event	S6,F12-->		
			Handler cycles devices... ... until ...
		<--S5,F1	Alarm : LoadDeviceFail

Host Acks Alarm	S5,F2-->		
ProcessPause			
Host sends operator to clear jam.			
Host sends resume.	S2,F41-->		
		<--S2,F42	Handler Acks and resumes.
Processing			
			Handler cycles devices... ... until ...
Host asks for Temperature x.	S6,F15-->		
		<--S6,F16	Handler sends Temp.x.
			...
		<--S6,F11	Event : HandlerEmpty
Host Acks Event	S6,F12-->		
Host commands Stop.	S2,F41-->		
		<--S2,F42	Handler acks Rmt.Cmd.
Stopping			
		<--S6,F11	Event : Lot Completed
Host Acks Event	S6,F12-->		
Idle			
Host requests end-of-lot-report.	S6,F15-->		
		<--S6,F16	Handler sends report.

R1-1.4 *Multi-Site Run Scenario*

R1-1.4.1 This is a run scenario with 64 test-sites and optional GEM events all turned on with no errors of any type occurring.

COMMENT	HOST	EQUIPMENT	COMMENT
Host selects GEM Alarms to enable (list).	S5,F3-->		
		<--S5,F54	Alarms xyz ON.
Host selects GEM Events to enable (list).	S2,F37-->		
		<--S2,F38	Events xyz ON.
Host selects Trace Data Item(s).	S2,F23-->		
		<--S2,F24	Trace Data Item x ON.
Host selects process program.	S2,F41-->		
		<--S2,F42	Equipment Ack
Setting Up			
Setup Complete			
		<--S6,F11	Event : PPLoadOk
Host Ack	S6,F12-->		
Process			
Process.SettingUp			
Process.Ready			
Host requests start-of-lot report.	S2,F41-->		
		<--S2,F42	Equip sends report items.
Host commands start-of-lot.	S2,F41-->		
		<--S2,F42	Equip Acks Start. ... for each trace item period.
		<--S6,F1	Trace Event x SEND.
Host receives and Acks Trace Item x.	S6,F2-->		
			... for each port event

```
                                                    (tray/tube).

                                  <--S6,F11       Event  :  PortLoaded

  Host Acks Event          S6,F12-->

                                                ... Handler cycles devices ...
                                                ... for each device report
                                                    internal states.

                                  <--S6,F11       Event : DeviceClearsInput

  Host Acks Event          S6,F12-->

                                  <--S6,F11       Event : DeviceEntersTemp

  Host Acks Event          S6,F12-->

                                  <--S6,F11       Event : DeviceClearsTemp

  Host Acks Event          S6,F12-->

                                  <--S6,F11       Event : DeviceEntersQueue.x

  Host Acks Event          S6,F12-->

                                  <--S6,F11       Event : StartTest.Contactor.x

  Host Acks Event          S6,F12-->

                                  <--S6,F11       Event : EndTestReceived.x

  Host Acks Event          S6,F12-->

                                  <--S6,F11       Event : DeviceBinReceived.x

  Host Acks Event          S6,F12-->

                                  <--S6,F11       Event : DeviceClearsContactor.x

  Host Acks Event          S6,F12-->

                                  <--S6,F11       Event:DeviceEntersUnloadQueue.x

  Host Acks Event          S6,F12-->

                                  <--S6,F11       Event : DeviceUnloaded.HardbinX

  Host Acks Event          S6,F12-->

                                              ... for each full/empty tray/tube on
                                                  input,output.

                                  <--S5,F1     Alarm : ContainerReplaceRequest.x

  Host Acks request.       S5,F2-->
```

Note that the messages reporting the above internal states may require sub-addressing of test-sites and ports similar to the tester SEM.

```
                                                    Handler cycles devices...
                                                    ... random messages
                                                    (e.g., ... variable request)

Host asks for variable    S6,F15-->
x (devices tested).

                                    <--S6,F16       Handler sends variable.
                                                    ...
                                                    ... eventually ends

                                    <--S6,F11       Event  :  HandlerEmpty

Host Acks Event           S6,F12-->

Host commands             S2,F41-->
end-of-lot.

                                    <--S2,F42       Handler acks Rmt.Cmd

Stopping

                                    <--S6,F11       Event  :  Lot

Completed

Host Acks Event           S6,F12-->

Idle

Host requests             S6,F15-->
end-of-lot report.

                                    <--S6,F16       Handler sends report.
```

R1-1.5 *Continuous Load Scenario*

R1-1.5.1 This is a run scenario where there are more units that need to be processed with same recipe and configuration but were not loaded to Handler inputs initially (due to limited input space, material delays etc.) This scenario assumes no direct link between Test Equipment and Handler, Host directs all run-to-run processing between the two equipments. Host and Station Controller represent the same entity here.

Test Equipment		**Host (Station Control)**		**Handler**
		Host waits for UnitsReady event…		
			⇦ S6,F11	Event: UnitsReady
		ACK: Event S6,F12	⇨	
	⇦	Start Processing on Test Equipment.		*Waiting for command…*
Testing units…				
Event: Test Complete (w/ bin list for multiple test sites).	⇨			

Test Equipment	Host (Station Control)			Handler	
	RCMD=BIN-UNITS (with bin list for multiple sites)	S2,F49	⇨		
			⇦	S2,F50	ACK: BIN-UNITS *Unload units …(UNLOAD state)* *Load next batch of unts… (LOAD state)*
	Repeat… until last unit is completed				
			⇦	S6,F11	Event: LastUnitCompleted *(no more units in all inputs and buffers).* *Handler transitions to READY state.*
	ACK: Event	S6,F12	⇨		
					Host/Operator determines current lot has more units to process, loads remainder of lot onto Handler Inputs.
	RCMD=START (Host initiated resume after load is complete.)	S2,F41	⇨		
			⇦	S2,F42	ACK: START *Alternatively Operator can resume directly from Handler after loading of units. Next event to Host would then be "UnitsReady" event.*
	Processing Repeats from top again until all units for lot is processed …				
			⇦	S6,F11	Event: LastUnitCompleted *Handler again transitions to READY state per above.*
	ACK: Event *Host/operator determines lot has completed.*	S6,F12	⇨		
	RCMD=STOP	S2,F41	⇨		
			⇦	S2,F42	ACK: STOP *Handler transitions to STOPPING state and ultimately to IDLE state.*

NOTICE: SEMI makes no warranties or representations as to the suitability of the standards set forth herein for any particular application. The determination of the suitability of the standard is solely the responsibility of the user. Users are cautioned to refer to manufacturer' s instructions, product labels, product data sheets, and other relevant literature, respecting any materials or equipment mentioned herein. These standards are subject to change without notice.

By publication of this standard, Semiconductor Equipment and Materials International (SEMI) takes no position respecting the validity of any patent rights or copyrights asserted in connection with any items mentioned in this standard. Users of this standard are expressly advised that determination of any such patent rights or copyrights, and the risk of infringement of such rights are entirely their own responsibility.

SEMI E125-0305
SPECIFICATION FOR EQUIPMENT SELF DESCRIPTION (EqSD)

This specification was technically approved by the Global Information & Control Committee and is the direct responsibility of the North American Information & Control Committee. Current edition approved by the North American Regional Standards Committee on August 16, 2004. Initially available at www.semi.org September 2004; to be published November 2004. Originally published July 2003; previously published July 2004.

NOTICE: The designation of SEMI E125 was updated during the 0305 publishing cycle to reflect the creation of SEMI E125.1.

Contents

1 Purpose

1.1 *Specification Purpose*

1.1.1 This specification describes a method for allowing equipment suppliers to provide a description of the variables, events, exceptions, and physical equipment configuration available from their equipment. With this information available for consumption by software systems, it can be used as a tool to aid the process of integrating equipment into a factory's automation system.

2 Scope

2.1 *In-scope*

2.1.1 This document specifies the classes that suppliers are to use to describe essential data, events, and exceptions provided by their equipment. The specification only describes information that is static in nature (that is, information that does not change dynamically while the equipment is running). This document also specifies an interface that clients can use to access this information.

2.1.2 This specification applies to all semiconductor manufacturing equipment that supports the data acquisition interface defined in the SEMI specification for Data Collection Management.

2.2 *Out of Scope*

2.2.1 This specification does not define any new behavior required of the equipment other than that necessary for retrieving information describing equipment configuration, interfaces, and available data, and keeping this information current.

2.2.2 The details of any underlying concepts and behavioral models (for example, carrier management, process/control job, etc.) that can be described by metadata are to be separately specified in a document dedicated to those concepts. Only the ability to describe the fact that a supplier has implemented such a concept and that a client can discover this implementation and any data it can produce is in scope for this specification.

2.2.3 This specification does not require that the metadata provided by the equipment be directly human readable. It is expected that applications will be written to organize and present this information to human users in a form that is easier for end users to digest.

NOTICE: This standard does not purport to address safety issues, if any, associated with its use. It is the responsibility of the users of this standard to establish appropriate safety and health practices and determine the applicability of regulatory or other limitations prior to use.

3 Limitations

3.1 *Abstract Model*

3.1.1 This specification is an abstract model only. Adjunct specifications must be developed to bind this specification to an implementation technology.

3.2 *Other Limitations*

3.2.1 This specification does not define the mechanism used for determining how to locate clients that are to receive notifications defined by this specification. It assumes that the equipment has established a communications context with any clients that can submit requests, and has provided any context necessary to send notifications to those clients.

3.2.2 This specification does not define the mechanism used for identifying or authenticating clients. It assumes that clients can be uniquely identified, and that the equipment is capable of distinguishing requests originating from different clients.

4 Referenced Standards

4.1 *SEMI Standards*

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

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

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

4.2 *ISO Standards*[1]

ISO 639:1988 — Code for the representation of names of languages

ISO 3166-1:1997 — Codes for the representation of names of countries and their subdivisions – Part 1: Country codes

ISO 8601 — Representations of dates and times, 1988-06-15

ISO 8601 Draft Revision — Representations of dates and times, draft revision, 2000

4.3 *Other Standards*

American National Standards Institute (ANSI) X3.135-1992 – Information Systems – Database Language – SQL (http://www.ansi.org) [2]

Internet Engineering Task Force (IETF) RFC 1776 – Tags for the Identification of Languages, ed. H. Alvestrand. 1995 (http://www.ietf.org/rfc/rfc1766.txt)

Unified Modeling Language (UML) Specification, Version 1.4, OMG Specification 01-09-67, available from http://www.omg.org/technology/documents/modeling_spec_catalog.htm

Uniform Resource Name (URN) Syntax, IETF RFC 2141, May 1997 (http://www.ietf.org/rfc/rfc2141.txt)

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

5 Terminology

5.1 *Abbreviations and Acronyms*

5.1.1 Descriptions of many of the abbreviations and acronyms used in this specification may be found in the SEMI Compilation of Terms, available on the SEMI web site, http://www.semi.org/. In most cases, these terms are not included in this section.

5.1.2 *IDL* — Interface Definition Language

5.1.3 *OMG* — Object Management Group

5.1.4 *SQL* — Standard Query Language

5.1.5 *UML* — Unified Modeling Language

5.1.6 *URN* — Uniform Resource Name

5.1.7 *XML* — eXtensible Markup Language

5.2 *Definitions*

5.2.1 Definitions or descriptions of many of the terms used in this specification may be found in the SEMI Compilation of Terms, available on the SEMI web site, http://www.semi.org/. In most cases, these terms are not included in this section.

5.2.2 Related Information 8 contains useful definitions of UML terms taken directly from the UML standard. UML terms used in this document conform to these definitions. Please refer to this section as needed.

5.2.3 *metadata* — data used to describe data. For example, if a tool can report an event with several associated variables under certain conditions, the metadata for that event would provide a description of what condition will produce the event, what the type and units are of each variable, and the id of the event itself.

---

1 International Organization for Standardization, ISO Central Secretariat, 1, rue de Varembé, Case postale 56, CH-1211 Geneva 20, Switzerland. Telephone: 41.22.749.01.11; Fax: 41.22.733.34.30, Website: /www.iso.ch

2 American National Standards Institute, Headquarters: 1819 L Street, NW, Washington, DC 20036, USA. Telephone: 202.293.8020; Fax: 202.293.9287, New York Office: 11 West 42nd Street, New York, NY 10036, USA. Telephone: 212.642.4900; Fax: 212.398.0023, Website: www.ansi.org

6 Conventions

6.1 *Document and Notation Conventions*

6.1.1 *Unified Modeling Language* (UML) — This specification uses the UML conventions for representing all class, state, object, and sequence diagrams. Refer to the UML specification for a detailed description of these conventions.

6.1.2 *Attribute Tables* — The table below provides an example of the tables used to list and describe attributes of classes defined in this specification.

Table 1 Attribute Table Format

Attribute Name	*Definition*	*Required*	*Form*
		Y or N	**See list below**

6.1.2.1 *Form* — Defines the data type of the attribute. The terms used to describe data types in this column are defined in the SEMI Compilation of Terms, or are included as part of the specification. Refer to the compilation of terms for the definition of SEMI type name meanings.

6.1.3 *UML Associations* — The mechanism used for representing UML associations between classes is implementation dependent. This document is abstract in nature, and does not specify or imply any such mechanism. Any adjunct standard that provides an implementation of this specification must include a description of the mechanism used for representing the UML associations shown in this document.

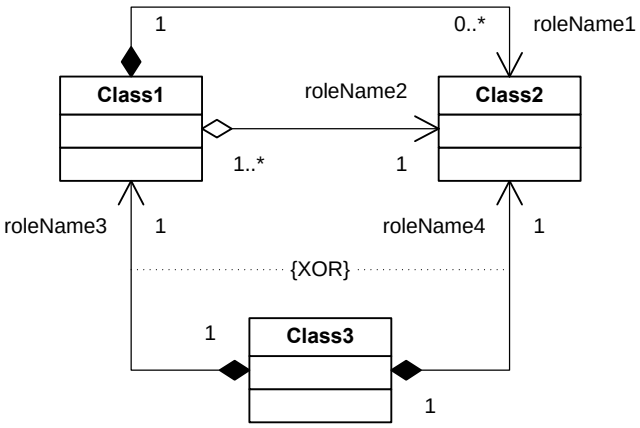

Figure 1
Association Notation

6.1.3.1 *Association Notation* — This document uses the UML aggregation diamond adornment, role names, end multiplicities, and navigability in all class diagrams specifying associations. Unadorned associations are not used, for economy of notation. See the UML 1.4 specification, Section 3.43 for details. All role names are public by default; therefore the visibility symbols are not used.

6.1.3.1.1 Open diamond adornments indicate that instances of the target class may be shared among aggregate classes. Closed diamond adornments indicate that target instances belong to at most one composite class. The part-whole semantics of the aggregation/composition symbol is not significant in this specification. See the UML 1.4 specification, sections 3.43.2.5 and 3.48 for further information.

6.1.3.1.2 Some diagrams use the UML "XOR" constraint where applicable. An "XOR" constraint indicates that only one of many possible associations can be instantiated at any one time for a given instance of the association class. See the UML 1.4 specification, Section 3.42.5.1 for further information.

6.1.4 *Association Tables* — The table below provides an example of the tables used to list and describe associations between classes defined in this specification.

Table 2 Association Table Format

Association Role Name	*Definition*	*Comments*

6.1.4.1 *Association Role Name* — The name of the association role being specified.

6.1.4.2 *Definition* — Describes the function or purpose of the association.

6.1.4.3 *Comments* — Any additional comments or notes regarding the association.

6.1.5 *Operation Definition Tables* — The table below provides an example of the tables used to list and describe the interface operations defined in this specification.

Table 3 Operation Definition

Operation	*Description*	*Type*
		See list below

6.1.5.1 *Operation* — Specifies the name of the operation.

6.1.5.2 *Type* — Specifies the messaging semantics of the operation. Only Request-Reply (RR) semantics are used in this specification. Request-Reply messages are messages that involve an initiator and a receiver. In a Request-Reply exchange, the initiator sends a single request message to the receiver, and the receiver sends a single reply message to that request back to the initiator. Fire-and-Forget messages are messages that involve a sender and a receiver. In a Fire-and-Forget exchange, the sender sends a single message to the receiver, with no associated response.

6.1.6 *Operation Argument Definition Table* — The table below provides an example of the tables used to list and describe arguments for interface operations defined in this specification.

Table 4 Operation Argument Definitions

Argument	*Description*	*Kind*	*Form*
		See list below	**See list below**

6.1.6.1 *Argument* — Specifies the name of the argument.

6.1.6.2 *Kind* — Specifies whether the argument is an ‘in’, ‘out’, or ‘error’ argument for the operation. ‘Error’ arguments always function as ‘out’ arguments, but indicate that the operation did not complete successfully.

6.1.6.3 *Form* — Defines the data type of the argument. The terms used to describe data types in this column are defined in the SEMI Compilation of Terms, or are included as part of the specification. Refer to the compilation of terms for the definition of SEMI type name meanings.

6.2 *Terms*

6.2.1 *Class, type* — This specification uses the terms “class” and “type”, interchangeably to refer to a UML class.

7 Background

7.1 *Motivation*

7.1.1 This specification is designed to help address the problem of integrating semiconductor equipment from a variety of suppliers into an automated factory environment, and to provide a standard mechanism for describing information and data that the equipment can provide for a wide variety of uses. See Related Information 1 for an overview of some example use cases.

7.1.2 This specification defines the information that is needed to communicate essential features of the equipment to software applications, and provides extensibility features to allow suppliers to communicate information unique to their equipment or that may not be specified in a SEMI standard.

7.1.3 Equipment metadata is designed to be useful to applications that provide an interface to human end users for the purposes of understanding and browsing equipment configuration, available data items, events, exceptions, etc. In many cases, these users will be referring to equipment metadata so they can perform data collection for a variety of objectives (diagnostics, troubleshooting, process control, etc.).

7.1.4 Equipment metadata is also designed to be useful to applications written to take advantage of standardized state models and other data so that discovery and usage of this information can be automated, without requiring human intervention. Diagnostics, health monitoring, or other supplier-provided applications that understand the details of specific equipment types, for example, can also discover and use common equipment-specific configuration and data without requiring manual assistance.

8 Overview

8.1 *Equipment Metadata Organization*

8.1.1 Figure 2 shows the concepts that are included in this specification and their relationship to the SEMI E120 Common Equipment Model. A brief description of the purpose of each metadata concept is provided in this section only as an overview. For the detailed specification of the corresponding classes, refer to Section 10.

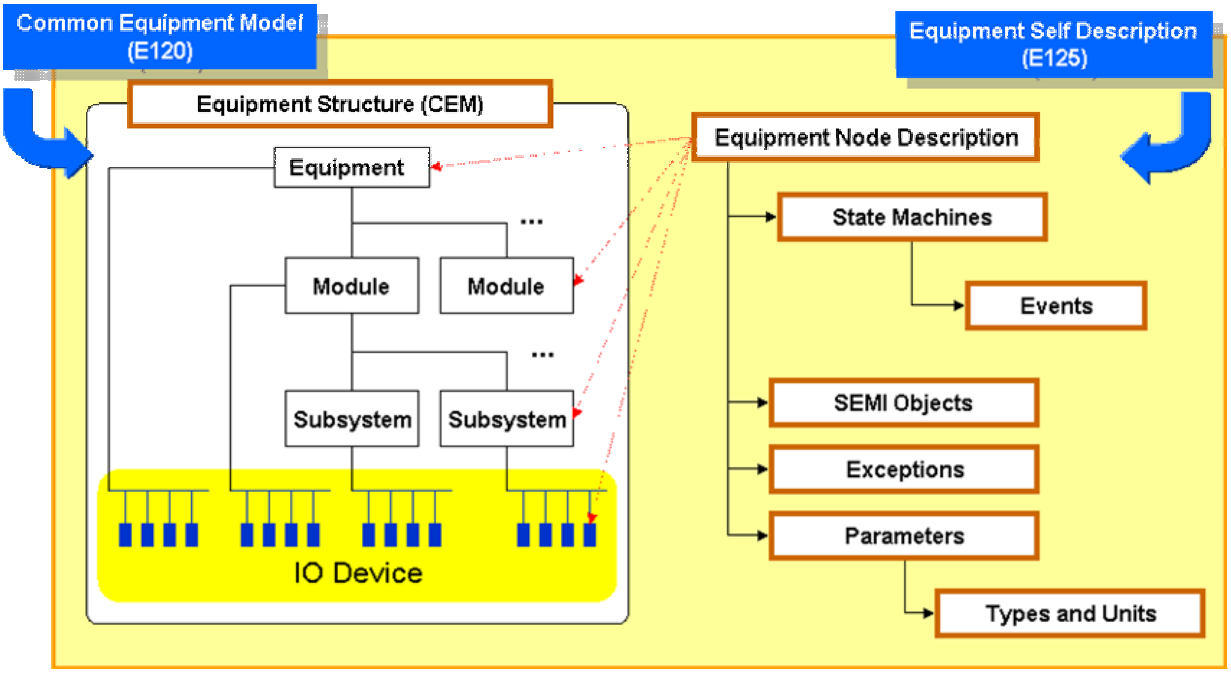

Figure 2
Organization of Metadata Concept

8.1.2 *Equipment Node Description*

8.1.2.1 This specification defines classes that provide a way to associate events, exceptions, parameters, and SEMI E39 ObjTypes with any node in the physical equipment structure. SEMI E120 defines the classes that are used to describe the physical equipment structure; this specification defines the classes that link E120 equipment nodes with the events, exceptions, parameters, and ObjTypes that they produce. Equipment node description classes are defined in Section 10.3 .

8.1.3 *Parameters*

8.1.3.1 This specification defines classes that can be used to describe any variables that are provided by the equipment, and refers to all such variables as “Parameters”. Parameters can be used to model any concept that can be represented as a data type. For example, the SEMI E30 concepts of equipment constants, data variables, and status variables can all be represented by a parameter. Any type of data that the equipment can produce (for example, measurement results on metrology equipment) can be represented by a Parameter. Parameters can only be

defined by an equipment node, an SEMI E39 ObjType, or an exception. Parameter description is defined in Section 10.4 .

8.1.4 *Type and Units*

8.1.4.1 This specification provides a way to describe the data types and units used on the equipment to represent Parameters. For example, if the equipment defines several Parameters that have a structured data type consisting of an integer, a string, and an array of Booleans, this specification provides classes that can be used to describe the structure of that data type. For numeric data types, this specification defines classes that can be used to describe the units that a value of that data type has. For example, if some Parameter represents a capacitance in microfarads, this specification can be used to describe the microfarad unit itself, as well as how to indicate that the parameter's value is in units of microfarads. Parameter type description is defined in Section 10.4.11 , and units description is defined in Section 10.5.5 .

8.1.5 *State Machine and Events*

8.1.5.1 This specification defines classes that can be used to describe the state machines (and the corresponding transitions and events) that are implemented by the equipment. These classes can be used to describe any SEMI-defined state machine or any supplier-defined state machine. This specification also defines classes that can be used to describe the Parameters that are available for reporting when a specific event occurs. State machine description is defined in Section 10.8 .

8.1.6 *SEMI Objects*

8.1.6.1 This specification defines classes that can be used to describe any SEMI E39 ObjTypes that are implemented by the equipment. SEMI ObjType state machines and events are described using the same classes that are used to describe state machines. ObjType attributes are described using the same classes that are used to describe Parameters and types. Description of ObjType services is not supported. SEMI ObjType description is defined in Section 10.8 .

8.1.7 *Exceptions*

8.1.7.1 his specification defines classes that can be used to describe the existence of error conditions that the equipment can detect and report as exceptions. An exception can represent any SEMI standard E30 alarm or equipment-supplier-defined exception that can be reported to clients. Exception description is defined in Section 10.7 .

8.2 *Accessing Equipment Metadata*

8.2.1 Equipment metadata is directly associated with each unique equipment installation within a factory. Access to metadata is provided by a collection of operations available from a specific equipment installation. This interface is described in Section 9.1 .

9 Interfaces for Managing Equipment Metadata

9.1 *Equipment Metadata Manager*

9.1.1 Figure 3 shows the interface used to access equipment metadata descriptions.

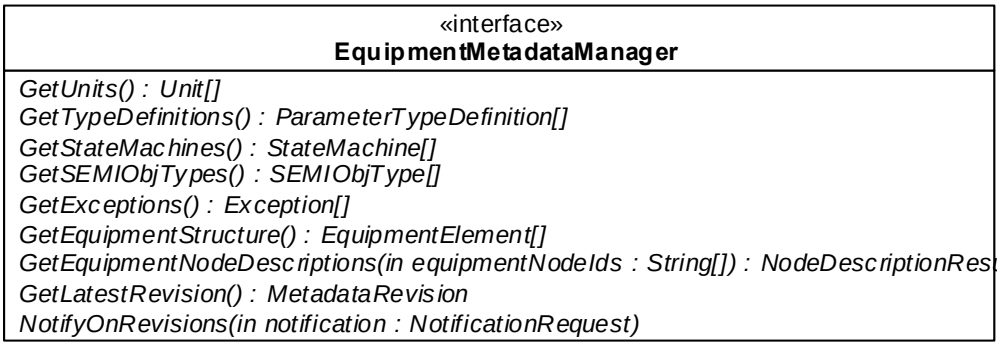

Figure 3
Interface for Accessing Equipment Metadata

9.1.2 *EquipmentMetadataManager Operations*

Table 5 EquipmentMetadataManager Operation Definition

Operation	*Description*	*Type*
GetUnits	Retrieves all unit metadata provided by the equipment.	RR
GetTypeDefinitions	Retrieves all type definitions provided by the equipment.	RR
GetStateMachines	Retrieves all state machine metadata provided by the equipment.	RR
GetSEMIObjTypes	Retrieves all SEMI ObjTypes metadata provided by the equipment.	RR
GetExceptions	Retrieves all exception metadata provided by the equipment.	RR
GetEquipmentStructure	Retrieves all equipment structural metadata provided by the equipment.	RR
GetEquipmentNodeDescriptions	Retrieves all requested equipment node description metadata.	RR
GetLatestRevision	Retrieve the last date and time at which the equipment metadata was revised	RR
NotifyOnRevisions	Request that the equipment notify the requestor when changes to the metadata are made	RR

9.1.2.1 *GetUnits* — Upon receiving this request, the equipment shall return all units metadata provided by the equipment.

9.1.2.1.1 *GetUnits Operation Arguments*

Table 6 GetUnits Argument Definitions

Argument	*Description*	*Kind*	*Form*
units	All units defined for the equipment.	out	Unordered list of elements of type Unit, described in Section 10.6 .

9.1.2.2 *GetTypesDefinitions* — Upon receiving this request, the equipment shall return all type definition metadata provided by the equipment.

9.1.2.2.1 *GetTypesDefinitions Operation Arguments*

Table 7 GetTypesDefinitions Argument Definitions

Argument	*Description*	*Kind*	*Form*
typeDefinitions	All type definitions provided by the equipment.	out	Unordered list of elements of type ParameterTypeDefinition, described in Section 10.5 .

9.1.2.3 *GetStateMachines* — Upon receiving this request, the equipment shall return all state machine metadata provided by the equipment.