SEMI E134-0710
SPECIFICATION FOR DATA COLLECTION MANAGEMENT

This Standard was technically approved by the Information & Control Global Technical Committee. This edition was approved for publication by the global Audits & Reviews Subcommittee on April 30, 2010. Initially available at www.semi.org in June 2010. Originally published July 2004; previously published March 2010.

Table of Contents

1 Purpose 2
2 Scope 2
3 Limitations 2
3.1 *Abstract Model* 2
4 Referenced Standards and Documents 2
4.1 *SEMI Standards* 2
4.2 *ISO Standards* 3
4.3 *OMG Standard* 3
5 Terminology 3
5.1 *Abbreviations and Acronyms* 3
5.2 *Definitions* 3
6 Conventions 4
6.1 *Terms* 4
6.2 *Notation* 4
7 Background 6
7.1 *Equipment Data Collection* 6
8 Overview 8
8.1 *Data Collection Concepts* 8
8.2 *Trace Data* 8
8.3 *Events* 8
8.4 *Exceptions* 9
8.5 *Data Collection Plans* 9
8.6 *Equipment Operational Performance Monitoring* 9
8.7 *Ad-hoc Data Requests* 9
9 Data Collection Plan Management 10
9.1 *DataCollectionManager Interface* 10
10 DCP Privilege Model 29
10.1 *Privileges* 29
11 Data Collection Plan Definition 31
11.1 *DataCollectionPlan* 31
12 Data Collection Plan State Models 41
12.1 *DataCollectionPlan State Model* 41
12.2 *DataCollectionPlan-Active State Model* 43
12.3 *Trace Request State Model* 46
13 Operational Performance Monitoring 49
13.1 *Operational Performance Conditions State Model* 49
14 Data Collection Report Formats 51
14.1 *Data Collection Reports* 51
14.2 *Time Format* 56
15 DCP Consumer Interface 56
15.2 *DCPConsumer* 57
16 Requirements for Compliance 61
Related Information 1 62
Related Information 2 65

Documents Document Sharing Hub
Full document details: SEMI E134-0710

1 Purpose

1.1 This specification describes a method for data acquisition consumers to request process and operational data from equipment to be communicated in an automated fashion by the equipment, or in an ad-hoc request from the consumer. It includes a mechanism for organizing related data into groups to make it more straightforward for enabling or disabling a large number of data sources, and to allow consumers to organize related data into groups according to their purpose.

2 Scope

2.1 This specification provides a means to acquire event, exception, and trace data from semiconductor equipment through the use of a named data collection plan. This specification defines what form a data collection plan takes, the meaning of its contents, and an interface for managing them that is to be supported by the equipment.

2.2 This specification defines the behavior associated with the execution of data collection plans in the form of finite state machines. The formats for data produced as a result of executing a data collection plan are also defined, as is the interface that must be supported by consumers of data collection plan output.

2.3 This specification defines a way for the equipment to notify consumers when the combination of activities on the equipment, including data acquisition, is causing the equipment to perform below supplier-defined criteria.

2.4 This specification defines a way for consumers to make ad-hoc on-demand requests for data from the equipment, outside of a data collection plan.

NOTICE: SEMI Standards and Safety Guidelines do not purport to address all safety issues associated with their use. It is the responsibility of the users of the Documents to establish appropriate safety and health practices, and determine the applicability of regulatory or other limitations prior to use.

3 Limitations

3.1 *Abstract Model*

3.1.1 This specification does not define a convention for naming or identifying sources, parameters, events, or exceptions. This document assumes that all such entities can be identified separately and represented as text. Such conventions shall be fully specified by any implementation mapping of this specification.

3.1.2 This specification does not define the mechanism used for identifying or authenticating consumers or enforcing privileges. It assumes that consumers can be uniquely identified by a text value, and that the equipment is capable of enforcing the standard privileges defined in this document. Such mechanisms shall be fully specified by any implementation mapping of this specification.

3.1.3 This specification does not define the mechanism used for determining how to locate consumers that are to receive the events defined by this specification. It assumes that the equipment has established a communications context with any consumers that can submit requests and has provided any context necessary to send events to those consumers. Such mechanisms shall be fully specified by any implementation mapping of this specification.

3.1.4 This specification does not define any algorithms or heuristics to use for detecting degrading equipment operational performance. Such techniques are specific to the equipment's internal hardware and software architecture and will very likely be unique to each class and/or model of equipment. It assumes that it is possible to detect such conditions and to identify which systems comprising the equipment are affected.

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

4 Referenced Standards and Documents

4.1 *SEMI Standards*

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

SEMI E40 — Standard for Processing Management

SEMI E94 — Specification for Control Job Management

SEMI E125 — Specification for Equipment Self Description (EqSD)

Documents
Document Sharing Hub

Full document details: SEMI E134-0710

4.2 *ISO Standards*[1]

ISO 11578:1996 — Information Technology – Open Systems Interconnection – Remote Procedure Call (RPC)

ISO 8601 — Data Elements and Interchange Formats – Information Interchange – Representation of Dates and Times

4.3 *OMG Standards*[2]

OMG 01-09-67 — Unified Modeling Language, v1.4

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 *DCP* — Data Collection Plan

5.1.3 *UML* — Unified Modeling 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 2, § R2-1, 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 *collection result* — the set of data values obtained during trace data collection.

5.2.4 *collection frequency* — the rate at which the collection of one or more data values is performed. This is not the same as the frequency with which the equipment internally samples these data from its components.

5.2.5 *data source* — a physical or logical entity associated with the equipment that is capable of providing data values independently of other equipment entities.

5.2.6 *event source* — a physical or logical entity associated with the equipment that is capable of generating events independently of other equipment entities

5.2.7 *exception source* — a physical or logical entity associated with the equipment that is capable of generating exceptions independently of other equipment entities

5.2.8 *parameter source* — a physical or logical entity associated with the equipment that is capable of providing parameters independently of other sources. This term may be used interchangeably with 'data source'.

5.2.9 *source id* — a name or other token that uniquely identifies a specific origin or producer of information from among possible sources.

[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; http://www.iso.ch

[2] Object Management Group, Inc., 250 First Ave. Suite 100, Needham, MA 02494, USA; Telephone: +1.781.444.0404; Fax: +1.781.444.0320; http://www.omg.org

Documents Document Sharing Hub

Full document details: SEMI E134-0710

6 Conventions

6.1 *Terms*

6.1.1 *Class, Data Type, Stereotype Name* — This specification uses the terms “class”, “data type”, “type”, or the class stereotype name interchangeably to refer to a UML class.

6.1.2 *Consumer* — This specification uses the term “consumer” to refer to any software entity that uses the DataCollectionManager interface to manage data collection on the equipment.

6.1.3 *Built-in DCP* — This specification uses the term “built-in” DCP to refer to pre-defined DCPs that are provided with the equipment. Built-in DCPs can be activated by any consumer, but cannot be changed or deleted by any consumer. See ¶ 11.1.2.1 for details.

6.1.4 *Reporting, Buffering, Sending* — This specification uses the term “reporting” to refer to the act of collecting event, exception, or trace data values to later be sent to a consumer. The term “buffering” is used to refer to the act of storing a data report on the equipment for some period of time before sending to a consumer. The term “sending” is used to refer to the act of sending a report to a consumer.

6.2 *Notation*

6.2.1 *Unified Modeling Language Notation* — All class and sequence diagrams in this specification make use of the Unified Modeling Language notation.

6.2.1.1 *UML Stereotypes* — This specification uses the following stereotypes in class diagrams:

- <<privilege>> — Classes with this stereotype identify an abstract privilege concept that grants permission to a consumer to perform an activity or operation.
- <<interface>> — Classes with this stereotype define a collection of operations and their semantics. It is the responsibility of the equipment and its consumers, where specified, to provide implementations of these classes.
- <<enumeration>> — Classes with this stereotype identify data types that have a finite set of discrete values that can be assumed by instances of the class.

6.2.2 *UML Associations* — The mechanism used for realizing 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.

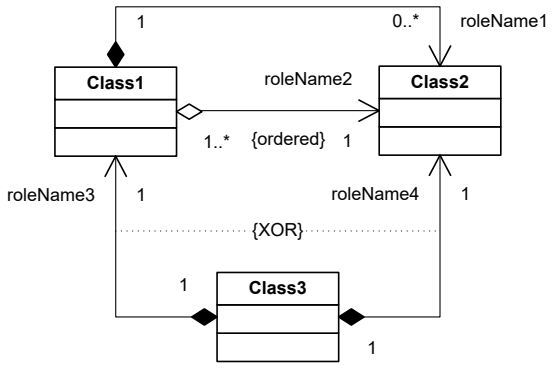

Figure 1
Association Notation

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

Documents
Document Sharing Hub
Full document details: SEMI E134-0710

6.2.3.1 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, ¶ 3.43.2.5 and ¶ 3.48 for further information.

6.2.3.2 Some diagrams use the UML "ordered" or "XOR" constraints where applicable. An "ordered" constraint on an association indicates that the order in which the target instances appear in the aggregation is semantically significant, and is described in the supporting text of the diagram. See the UML 1.4 specification, ¶ 3.43.2.2 for further information.

6.2.3.3 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, ¶ 3.42.5.1 for further information.

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

Table 1 Association Table Format

Association Role Name	*Definition*	*Comments*

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

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

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

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

Table 2 Attribute Table Format

Attribute Name	*Definition*	*Form*
		See list below

6.2.5.1 *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.

6.2.6 *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*	*Requestor/Sender*	*Responder/Receiver*
		See list below		

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

6.2.6.2 *Type* — Specifies the messaging semantics of the operation. Only Request-Reply (RR) and Fire-and-Forget (FF) 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.

Documents Document Sharing Hub
Full document details: SEMI E134-0710

6.2.6.3 *Requestor/Sender* — For RR semantics, identifies the entity that makes the request, for FF semantics, identifies the entity that sends the message. Can be either 'consumer' or 'equipment'.

6.2.6.4 *Responder/Receiver* — For RR semantics, identifies the entity that responds to the request, for FF semantics, identifies the entity that receives the message. Can be either 'consumer' or 'equipment'.

6.2.7 *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.

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

6.2.7.2 *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. All possible errors that could be returned by an operation are listed in these tables. Unless otherwise indicated, only one of the listed errors can be returned by an operation in a single invocation.

6.2.7.3 *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.

7 Background

7.1 *Equipment Data Collection*

7.1.1 Manufacturing equipment can be comprised of hundreds or thousands of components that interoperate in order for the equipment to perform its intended function. The volume of information that such equipment can produce, and the variation among equipment users in the data used for process control and other applications, makes it impractical to send all equipment data off-tool all the time, or to agree on a fixed set of data to make available for each equipment type. It is therefore important for the equipment to support the ability for consumers to dynamically define the data that is of interest to them (see Figure 2).

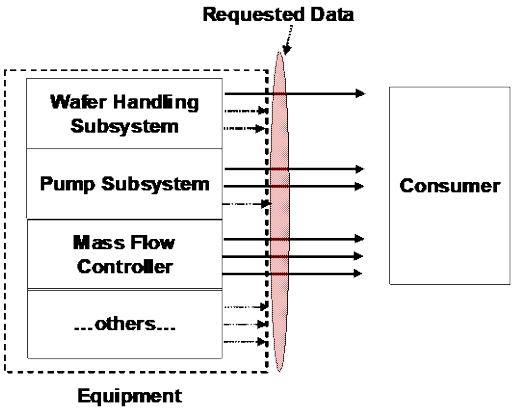

Figure 2
Consumer Specifies What Data Should Be Sent Off-Tool

7.1.2 In addition to declaring interest in specific data that a consumer is interested in, different consumers have different requirements for the acceptable latency with which data should be reported off-tool. For example, a consumer that is monitoring a trace of several process variables in order to detect an excursion needs to have this data available

Documents
Document Sharing Hub
Full document details: SEMI E134-0710

off-tool almost as quickly as the equipment can produce it (see Figure 3). A consumer that is monitoring low-level sensor/actuator activity for later diagnosis prefers not to have all of this data transmitted as it is produced, but buffered on the tool and transmitted at periodic intervals (see Figure 4).

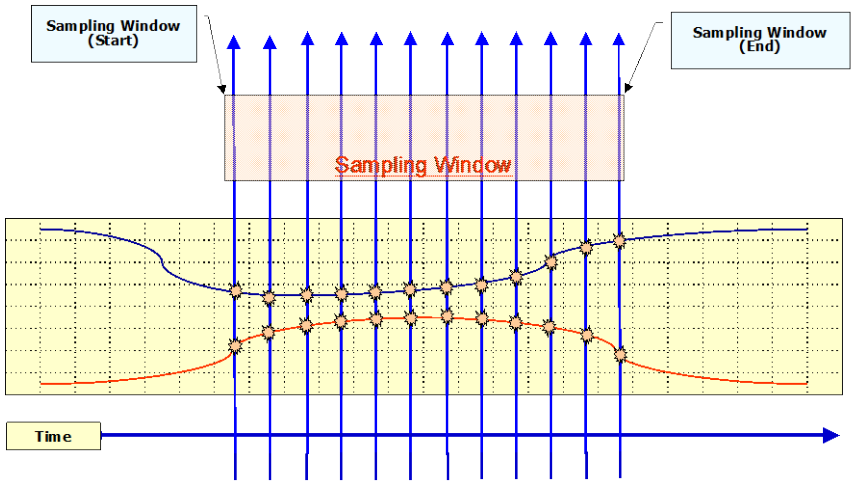

Figure 3
Un-buffered Data Transmission

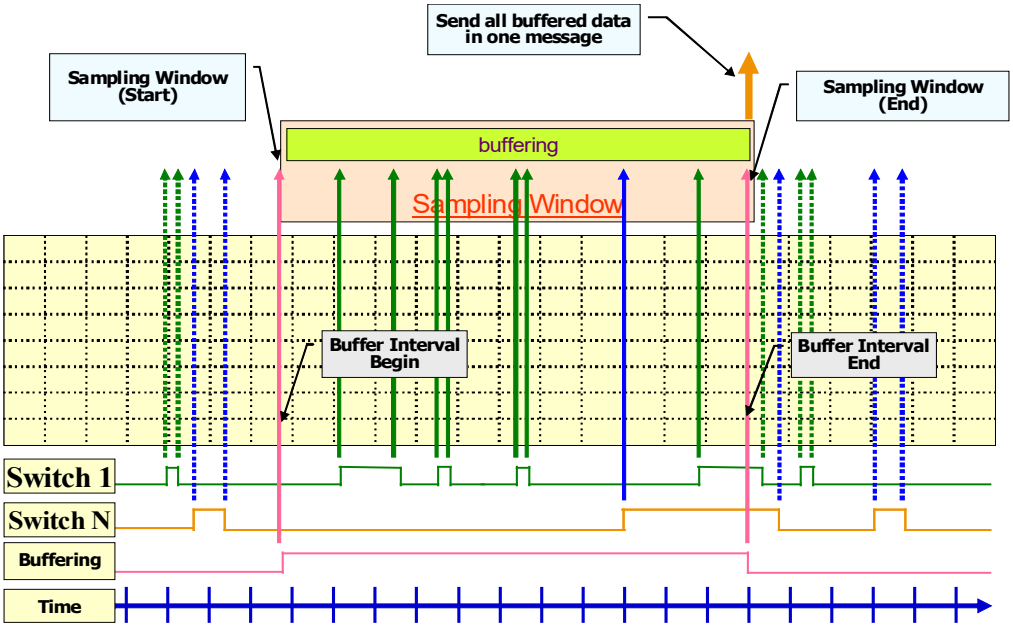

Figure 4
Buffered Data Transmission

7.1.3 Because a single tool is likely to have more than one consumer simultaneously interested in these different styles of data collection, a method for managing groups of related data as a unit can make it easier for consumers to define, enable, and disable the transmission of data. An example of this organization is shown in Figure 5.

Documents Document Sharing Hub
Full document details: SEMI E134-0710

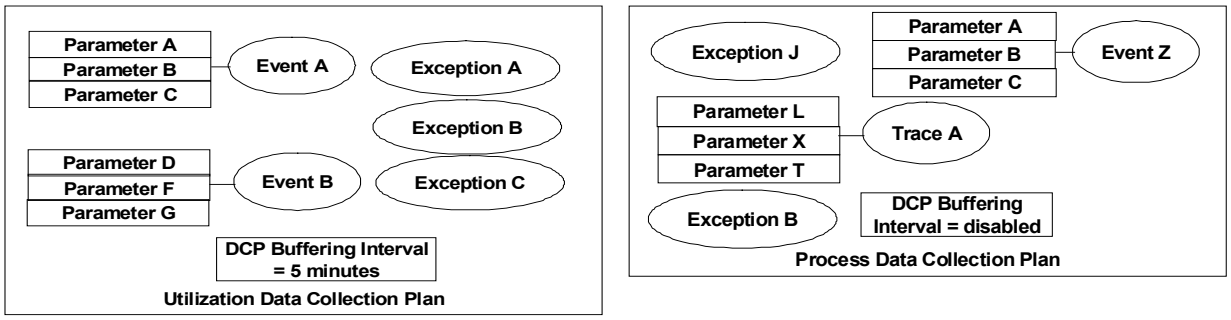

Figure 5
Data Collection Schemes with Different Objectives

7.1.4 It is possible for consumers to submit data collection requests which, when executing together, significantly tax the equipment's computing resources. When the equipment detects that this is occurring, it should be able to provide a warning to consumers that a performance threshold is being reached. Consumers can respond to these warnings by adjusting or disabling the active data collection requests. In the worst case, if the equipment determines that continuing to operate under degraded performance conditions may seriously jeopardize its ability to perform its intended function, the equipment can terminate all data collection activity.

7.1.5 This specification describes the concepts and behaviors necessary to facilitate these capabilities on equipment.

8 Overview

8.1 *Data Collection Concepts*

8.1.1 This specification allows consumers to describe a set of data in which they are interested, and to ask the equipment to send only that data off-tool, according to a buffering policy specified by the consumer. The types of data that are supported by this specification are described in the following sections.

8.2 *Trace Data*

8.2.1 Trace data allows data acquisition consumers to request the collection of data values that are continuously changing, or that change frequently such as pressure, temperature, RF power, magnetic field strength, or individual actuator state changes, etc. This document defines a mechanism for specifying when trace data collection begins and ends, what the collection frequency is, and what variables are to be collected in the trace (see ¶ 11.1.5).

8.2.2 This specification defines a mechanism for specifying when to start tracing (see ¶ 11.1.5.7):

- On the occurrence of a specific event or exception from a set of possible events and exceptions
- As soon as enabled

8.2.3 This specification defines a mechanism for specifying when to stop tracing (see ¶ 11.1.5.8):

- On the occurrence of a specific event or exception from a set of possible events and exceptions
- After a specified number of collection results have been accumulated
- As soon as disabled

8.2.4 This specification defines a mechanism for identifying parameters to be included in trace data collection (see ¶ 11.1.12). Consumers can use any naming convention supported by the equipment, provided that such conventions can support the concept of a parameter source and a parameter name represented as text fields.

8.2.5 This specification defines a mechanism to support on-tool buffering, allowing trace reports containing many collection results to be sent off-tool at a regular interval (see § 11.1.2.3 and § 11.1.5.3).

8.3 *Events*

8.3.1 Events allow data acquisition consumers to determine when a subsystem, process job, sensor/actuator, or other modeled system undergoes a state transition. Events may have data variables associated with the occurrence of the

Documents
Document Sharing Hub
Full document details: SEMI E134-0710

event. Such variables can include process data such as pressures, temperatures, or voltages, or context data such as job related data, recipe ID's, equipment configuration or offset values. Typical consumers need to be able to specify which variables are to be provided with the event when it is communicated.

8.3.2 This specification defines a mechanism for identifying events and parameters to be included in data collection. Consumers can use any naming convention that is supported by the equipment, provided that such convention can support the concept of event sources and event id's, and parameter sources and parameter names, each represented as text fields (see § 11.1.3 and § 14.1.3).

8.3.3 This specification defines a mechanism to support on-tool buffering of events, allowing a periodic report of any events that occurred during a specified interval to be sent off-tool (see § 11.1.2.3).

8.4 *Exceptions*

8.4.1 Exceptions allow data acquisition consumers to know when errors, warnings, or alarms occur on the equipment. These may indicate conditions which affect processing, safety, or the proper operation of the equipment. Exceptions may have data variables associated with the occurrence of the exception. The list of these variables will be fixed by the equipment implementation.

8.4.2 This specification defines a mechanism for identifying exceptions to be included in data collection. Consumers can use any naming convention that is supported by the equipment, provided that such convention can support the concept of exception sources and exception id's, each represented as text fields (see § 11.1.4 and § 14.1.4).

8.4.3 This specification defines a mechanism to support on-tool buffering of exceptions, allowing a periodic report of any exceptions that occurred during a specified interval to be sent off-tool (see § 11.1.2.3).

8.5 *Data Collection Plans*

8.5.1 A data collection plan assembles all of the related trace, event, and exception data that may be required for a specific consumer's purpose. For example, a consumer interested in tracking the utilization of the equipment may only be interested in specific events that mark actions the equipment takes that indicate how it is being exercised, certain exceptions that may interrupt the equipment, and no trace data. Such a data collection plan might be in effect at all times the equipment is running.

8.5.2 A fault detection consumer may be interested in trace, event, and exception data for a specific processing run, and may change the collected data depending on the recipe used. Such data collection plans may only be in effect while specific jobs are in process on the equipment. Coordination of data collection plan activation with job activity on the equipment is the consumers' responsibility, and is not specified in this document.

8.5.3 This document specifies a mechanism for consumers to define one or more data collection plans off-tool and submit those definitions to the equipment (see ¶ 9.1.2.2 and § 11). This document specifies a mechanism for determining what plans have been created on the equipment, how they are defined, and whether or not they are activated (see ¶¶ 9.1.2.2.16, 9.1.2.4, and 9.1.2.6). A mechanism for activating, deactivating, and deleting individual plans is also specified (see ¶¶ 9.1.2.5, 9.1.2.7, and 9.1.2.8). The required behavior of data collection plans is specified in § 12.

8.6 *Equipment Operational Performance Monitoring*

8.6.1 It is possible for consumers to submit data collection requests which, when executing together, significantly tax the equipment's computing resources, in addition to any load placed on the equipment by jobs that are in progress. This document specifies a mechanism for notifying all DCP consumers when such a condition has been detected, to identify what DCPs and jobs are in progress at the time, and to identify the specific equipment components that are being adversely affected (see ¶¶ 9.1.2.11, 13, 15.2.5, and 15.2.6).

8.7 *Ad-hoc Data Requests*

8.7.1 Ad-hoc data requests are used in cases where the value of some data items is needed at a specific instance in time, and which may not be associated with a specific event or exception. As a simple example, consider an application that monitors equipment utilization. Such an application may have an initialization procedure that includes determining the current state of each monitored component on the equipment. Without a mechanism for requesting this information at initialization, the application would instead have to wait for state change events from each component before it could determine their state. This may not be acceptable if the application provides a visualization of the equipment to an end user who relies on the application to call attention to problem conditions on the equipment.

Documents Document Sharing Hub

Full document details: SEMI E134-0710

8.7.2 This document specifies a mechanism for consumers to issue ad-hoc requests for data in order to support these uses, through the GetObjectInstanceIds and GetParameterValue operations of the DataCollectionManager interface (see ¶ 9.1.2.9 and ¶ 9.1.2.10).

9 Data Collection Plan Management

9.1 *DataCollectionManager Interface*

9.1.1 Figure 6 shows the interface used to manage data collection plans on the equipment, as well as making ad-hoc requests for specific data, and for determining the current performance conditions on the equipment.

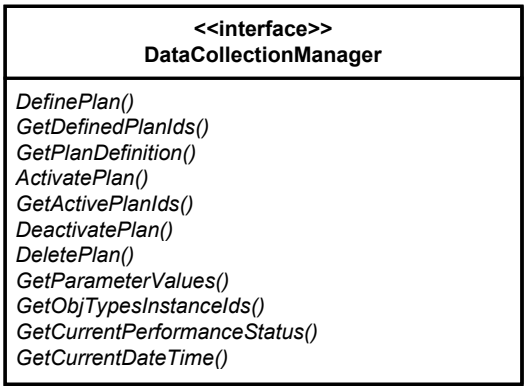

Figure 6
The DataCollectionManager Interface

9.1.2 *DataCollectionManager* — The DataCollectionManager interface provides consumers with operations to manage data collection activities on the equipment. The interface provides operations for consumers to submit new DCP's, activate and deactivate them, look up existing plans, and delete defined plans. The DataCollectionManager interface also provides operations to request the current equipment performance status as well as requesting ad-hoc data on demand.

9.1.2.1 *DataCollectionManager Operations*

Table 5 DataCollectionManager Operation Definition

Operation	*Description*	*Type*	*Requestor/ Sender*	*Responder/ Receiver*
DefinePlan	Submit a new plan definition to the equipment for validation.	RR	Consumer	Equipment
GetDefinedPlanIds	Retrieve a list of all valid DCP id's known to the equipment.	RR	Consumer	Equipment
GetPlanDefinition	Retrieve the definition of a specific DCP.	RR	Consumer	Equipment
ActivatePlan	Activate a defined DCP.	RR	Consumer	Equipment
GetActivePlanIds	Retrieve a list of all currently active DCPs.	RR	Consumer	Equipment
DeactivatePlan	Deactivate an active DCP.	RR	Consumer	Equipment
DeletePlan	Delete a DCP.	RR	Consumer	Equipment
GetParameterValues	Retrieve the most recent values of one or more Parameters.	RR	Consumer	Equipment
GetObjTypeInstanceIds	Retrieve a current list of the unique instance id's for one or more E39 ObjTypes.	RR	Consumer	Equipment

Documents
Document Sharing Hub
Full document details: SEMI E134-0710

Operation	*Description*	*Type*	*Requestor/ Sender*	*Responder/ Receiver*
GetCurrentPerformanceStatus	Retrieve the most recent equipment performance status.	RR	Consumer	Equipment
GetCurrentDateTime	Retrieve the current equipment date and time.	RR	Consumer	Equipment

9.1.2.2 *DefinePlan* — Upon receiving this request, the equipment shall first verify that the consumer has sufficient privilege (see § 10) to define a new DCP. If so, the equipment shall then validate the plan name for uniqueness and all referenced events, exceptions, parameters, and trace reports for existence and correctness. The equipment shall place no a-priori limitation on the number of plans that can be defined.

9.1.2.2.1 Any hard limitations imposed by constraints on available storage capacity shall be documented by the equipment supplier. If such limitations are encountered at runtime, the equipment shall return an equipment-defined error indicating that the DCP was not successfully defined, and a human-readable description of the reason. The mechanism for supporting such errors shall be fully specified in implementation mappings provided for this specification.

9.1.2.2.2 If there are any issues with the submitted plan, the equipment shall return a description of all problems detected with the plan (see ¶ 9.1.2.2.7 for a description of possible DCP definition problems), and the plan shall not be defined on the equipment.

9.1.2.2.3 If there are no issues with the submitted plan, the equipment shall record the date and time at which the plan definition was submitted and the id of the consumer that submitted the plan. The equipment shall store this information and the plan definition in non-volatile storage. It shall thereafter be legal for DCP requests to refer to the submitted plan by its id field.

9.1.2.2.4 *DefinePlan Operation Arguments*

Table 6 DefinePlan Argument Definitions

Argument	*Description*	*Kind*	*Form*
newPlan	The definition of a new DCP.	in	Structured data, of type DataCollectionPlan, described in ¶ 11.1.2.
planDefined	An acknowledgement that the new DCP definition was well-formed, correct, and supportable.	out	Structured data, of type DCPDefined, described in ¶ 9.1.2.2.5.
error	The consumer submitting the DCP does not have sufficient privilege to define DCPs.	error	Structured data, of type UnauthorizedOperation, described in ¶ 9.1.2.2.6.
error	The DCP definition is invalid.	error	Structured data, of type InvalidPlan, described in ¶ 9.1.2.2.7.

Figure 7
DCPDefined Class

9.1.2.2.5 *DCPDefined* — This class describes a DCP that has been successfully defined on the equipment.

Documents Document Sharing Hub
Full document details: SEMI E134-0710

9.1.2.2.5.1 *DCPDefined Attribute Definition Table*

Table 7 DCPDefined Attribute Definition

Attribute Name	*Definition*	*Form*
planId	Identifies a defined DCP.	Text, equal to the 'id' attribute of the DataCollectionPlan defined on the equipment.
name	Consumer-defined (or supplier-defined, in the case of built-in DCPs) human-readable name for the DCP.	Text, equal to the 'name' attribute of the DataCollectionPlan defined on the equipment.
timeDefined	The date and time at which the DCP was originally defined.	Text, formatted according to ¶ 14.2. For built-in plans (see ¶ 11.1.2.1), this field shall be equal to the date and time at which the plan was installed on the equipment.
definedBy	The identity of the consumer that defined the DCP.	Text, equal to a valid identifier for the consumer that took the action (see Limitations, ¶ 3.1.2). For built-in plans (see ¶ 11.1.2.1), this field shall be a URN of the form "urn:semi-org:equipment". The valid values to use for this attribute to identify consumers shall be specified by any adjunct document defining a technical binding for this specification.
description	Consumer-defined (or supplier-defined, in the case of built-in DCPs) description of the purpose of the DCP.	Text, equal to the 'description' attribute of the DataCollectionPlan defined on the equipment.

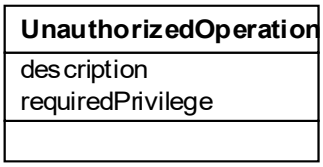

Figure 8
UnauthorizedOperation Error

9.1.2.2.6 *UnauthorizedOperation* — This class describes an authorization error associated with a specific operation.

9.1.2.2.6.1 *UnauthorizedOperation Attribute Definition Table*

Table 8 UnauthorizedOperation Attribute Definition

Attribute Name	*Definition*	*Form*
description	A description of the error.	Text, including the name of the operation and a description of any important context associated with the original request.
requiredPrivilege	The name of the privilege that is required for the operation.	Text, equal to a valid privilege name or identifier (see § 10) defined on the equipment that the consumer must possess in order to execute the operation. The valid values to use for this attribute shall be specified by any adjunct document defining a technical binding for this specification.

Documents
Document Sharing Hub
Full document details: SEMI E134-0710

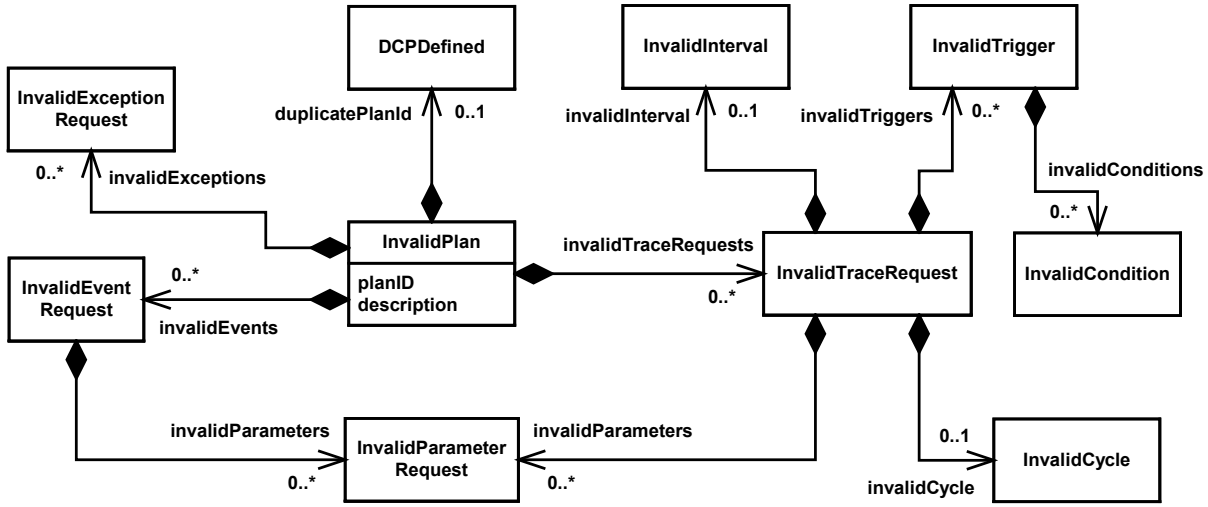

Figure 9
InvalidPlan Error

9.1.2.2.7 *InvalidPlan* — This class provides an overall description of all errors detected with a submitted DCP.

9.1.2.2.7.1 *InvalidPlan Attribute Definition Table*

Table 9 InvalidPlan Attribute Definition

Attribute Name	*Definition*	*Form*
planId	The identifier of the invalid DCP.	Text, equal to the 'id' attribute of the submitted DataCollectionPlan.
description	A human-readable description of why the DCP was rejected.	Text.

9.1.2.2.7.2 *InvalidPlan Association Definition Table*

Table 10 InvalidPlan Association Definition

Association Role Name	*Definition*	*Comments*
invalidEvents	List of event requests that have one or more errors.	Unordered list of elements of type InvalidEventRequest, described in ¶ 9.1.2.2.8.
invalidExceptions	List of exception requests that have one or more errors.	Unordered list of elements of type InvalidExceptionRequest, described in ¶ 9.1.2.2.9.
duplicatePlanId	Description of a pre-existing plan that has an identical 'id' attribute.	Structure, of type DCPDefined, described in ¶ 9.1.2.2.5.
invalidTraceRequests	List of trace requests that have one or more errors.	Unordered list of elements of type InvalidTraceRequest, described in ¶ 9.1.2.2.11.

9.1.2.2.8 *InvalidEventRequest* — This class describes errors detected in requested events in a submitted plan definition. An event request can be invalid if (1) the equipment does not recognize the sourceId, (2) the equipment does not recognize the eventId, (3) the sourceId and eventId are both recognized by the equipment, but the specified source is not the source of the requested event, or (4) the same source/event combination appears in more than one event request within the DCP.

Documents Document Sharing Hub
Full document details: SEMI E134-0710

9.1.2.2.8.1 *InvalidEventRequest Attribute Definition Table*

Table 11 InvalidEventRequest Attribute Definition

Attribute Name	*Definition*	*Form*
sourceId	The sourceId from the invalid event request provided in the DCP.	Text.
eventId	The eventId from the invalid event request provided in the DCP.	Text.
invalidSourceId	Whether or not the sourceId identifies a valid source supported by the equipment.	Boolean. If true, the identified source is not recognized.
invalidEventId	Whether or not the eventId identifies a valid event supported by the equipment.	Boolean. If true, the event id is not recognized.
notProducedBySource	Whether or not the identified source is the source of the identified event.	Boolean. If true, both the source and event id's are recognized, but the identified source does not generate the requested event.
isDuplicate	Whether or not the requested event appears more than once in the DCP.	Boolean.

9.1.2.2.8.2 *InvalidEventRequest Association Definition Table*

Table 12 InvalidEventRequest Association Definition

Association Role Name	*Definition*	*Comments*
invalidParameters	List of parameter requests for the event that have one or more errors.	Unordered list of zero or more elements of type InvalidParameterRequest, described in ¶ 9.1.2.2.10.

9.1.2.2.9 *InvalidExceptionRequest* — This class describes errors detected in requested exceptions in a submitted plan definition. An exception request can be invalid if (1) the equipment does not recognize the sourceId, (2) the equipment does not recognize the exceptionId, (3) the equipment recognizes both the sourceId and exceptionId, but the specified source is not the source of the requested exception, (4) the requested severity is not supported, or (5) the same exception criteria appears in more than one event request within the DCP.

9.1.2.2.9.1 *InvalidExceptionRequest Attribute Definition Table*

Table 13 InvalidExceptionRequest Attribute Definition

Attribute Name	*Definition*	*Form*
sourceId	The sourceId from the invalid exception request provided in the DCP.	Text.
exceptionId	The eventId from the invalid exception request provided in the DCP.	Text.
severity	The severity from the invalid exception request provided in the DCP.	Text. The value shall match one of the enumeration strings defined for the severity attribute of the Exception class defined in SEMI E125.
invalidSourceId	Whether or not the sourceId identifies a valid source supported by the equipment.	Boolean. If true, the identified source is not recognized.
invalidExceptionId	Whether or not the exceptionId identifies a valid exception supported by the equipment.	Boolean. If true, the exception id is not recognized.
invalidSeverity	Whether or not the specified severity is recognized by the equipment.	Boolean. If true, the severity is not recognized.

Documents
Document Sharing Hub
Full document details: SEMI E134-0710

Attribute Name	*Definition*	*Form*
notProducedBySource	Whether or not the identified source is the source of the identified exception.	Boolean. If true, both the source and exception id's are recognized, but the identified source does not generate the requested exception.
isDuplicate	Whether or not the same exception criteria appears in more than one exception request in the DCP.	Boolean.

9.1.2.2.10 *InvalidParameterRequest* — This class describes errors detected in requested parameters in a submitted plan definition. Parameters are requested with either an event or as part of a trace request. A parameter request can be invalid if (1) the equipment does not recognize the sourceId, (2) the equipment does not recognize the parameterName, (3) the equipment recognizes both the sourceId and parameterName but the specified source is not the source of the requested parameter, or (4) the Parameter is produced by the specified source but (a) the equipment does not support reporting the parameter with the corresponding event request or (b) the equipment does not support including this parameter in a trace request.

9.1.2.2.10.1 *InvalidParameterRequest Attribute Definition Table*

Table 14 InvalidParameterRequest Attribute Definition

Attribute Name	*Definition*	*Form*
sourceId	The sourceId from the requested parameter.	Text.
parameterName	The name of the requested parameter.	Text.
invalidSourceId	Whether or not the sourceId identifies a valid source supported by the equipment.	Boolean. If true, the identified source is not recognized.
invalidParameterName	Whether or not the parameterName identifies a valid parameter supported by the equipment.	Boolean. If true, the identified parameter is not recognized.
notProducedBySource	Whether or not the identified source is the source of the requested parameter.	Boolean. If true, the source does not provide the requested parameter.
invalidContext	Whether or not the requested parameter is available in the requested context.	Boolean. If true, the equipment does not support reporting this parameter with the requested event, or including the parameter in a trace request.

9.1.2.2.11 *InvalidTraceRequest* — This class describes errors detected in requested traces in a submitted plan definition. A trace request can be invalid if (1) the same trace id appears in more than one TraceRequest within the same DCP, (2) the TraceRequest includes an invalid ParameterRequest, (3) the TraceRequest specifies an invalid start/stop trigger, (4) the TraceRequest specifies an invalid trigger cycle, or (5) the TraceRequest specifies an interval that can't be supported by the equipment.

Documents Document Sharing Hub
Full document details: SEMI E134-0710

9.1.2.2.11.1 *InvalidTraceRequest Attribute Definition Table*

Table 15 InvalidTraceRequest Attribute Definition

Attribute Name	*Definition*	*Form*
traceId	The traceId from the invalid trace request provided in the DCP.	Integer.
duplicateId	Whether or not the trace id is duplicated.	Boolean. If true, the trace id appears more than once in the same DCP.

9.1.2.2.11.2 *InvalidTraceRequest Association Definition Table*

Table 16 InvalidTraceRequest Association Definition

Association Role Name	*Definition*	*Comments*
invalidParameters	List of parameter requests for the trace request that have one or more errors.	Unordered list of zero or more elements of type InvalidParameterRequest, described in ¶ 9.1.2.2.10.
invalidTriggers	List of start/stop triggers for the trace request that have one or more errors.	Unordered list of zero or more elements of type InvalidTrigger, described in ¶ 9.1.2.2.12.
invalidInterval	Indicates an unsupportable trace interval.	Zero or one element of type InvalidInterval, described in ¶ 9.1.2.2.12.2.
invalidCycle	Indicates that the trace request contains an invalid cycle.	Zero or one element of type InvalidCycle, described in ¶ 9.1.2.2.15.
invalidGroupSize ForInterval	Indicates an unsupportable groupSize for the specified intervalInSeconds.	Zero or one element of type InvalidGroupSizeForInterval, described in ¶ 9.1.2.2.16.

9.1.2.2.12 *InvalidTrigger* — This class describes errors detected in requested start/stop triggers for traces in a submitted plan definition. A trigger can be invalid if (1) the equipment does not recognize the sourceId, (2) the equipment does not recognize the eventId or exceptionId, (3) the sourceId and eventId or exceptionId are both recognized by the equipment, but the specified source is not the source of the requested event or exception, or (4) the same start trigger appears more than once in the same trace request, or the same stop trigger appears more than once in the same trace request.

9.1.2.2.12.1 *InvalidTrigger Attribute Definition Table*

Table 17 InvalidTrigger Attribute Definition

Attribute Name	*Definition*	*Form*
invalidStartTrigger	Whether or not the problem is with a start trigger. False indicates that the problem is with a stop trigger.	Boolean.
invalidEventTrigger	Whether or not the problem is with an event trigger. False indicates that the problem is with an exception trigger.	Boolean.
invalidExceptionState	The requested exception state is not defined for the exception.	Boolean.
sourceId	The sourceId from the invalid trigger provided in the DCP.	Text.
itemId	The event or exception id from the invalid trigger provided in the DCP.	Text.
invalidSourceId	Whether or not the sourceId identifies a valid source supported by the equipment.	Boolean. If true, the identified source is not recognized.
invalidItemId	Whether or not the eventId or exceptionId identifies a valid event or exception supported by the equipment.	Boolean. If true, the identified exception or event is not recognized.

Documents
Document Sharing Hub
Full document details: SEMI E134-0710

Attribute Name	*Definition*	*Form*
notProducedBySource	Whether or not the identified source is the source of the identified event or exception.	Boolean. If true, the source does not generate the requested exception or event.
isDuplicate	Whether or not the same trigger appears more than once as a start trigger, or as a stop trigger.	Boolean.

9.1.2.2.12.2 *InvalidTrigger Association Definition Table*

Table 18 Trigger Association Definition

Association Role Name	*Definition*	*Comments*
invalidConditions	List of zero or more instances of type InvalidCondition.	Allowed only if the InvalidTrigger is for an EventTrigger.

9.1.2.2.13 *InvalidCondition* — This class describes errors detected in the Conditions defined for requested start/stop triggers for traces in a submitted plan definition.

9.1.2.2.13.1 *InvalidCondition Attribute Definition Table*

Table 19 InvalidCondition Attribute Definition

Attribute Name	*Definition*	*Form*
unknownOperator	The operator is not one of the defined choices.	Boolean
badOperator	The operator specified is not compatible with the type of Parameter referenced by the Condition.	Boolean
unknownParameter	The referenced Parameter is not defined for this equipment.	Boolean
invalidParameter	The referenced Parameter is not valid for the trigger event or the format of the reference Parameter is not allowed for Conditions.	Boolean
badValue	The conditionValue specified is not compatible with the referenced Parameter or is a value that is not allowed.	Boolean

9.1.2.2.14 *InvalidInterval* — This class describes errors detected in requested trace intervals in a submitted plan definition. A trace interval is invalid if the equipment cannot collect the requested parameters at the requested frequency. The error returns the closest valid trace interval that can be supported by the equipment.

9.1.2.2.14.1 *InvalidInterval Attribute Definition Table*

Table 20 InvalidInterval Attribute Definition

Attribute Name	*Definition*	*Form*
validInterval	A valid trace interval closest to the requested interval, expressed in seconds.	Floating point.

9.1.2.2.15 *InvalidCycle* — This class describes errors detected in a cyclical trace request in a submitted plan definition. A trace cycle is invalid if the TraceRequest 'isCyclical' attribute (see ¶ 11.1.5) is true, but does not include both a start and stop trigger.

Documents Document Sharing Hub
Full document details: SEMI E134-0710

9.1.2.2.15.1 *InvalidCycle Attribute Definition Table*

Table 21 InvalidCycle Attribute Definition

Attribute Name	*Definition*	*Form*
needsStartTrigger	Whether or not the TraceRequest is missing a start trigger.	Boolean.
needsStopTrigger	Whether or not the TraceRequest is missing a stop trigger.	Boolean.

9.1.2.2.16 *InvalidGroupSizeForInterval* — This class describes a problem detected in a submitted plan definition when the groupSize is too small for the specified intervalInSeconds. The resulting reporting interval (intervalInSeconds * groupSize) requires that reports be sent more frequently than the equipment can support. The error returns the smallest valid groupSize that can be supported by the equipment.

9.1.2.2.16.1 *InvalidReportInterval Attribute Definition Table*

Table 22 InvalidGroupSizeForInterval Attribute Definition

Attribute Name	*Definition*	*Form*
validGroupSize	The smallest groupSize value allowed for the requested intervalInSeconds.	Integer >0.

9.1.2.3 *GetDefinedPlanIds* — Upon receiving this request, the equipment shall return a list of all currently defined data collection plan id's that are accessible by the consumer (see § 10), the time at which they were defined, and the id of the consumer that defined them. If the consumer does not have sufficient privilege (see § 10), then the equipment shall return an error.

9.1.2.3.1 *GetDefinedPlanIds Operation Arguments*

Table 23 GetDefinedPlanIds Argument Definitions

Argument	*Description*	*Kind*	*Form*
definedPlans	List of defined plans that are visible to the consumer.	out	Unordered list of structured data, of type DCPDefined, described in ¶ 9.1.2.2.5.
error	The consumer does not have sufficient privilege to retrieve the list of defined plan id's.	error	Structure, of type UnauthorizedOperation, described in ¶ 9.1.2.2.6.

9.1.2.4 *GetPlanDefinition* — Upon receiving this request, the equipment shall return the requested plan definition to the consumer as it was originally submitted to the equipment.

9.1.2.4.1 If the consumer does not have sufficient privileges (see § 10) to access the requested DCP definition, the equipment shall return a UnauthorizedOperation error.

9.1.2.4.2 *GetPlanDefinition Operation Arguments*

Table 24 GetPlanDefinition Argument Definitions

Argument	*Description*	*Kind*	*Form*
planId	Identifier for the requested plan.	in	Text, equal to a valid DCP id.
planDefinition	The requested plan definition.	out	Structure, of type DataCollectionPlan, described in ¶ 11.1.

Documents
Document Sharing Hub
Full document details: SEMI E134-0710

Argument	*Description*	*Kind*	*Form*
error	The consumer submitting the DCP does not have sufficient privilege to retrieve the requested DCP.	error	Structure, of type UnauthorizedOperation, described in ¶ 9.1.2.2.6.
error	The equipment does not recognize the requested planId.	error	Structure, of type NoSuchPlan, described in ¶ 9.1.2.4.3.

NoSuchPlan
planId

Figure 10
NoSuchPlan Error

9.1.2.4.3 *NoSuchPlan* — This class describes an error indicating an unrecognized DCP id.

9.1.2.4.4 *NoSuchPlan Attribute Definition Table*

Table 25 NoSuchPlan Attribute Definition

Attribute Name	*Definition*	*Form*
planId	The unrecognized planId.	Text.

9.1.2.5 *ActivatePlan* — Upon receiving this request, the equipment shall begin sending all DCP events, exceptions, and trace data collection reports to the consumer according to any buffering policies specified in the DCP. The equipment shall place no a-priori limitation on the number of DCPs that a consumer can activate. The equipment shall place no a-priori limitation on the number of consumers that can activate the same DCP (see § 12.1 Table 61, transition 4). This operation is the only means by which a DCP can be activated.

9.1.2.5.1 Any limitations on supportable DCP activity imposed by constraints on available equipment processing power or network interface bandwidth shall be documented by the equipment supplier. If such limitations are encountered at runtime, the equipment shall use the equipment operational performance monitoring messages and behavior defined in § 13 to manage DCP activity with consumers.

9.1.2.5.2 The equipment shall ensure that the requested DCP is valid (see ¶ 9.1.2.2.7) at the time it is activated. If, at activation time, the requested DCP is not valid, the equipment shall return an InvalidPlan error, and shall not activate the DCP. The equipment shall not prevent DCP activation for any reason (including internal communication failures) if the DCP is valid and the consumer has sufficient privilege (see § 10). For required behavior in the event of internal communication failures, see ¶ 12.2.2.

9.1.2.5.3 If the consumer does not have sufficient privilege (see § 10) to activate the requested DCP, the equipment shall return a UnauthorizedOperation error.

9.1.2.5.4 The equipment shall respond to duplicate requests from the same consumer to activate the same DCP by returning an error including the DCPActivated information corresponding to that consumer's original activation request, but shall take no other actions. The equipment shall not activate a DCP more than once for the same consumer.

9.1.2.5.5 *ActivatePlan Operation Arguments*

Table 26 ActivatePlan Argument Definitions

Argument	*Description*	*Kind*	*Form*
planId	Identifier for the requested plan.	in	Text, equal to a valid DCP id.

Documents Document Sharing Hub
Full document details: SEMI E134-0710

Argument	*Description*	*Kind*	*Form*
activatedPlan	Confirmation that the requested plan has been activated.	out	Structured data, of type DCPActivated, described in ¶ 9.1.2.8.4.
error	The consumer submitting the DCP does not have sufficient privilege to activate the requested DCP.	error	Structured data, of type UnauthorizedOperation, described in ¶ 9.1.2.2.6.
error	The equipment does not recognize the requested planId.	error	Structured data, of type NoSuchPlan, described in ¶ 9.1.2.4.3.
error	The requested DCP is invalid. This can result from a revision to the metadata.	error	Structured data, of type InvalidPlan, described in ¶ 9.1.2.2.7.
error	The consumer making the request has previously activated the same DCP.	error	Structured data, of type DCPIsActive, described in ¶ 9.1.2.8.3.

9.1.2.6 *GetActivePlanIds* — Upon receiving this request, the equipment shall return a list of the id's of all DCPs that have been activated by the requesting consumer at the time of the request. If the consumer has sufficient privilege (see § 10), the id's of all activation requests for all currently active DCPs shall be returned.

9.1.2.6.1 If the consumer does not have sufficient privilege (see § 10) to retrieve the currently-active DCP id's, the equipment shall return a UnauthorizedOperation error.

9.1.2.6.2 *GetActivePlanIds Operation Arguments*

Table 27 GetActivePlanIds Argument Definitions

Argument	*Description*	*Kind*	*Form*
activePlans	List of active DCP id's.	out	Unordered list of elements of type DCPActivated, described in ¶ 9.1.2.8.4. Empty if no DCPs are active. See ¶ 9.1.2.6.3 for further explanation.
error	The consumer submitting the DCP does not have sufficient privilege to retrieve a list of active DCPs.	error	Structured data, of type UnauthorizedOperation, described in ¶ 9.1.2.2.6.

9.1.2.6.3 *activePlans* — The information returned in the response to this operation depends on the privilege assigned to the requesting consumer. The equipment shall only return the DCPActivated information for activation requests made by the consumer that is invoking the GetActivePlanIds operation (unless the consumer has sufficient privilege—see ¶ 9.1.2.6.3.1). For example, if some DCPs have been activated by other consumers, but not by the requesting consumer, the equipment shall not return the DCPActivated information for those other DCPs. Also, if the requesting consumer has activated a DCP, and other consumers have also activated that same DCP, the equipment shall not return the DCPActivated information for those other consumers.

9.1.2.6.3.1 For consumers with sufficient privilege only (see § 10), the equipment shall return all available DCPActivation information for all consumers of all DCPs that are active at the time of the request.

9.1.2.7 *DeactivatePlan* — Upon receiving this request, the equipment shall stop sending of all event, exception, and trace requests associated with the identified DCP to the requesting consumer. If there are no other consumers receiving data from the identified DCP, the equipment shall disable all collection/buffering activity related to this DCP (see Table 61, transition 8). In this case, the equipment shall discard any buffered data at the time of the request.

9.1.2.7.1 If the identified DCP has not been previously activated by the requesting consumer, and the request is not a termination request (see ¶ 9.1.2.7.5), the equipment shall return the DCPNotActive error. If the request is a termination request, but the requested plan has not been previously activated by any consumer, the equipment shall return the DCPNotActive error.

9.1.2.7.2 If the consumer does not have sufficient privilege (see § 10) to deactivate the requested DCP, the equipment shall return a UnauthorizedOperation error.

Documents
Document Sharing Hub
Full document details: SEMI E134-0710

9.1.2.7.3 *DeactivatePlan Operation Arguments*

Table 28 DeactivatePlan Argument Definitions

Argument	*Description*	*Kind*	*Form*
planId	Identifier for the requested plan.	in	Text, equal to a valid DCP id.
terminate	Whether or not the DCP should be deactivated for all consumers.	in	Boolean. See ¶ 9.1.2.7.4 for more information.
deactivatedPlan	Confirmation that the requested plan has been deactivated	out	Structure, of type DCPDeactivated, described in ¶ 9.1.2.7.6.
error	The consumer submitting the DCP does not have sufficient privilege to deactivate the DCP	error	Structured data, of type UnauthorizedOperation, described in ¶ 9.1.2.2.6.
error	The equipment does not recognize the requested planId	error	Structured data, of type NoSuchPlan, described in ¶ 9.1.2.4.3.
error	The requested plan is not active.	error	Structured data, of type DCPNotActive, described in ¶ 9.1.2.7.7.

9.1.2.7.4 *planId* — This operation defines the following reserved planId to name groups of DCPs for deactivation. This identifier is not valid for use as a DCP id in any operation other than DeactivatePlan.

9.1.2.7.4.1 *"urn:semi-org:dcm:allDCPs"* — This identifier can be used to request deactivation of all plans that are currently active for the requesting consumer. If other consumers have activated any of these same plans, those consumers shall continue to receive DataCollectionReports from those plans. When used together with a request to terminate (see ¶ 9.1.2.7.5), the equipment shall terminate all DCPs currently active on the equipment for all consumers, including those that have been activated by the requesting consumer. The consumer must have sufficient privilege (see § 10) to request termination of a DCP.

9.1.2.7.5 *terminate* — The following sections describe the behavior of the terminate argument.

9.1.2.7.5.1 *planId=(normal), terminate=false*. If the DCP is active for the requesting client, deactivate the requested DCP for this client. The DCP remains active for all other clients, if any. If the request is successful, the planId of the deactivatedPlan in the response shall match the planId in the request. If the DCP is not active for this client, the request fails with DCPNotActive. If the specified DCP does not exist, the request fails with NoSuchPlan.

9.1.2.7.5.2 *planId=(normal), terminate=true*. If the client has the ManageAnyDCP privilege and the DCP is active for at least one client, deactivate the DCP for all clients. If the client does not have the ManageAnyDCP privilege, the request fails with UnauthorizedOperation. Otherwise, the request will fail if the DCP is not active (DCPNotActive) or if the DCP does not exist (NoSuchPlan).

9.1.2.7.5.3 *planId=urn:semi-org:dcm:allDCPs, terminate=false*. Deactivate for this client all DCPs currently active for the client. If any of these DCPs are also active for other clients, they remain active for those clients. If the request is successful, the planId of the deactivatedPlan in the response shall be *urn:semi-org:dcm:allDCPs*. If no DCPs are active for the requesting client, the request fails with DCPNotActive with a planId equal to *urn:semi-org:dcm:allDCPs*.

9.1.2.7.5.4 *planId=urn:semi-org:dcm:allDCPs, terminate=true*. If the requesting client has the ManageAnyDCP privilege, then deactivate all DCPs for all clients. Otherwise fail with UnauthorizedOperation. If the request is successful, the planId of the deactivatedPlan in the response shall be *urn:semi-org:dcm:allDCPs*. If no DCPs are active for any client, the request fails with DCPNotActive with a planId equal to *urn:semi-org:dcm:allDCPs*.

Documents Document Sharing Hub
Full document details: SEMI E134-0710