# Introduction to the SEMI Standards: EDA - Interface A

## Table of Contents


**1. EDA/INTERFACE A OVERVIEW .................................................. 3**

**2. TERMINOLOGY AND ACRONYMS .................................................. 4**

**3. SEMI STANDARDS .................................................. 7**

3.1 E120 Specification for the Common Equipment Model (CEM) .................................................. 7
3.2 E125 Specification for Equipment Self Description (EqSD) .................................................. 10
3.3 E132 Specification for Equipment Client Authentication and Authorization .................................................. 12
3.4 E134 Specification for Data Collection Management .................................................. 14

3.4.1 Events .................................................. 17
3.4.2 Exceptions .................................................. 17
3.4.3 Traces .................................................. 17

3.5 E164 Specification for EDA Common Metadata .................................................. 17
3.6 EDA Client Operations .................................................. 18

**4. FREQUENTLY ASKED QUESTIONS (FAQ) .................................................. 19**

## - NOTICE -

The information contained in this document is subject to change without notice. Every effort has been made to supply complete and accurate information. However, Cimetrix Inc. makes no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Cimetrix shall not be liable for errors contained herein or direct, indirect, special, incidental, or consequential damages in connection with the furnishing, performance, or use of this document.

## - COPYRIGHT -



## - TRADEMARKS -

Cimetrix is a registered trademark. CIM300, CIMConnect, CIMFoundation, CIM40, CIM87, CIM90, CIM94, CIM58, CIM116, CIM148, CIM157, SECSConnect, ECCE Plus, TESTConnect, CIMPortal, CIMPortal Plus, EDAConnect, and CIMControlFramework (CCF) are trademarks of Cimetrix Incorporated. All other registered trademarks and trademarks are properties of their respective holders.

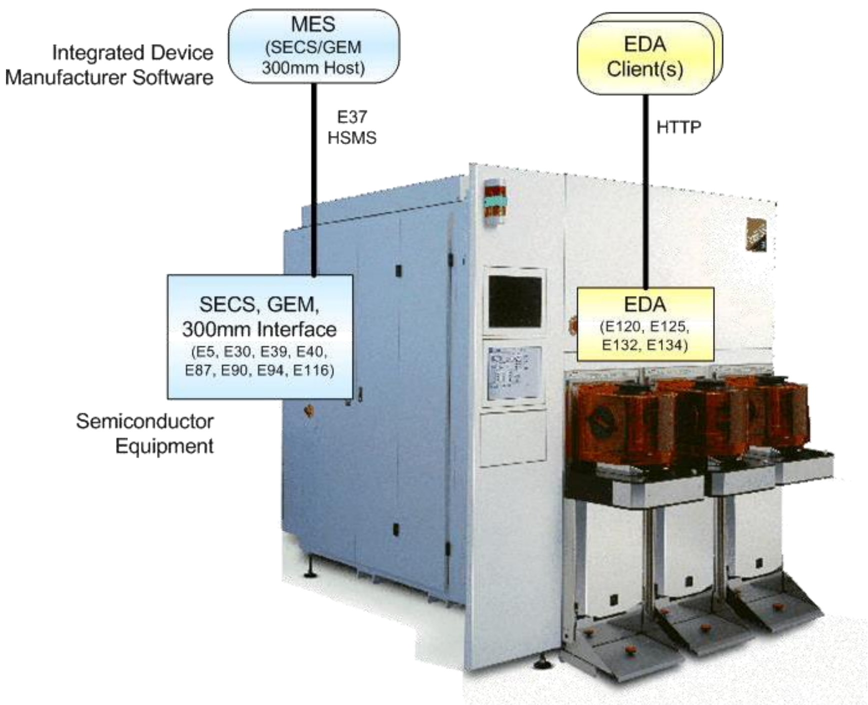
Cimetrix®
6979 High Tech Dr
Midvale, UT 84047
Phone: 801.256.6500
www.cimetrix.com

# 1. EDA/Interface A Overview

The SEMI Equipment Data Acquisition (EDA, also known as Interface A) standards suite is an integrated set of standards designed for the electronics manufacturing industry to enable high-performance, flexible communications between the factory data collection infrastructure and the manufacturing equipment. These standards do not replace the existing SECS/GEM command and control capabilities, but rather supplement them to provide support the mission-critical analysis, process control, and optimization applications on which companies increasingly depend to manage their factories.

Note: The terms **EDA** and **Interface A** are used interchangeably in the semiconductor industry, and for consistency, we will primarily use the term **EDA** in this paper.

EDA-compliant equipment must implement multiple interfaces as Web Services. EDA clients must also implement certain interfaces as Web Services. The Web Services use SOAP/XML-formatted messages over an HTTP or HTTPS connection to transfer information. The main SEMI EDA standards include E120, E125, E128, E132, E134, E138, and E164. Solutions must comply with the specific SOAP/XML implementations (called "bindings") of these standards: E120.1, E125.1, E132.1, and E134.1. In 2012, SEMI standard E164 was added to the set of EDA standards with detailed implementation guidelines for the equipment metadata model structure and content.




EDA also supports multiple, independent client access to the equipment for data gathering, which provides far more factory system architecture flexibility than previous generations of connectivity standards. As mentioned earlier, EDA does not replace the SEMI SECS/GEM standards (E4, E5, E30, and E37) or the SEMI GEM300 standards (E39, E40, E87, E90, E94, E116, E148, and E157), since EDA does not include features for equipment control or configuration. Instead, EDA must be supported by the equipment suppliers in addition to the other required interfaces. Initially, some EDA deployments provided little more data than the SECS/GEM interface, but factory customers ultimately expect access to much more detailed equipment data—particularly state information about sensors, actuators, key mechanisms and subsystems, internal control parameters, and other raw data—necessary to better analyze, understand, and control equipment and process behavior. To make all this data available while achieving performance and data quality expectations, it may be necessary to redesign the embedded control systems of some equipment.

Since the initial release of the EDA standards in 2005, IC makers began to include them in their automation purchase specifications for 300mm equipment, and at this writing, EDA interfaces are available on every type of equipment required for the most advanced manufacturing processes.

## 2. Terminology and Acronyms

| Term or Acronym | Description |
|---|---|
| Access Control List (ACL) | Part of E132, the Client Authorization details that grant or deny client sessions and impose restrictions on client access to specific EDA information and operations. |
| CEM | E120.1 XML Schema for the Common Equipment Model |
| Client Authentication | In order for an EDA client to gather data, E132 requires clients to first establish a session. The client must provide credentials to the EDA Server. The EDA Server must be preconfigured to grant the client permission (based on the credentials) to establish a session. |
| Client Authorization | Part of E132, before the EDA Server accepts an operation request from the client, the Server must verify that the client has permission. These permissions are preconfigured in the Server using Access Control List entries. |
| Client Consumer | An EDA client that receives the Data Collection Reports and other E134 consumer operations. |
| Client Manager | An EDA client that establishes a session, identifies the Consumer to receive the data, sets up Data Collection Plans, and uses other E134, E132, and E125 manager operations. |
| Data Collection Plan (DCP) | Part of E134, a data gathering request that includes a set of events (with a configurable set of parameters), Exceptions (with a fixed set of parameters), and Traces (with a configurable set of parameters). After being successfully created and stored on the equipment, a Data Collection Plan must be activated. Then the client will receive the respective Data Collection Reports as configured in the plan. |
| Data Collection Report (DCR) | The equipment sends the requested data as specified in the Data Collection Plan in this standard format. |
| DCM | E134.1 Specification for SOAP Binding of Data Collection Management |
| ECA | E132.1 Specification for SOAP Binding for Equipment Client Authentication and Authorization |

| | |
|---|---|
| EDA | Equipment Data Acquisition. The combination of SEMI standards E120, E125, E128, E132, E134, E138 and E164. This term is more common than "Interface A", but both refer to the same set of SEMI standards.<br><br>Note: the EDA acronym is also commonly used in the semiconductor industry to mean Electronic Design Automation, which is unrelated to this document or the referenced SEMI standards. |
| EDA Client (Interface A Client) | Software that uses the equipment's EDA interface by establishing a session. Typically, this is developed by the IC Maker or a third-party software provider to the IC Maker. Equipment suppliers can also develop EDA clients, such as for capturing diagnostic information. |
| EDA Server | The equipment software that implements the EDA standards. This software should be installed on the equipment's internal computer and fully integrated into the equipment's control system. However, the software can be run on an external computer to retrofit an EDA interface on equipment that does not have an integrated solution.<br><br>This is also called an EDA Web Server or Interface A Server. |
| Equipment | Equipment refers to the EDA Server that implements the interface for the manufacturing hardware and software interface to the EDA clients. |
| ESDS | E125.1 Specification for SOAP Binding for Equipment Self-Description |
| Exception | Part of E134, an equipment alarm, error, or warning notification. |
| Event | Part of E134, a notification that something important occurred on the equipment, often associated with a specific state machine. An Event Request in a DCP can include any set of parameters. |
| FF | Fire-and-Forget messaging paradigm in which a message sender does not expect a reply message. |
| Host | The software an IC Maker is running to communicate with the equipment. A host typically refers to the SECS/GEM connection but could also refer to a client using EDA. |
| HTTP | HyperText Transfer Protocol: the protocol for moving hypertext files across the Internet. Requires an HTTP client program on one end, and an HTTP server program on the other end. HTTP is the most important protocol used in the World Wide Web (WWW) |
| HTTPS | HyperText Transport Protocol (Secure): the standard encrypted communication mechanism on the World Wide Web. This uses the Secure Socket Layer (SSL) as a layer under its regular HTTP application layering. |
| Interface A | Synonymous with EDA in a SEMI standards context. |
| Interface A Client | Same as EDA Client |
| Metadata | In an EDA context, this is information that describes the data structures and behavior supported by an equipment supplier's implementation of EDA, such as when an event occurs or the interpretation of a parameter's value. |

| | |
|---|---|
| Operation | An EDA transaction, method, or message initiated by an EDA client or server. Each operation has a name, well-defined format and meaning. E125, E132, and E134 each define a set of operations for the client and server. |
| Parameters | The set of data available for gathering from the equipment's EDA interface. |
| RR | Request-Response message paradigm in which a message sender expects a reply message. |
| Security Admin | A part of E132, Security Admin is a utility provided with the EDA Server to provide administrative configuration capability. |
| Session | Also called an Authenticated Session, a session is established between the server and client by following the E132 procedures. Once a session is established, the client can send authorized operation messages. |
| SOAP | Simple Object Access Protocol (SOAP). In order to make the EDA standards easier to implement, the SOAP protocol was used. It is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML-based protocol that consists of three parts:<br>• an envelope that defines a framework for describing what is in a message and how to process it<br>• a set of encoding rules for expressing instances of application-defined data types<br>• a convention for representing remote procedure calls and responses.<br>SOAP can potentially be used in combination with a variety of other protocols; however, the only bindings defined in this document describe how to use SOAP in combination with HTTP and HTTP Extension Framework.<br>See www.w3.org/TR/2000/NOTE-SOAP-20000508 for more information. |
| Tool | A synonym for Equipment, frequently used in the standards. |
| Trace | Data polling performed by the equipment as defined by the client. The client defines the polling frequency, the set of polled parameters, and the conditions to start and stop polling. |
| UML | Unified Modeling Language (UML): All EDA standards use UML notation for class diagrams and for object-oriented diagrams provided as examples.<br>UML is a notation for representing object-oriented designs and views created by Booch, Rumbaugh, and Jacobson to merge their three popular notations and aspects of other existing notations into a single object-oriented notation. UML is an open modeling language to specify, visualize, design, and document models of software systems.<br>See www.uml.org/ or www.omg.org/technology/documents/modeling_spec_catalog.htm for more information. |

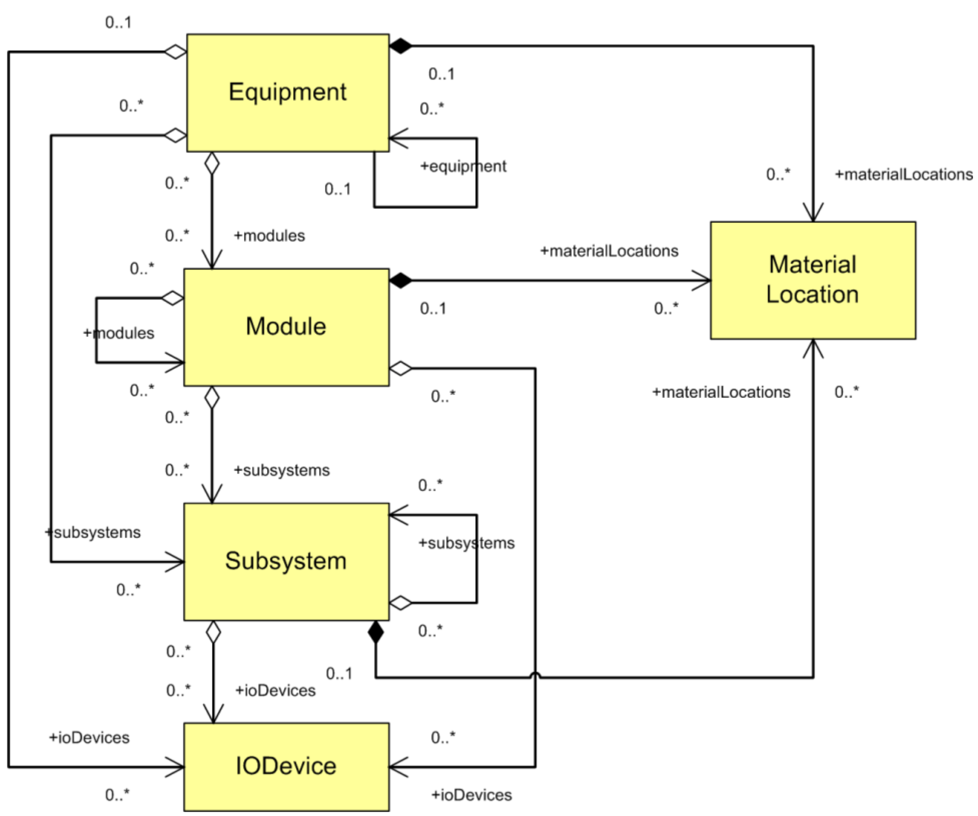
| | |
|---|---|
| XML | Extensible Markup Language: A markup language used for representing data rich with context and content in documents and in communications. XML is an extension of SGML, a document-oriented markup language. It was created by W3C for use on the Internet.<br><br>See www.w3.org/TR/REC-xml for more information. |
| XML Schema | An XML Schema defines the structure, content and semantics of XML documents. |

# 3. SEMI Standards

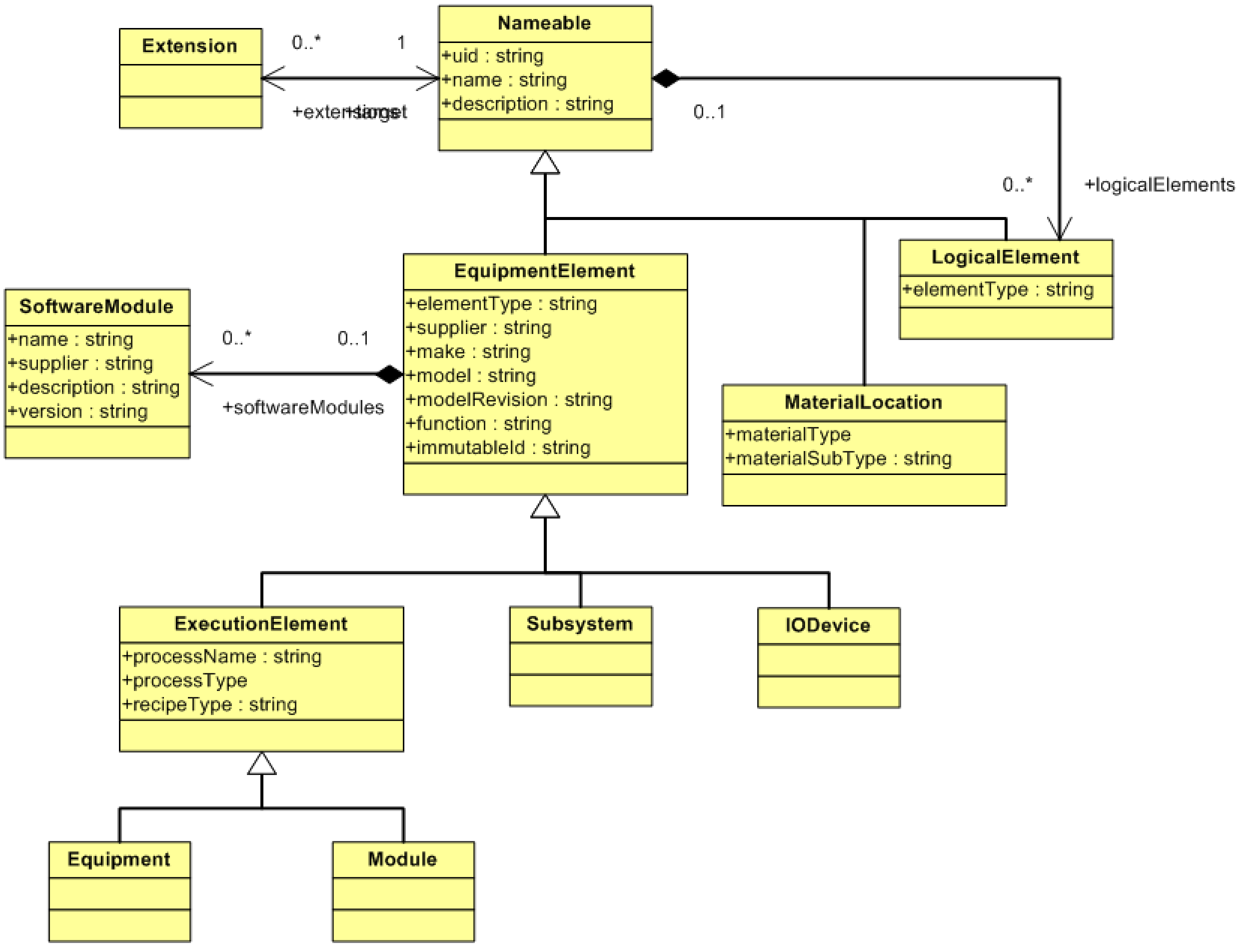
EDA consists of multiple SEMI standards including the following specifications. All are available for download from the SEMI website for a fee, http://semi.org/en/Standards. The EDA schema files are available for free from the SEMI website.

## 3.1 E120 Specification for the Common Equipment Model (CEM)

E120 provides a general object model that represents an external view of equipment. The model is composed of various classes organized in a logical hierarchy. A fully implemented model summarizes all of the equipment's major hardware and software components. SEMI standard **E120.1 XML Schema for the Common Equipment Model (CEM)** maps the E120 standard into a specific XML implementation.



| Class | Description |
|---|---|
| Equipment | Models the equipment as a whole and contains MaterialLocations, Modules, Subsystems, and IODevices. |
| Module | Models a major equipment subsystem that can process material and execute a recipe. For metrology and inspection equipment, 'process' means measure some physical or electrical parameter on the material or locate and classify defects on the material.<br>A module may contain MaterialLocations, Modules, Subsystems, and IODevices. |
| Subsystem | Models major equipment subsystems that cannot process material. It may contain MaterialLocations, Subsystems, and IODevices. |
| IODevice | Models sensors, actuators, or intelligent actuator/sensor devices. |
| MaterialLocation | Models the ability of an equipment component to hold material. Specifies the material type such as Carrier, Substrate, or ProcessDurable (e.g., a Fixture). |

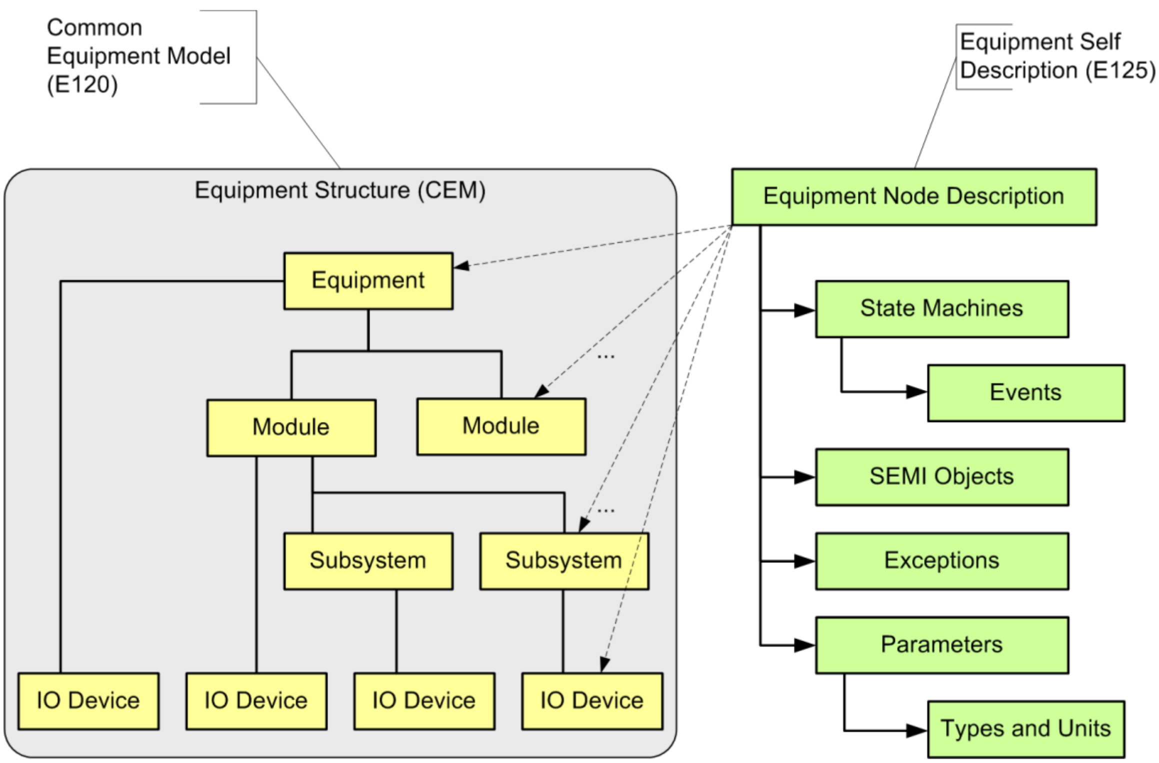



| Class | Description |
|---|---|
| SoftwareModule | Describes the existence and version of software that is in use on the equipment and on equipment components. The software might be supplied by the equipment supplier or a third-party. Attributes identify the supplier, version, and a description. |
| EquipmentElement | An abstract class representing the basic information required for each hardware component, such as a Module, Subsystem, or IODevice. Attributes identify the hardware’s supplier, make, model, and revision. |
| ExecutionElement | A subclass of EquipmentElement that models the parts of the equipment structure capable of processing material, namely Modules and Equipment. The attributes specify the process type as Measurement, Process, Storage, or Transport. This is called an AbstractModule in version 1105. |
| Nameable | The root of every class that provides unique identification and a description for each component. |

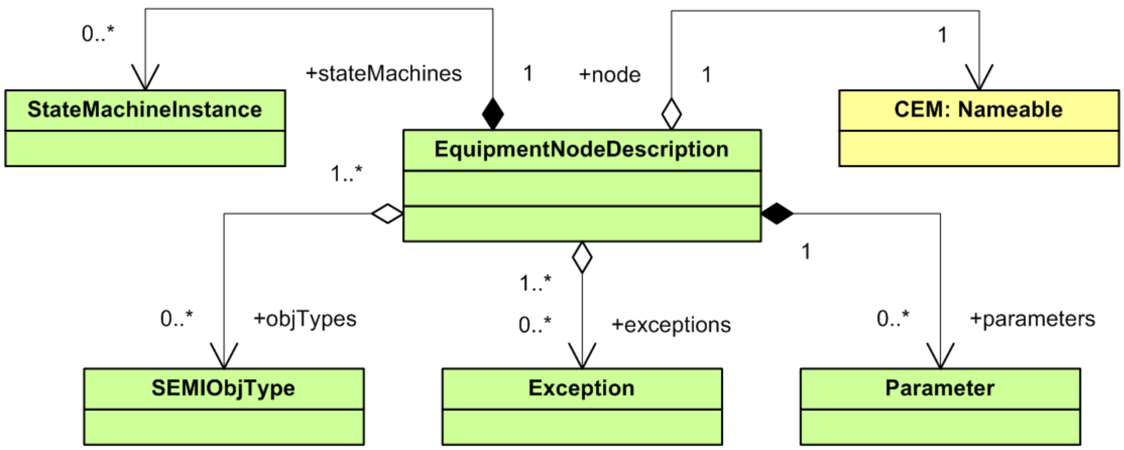
| | |
|---|---|
| Extension | Provides the ability for other standards and specific implementations to extend the Common Equipment Model. |
| LogicalElement | A non-structural element in the equipment. This is part of the 0710 and later versions, but not 1105. The LogicalElement is used to improve Equipment Model organization.<br>Examples of LogicalElement required by SEMI E164 (EDA Common Metadata) include MaterialManager and JobManager. |

## 3.2 E125 Specification for Equipment Self Description (EqSD)

The E125 standard allows clients to request complete information about all information available for data gathering, including the parameters (specific data, units, and types), events, exceptions, state machines, SEMI E39 data, and the E120 physical configuration. All available information is mapped into the E120 Common Equipment Model object hierarchy. SEMI standard **E125.1 Provisional Specification for SOAP Binding for Equipment Self Description (ESDS)** maps the E125 standard into a specific SOAP/XML implementation.



The E125 standard allows the end user to know exactly what information can be monitored on the equipment without having to rely on the equipment's documentation. Each parameter, event, and exception is assigned to an E120 node to provide context and hardware association. EDA clients can implement plug-and-play methodology, automatically utilize new information immediately after the equipment's interface is revised and implement graceful error recovery if data is unavailable unexpectedly.




E125 defines the EquipmentMetadataManager interface with client-initiated operations to query the Equipment's available metadata information and to know what data can be collected. The equipment must implement this interface.

| Operation | Description |
|---|---|
| GetUnits | Retrieve all available unit definition information. |
| GetTypeDefinitions | Retrieve all available type definition information. |
| GetStateMachines | Retrieve all available state machine information and the associated events. |
| GetSEMIObjTypes | Retrieve all available SEMI E39 object information. |
| GetExceptions | Retrieve all available exception information. |
| GetEquipmentStructure | Retrieve all SEMI E120 Common Equipment Model information. Each component in the equipment structure is a node. Clients use GetEquipmentNodeDescriptions to get details about what data can be collected from each node identified in the equipment structure. |

| | |
|---|---|
| GetEquipmentNodeDescriptions | Retrieve one or more equipment node descriptions including the associated state machines, SEMI E39 information, exceptions, simple events, and parameters. A parameter represents an element of data that can be gathered. Its attributes describe its meaning, unit, and the type (integer, real, string, Boolean, array, enumeration, or some other composite type). A parameter's attributes also describe if the data's value is available with or without restrictions. Events, exceptions, and parameters can list associated parameters. |
| GetLatestRevision | Retrieve the date and time the available metadata information changed. |
| NotifyOnRevisions | Request the equipment to send notification when any metadata availability changes. |

E125 also defines the MetadataClient interface with an equipment-initiated operation for consuming clients. Each client must implement this interface.

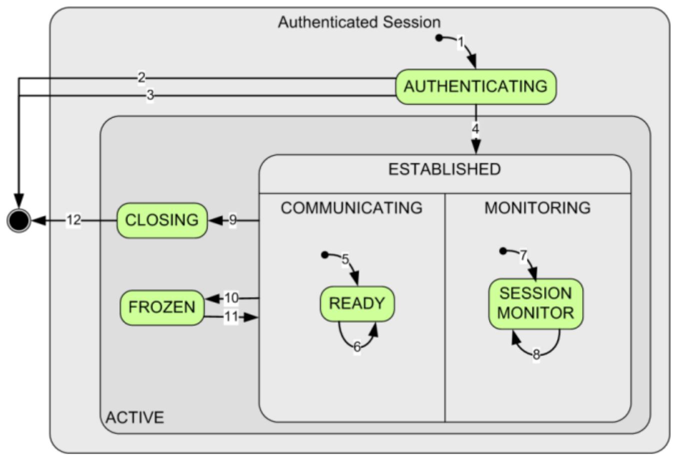
| **Operation** | **Description** |
|---|---|
| MetadataRevised | Notify the client that the Metadata in the Equipment Model has changed if NotifyOnRevisions is enabled. |

## 3.3 E132 Specification for Equipment Client Authentication and Authorization

E132 defines two related security features for EDA messaging: Client Authentication and Client Authorization. Client Authentication determines how the client establishes a session before it can do anything else. Client Authorization manages what the client can access after the session is established. The equipment must provide a Security Admin for configuring the Client Authentication and Authorization after installation in the fab. SEMI standard **E132.1 Specification for SOAP Binding for Equipment Client Authentication and Authorization (ECA)** maps the E132 standard into a specific SOAP/XML implementation.

In order for an EDA client to establish a session where it may use E125 or E134 service requests, the client must provide credentials and be authenticated, and the equipment must be in the ALLOWED Session Establishment state. Credentials include a client ID, an encrypted session key, and an encrypted client ID proof-of-identity key. Any attempts to use services requested before the authentication are rejected.

Once the session is established, the client authorization becomes effective based on the client's credentials while establishing the session. EDA communication can be established using either the HTTP or HTTPS (SSL) protocol. Client authentication is implemented by means of SSL certificates; therefore, there is no client authentication when using the HTTP protocol which has no certificates. In practice, most EDA implementations use the HTTP protocol presumably because the factory networks are generally secure and managing certificates is extra work.




E132 defines the SessionManager interface which includes the following client-initiated operations to establish communication. The equipment must implement this interface.

| Class | Description |
|---|---|
| EstablishSession | Request to establish a new authenticated session and to set the client endpoint, the consumer for all notifications. |
| PersistSession | Request the equipment to maintain the session, even after shutting down the equipment. In versions 0710 (Freeze II) and later, this operation does not exist because all sessions are persisted. |
| EnhancedEstablishSession | Same as EstablishSession, with the ability to specify specific session capabilities.<br>This is available only in version 0710 (Freeze II) and later. |
| SessionPing | A check to see if the equipment is still active. |
| CloseSession | Request to terminate the session. |

Authorization is configured using Access Control Lists (ACLs). An ACL is a collection of ACL entries where each entry gives the client access permission to something in the interface. The E132 standard uses the terms Principal, a client defined in the ACL, and Privilege, permission to use an operation or access specific data. In practice, the easiest way to setup the ACL is by defining what E132 calls roles and then assigning clients to one or more roles. For example, it might be convenient to define "Operator", "Technician", and "Engineer" roles. Then EDA client applications can be assigned to these roles to give them the appropriate access level.

E132 defines the SessionClient interface which includes some equipment-initiated operations for clients. Each client must implement this interface.

| Operation | Description |
|---|---|
| SessionPing | Used by the equipment to check if the client is active. |
| SessionFrozen | Notification to the client that the session is frozen. |
| SessionClosed | Used by the equipment to close an active session. |

E132 also defines the SecurityAdmin interface which includes the following operations. Only one active Security Admin client session is allowed at a time. The SecurityAdmin client is identified by a unique principal as defined by the equipment.

| Operation | Description |
|---|---|
| GetDefinedPrivileges | Request the list of all defined privileges. |
| GetACL | Request the list of all defined Access Control Lists |
| AddACLEntry | Add a new ACL entry |
| DeleteACLEntry | Delete an existing ACL entry |
| GetActiveSessions | Request the list of information on all active sessions |
| SetMaxSessions | Sets the maximum number of active sessions |
| GetMaxSessions | Requests the maximum number of active sessions |

In version 0710 and later, E132 defines the InterfaceDiscovery interface which allows the EDA client to query a list of the equipment's web service URLs. This is useful to the client because the exact URL is equipment defined. With InterfaceDiscovery, the client only needs to know the InterfaceDiscovery URL: therefore, it is easier to establish an EDA session.

| Operation | Description |
|---|---|
| GetInterfaces | Request the list of all available web services on the equipment. The list includes a name, URL, description, and which SEMI standard and version is implemented by each web service. |

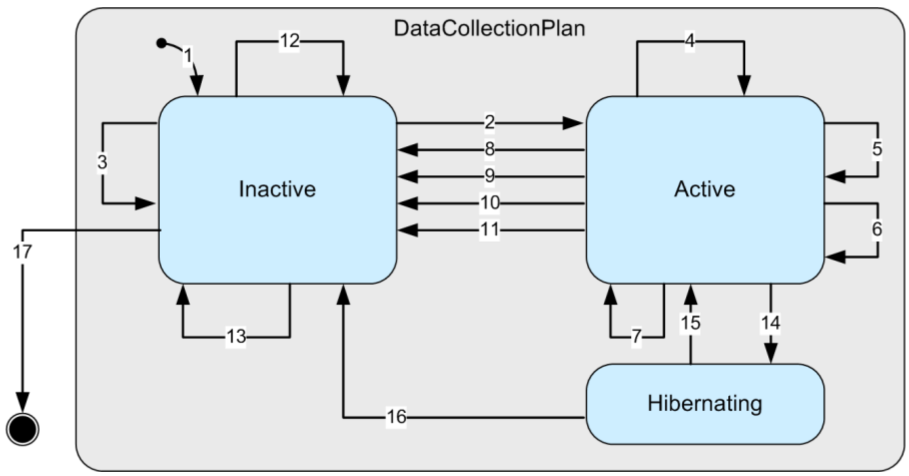
## 3.4 E134 Specification for Data Collection Management

The E134 standard defines several methods for EDA clients to acquire E125 information from the equipment. A client can request data ad hoc where the requested set of data is returned immediately. Typically, however, the client will define Data Collection Plans (DCPs) to configure the desired data collection approach. Once DCPs are activated, the equipment sends Fire-and-Forget Data

Collection Reports (DCRs) as the data becomes available. If the optional buffering is enabled, then the DCRs are buffered at the equipment and transmitted periodically at the end of each

buffering interval. Buffering has been shown to allow much more data to be collected, presumably due to the reduced messaging overhead. If permissions allow, clients can activate DCPs defined by other clients.

E134 also defines how to manage Data Collection Plans when either the equipment or client shuts down and restarts. Specifically, DCPs that are supposed to restart automatically are defined as "persistent." SEMI Standard **E134.1 Specification for SOAP Binding of Data Collection Management (DCM)** maps the E134 standard into a specific SOAP/XML implementation.




Data Collection Plans can include any number and combination of traces, events, and exceptions to report when the DCP is activated. Traces, events, and exceptions that are not listed in activated Data Collection Plans are not sent to any clients. Therefore, the equipment supplier determines how much data is available, but the clients determine how much data is gathered and reported. Bandwidth requirements are based upon the number of clients, the number of activated DCPs, the number of Event Requests, Exception Requests, and Trace Requests in each DCP, the number of parameters defined for each Event Request and Trace Request, the frequency of activated events and exceptions, and the frequency of trace data collection. Therefore, the precise EDA bandwidth requirements are difficult to determine. Clients can potentially submit enough data collection requests to overload the equipment's interface computer and affect the equipment's throughput. Therefore, E134 also defines a Performance Status operation to notify consumers of an overload condition and allows the equipment to take appropriate action.

E134 defines the DataCollectionManager interface, which includes the following client-initiated operations for setting up data collection. The equipment must implement this interface.

| Operation | Description |
|---|---|
| DefinePlan | Submit a Data Collection Plan. This includes a set of Trace Requests, Event Requests and/or Exception Requests. |
| GetDefinedPlanIds | Request a list of all Data Collection Plan IDs. |
| GetPlanDefinition | Request the definition of a Data Collection Plan. |
| ActivatePlan | Activate the specified DCP. |
| GetActivePlanIds | Request a list of all activated DCP IDs. |
| DeactivatePlan | Deactivate the specified DCP. |
| DeletePlan | Delete the specified DCP. |
| GetParameterValues | Ad hoc request to retrieve the current value of one or more E125 parameters. |
| GetObjTypeInstanceIds | Request a current list of unique instance IDs for one or more E39 ObjTypes. |
| GetCurrentPerformanceStatus | Retrieve the current equipment performance status. |
| GetCurrentDateTime | Request the current date and time on the equipment. This is available only in version 0710 and later. |

E134 also defines the DCPConsumer interface which includes the following equipment-initiated, Fire-and-Forget operations. Each client must implement this interface to receive data collection reports.

| Operation | Description |
|---|---|
| NewData | Data Collection Report from an active DCP. This includes trace, event and/or exception data. |
| PerformanceWarning | The equipment has detected performance degradation. |
| PerformanceRestored | The equipment has detected a return to normal conditions. |
| DCPDeactivation | Notification that an active DCP for that consumer is deactivated. |
| DCPHibernation | Notification when one or more persisted DCPs are put into the hibernation state as part of equipment shutdown. |
| DCPDefined | Notification when a DCP is defined by another client. This is available only in version 0710 (Freeze II) and later. |

| | |
|---|---|
| DCPDeleted | Notification when a DCP is deleted by another client. This is available only in version 0710 (Freeze II) and later. |

### 3.4.1 Events

Events provide a convenient mechanism to receive notification when something important happens and optionally gather data with the Event Report. Events can be either part of a state machine or stand alone. In the Data Collection Plan, the client chooses E125 parameters to include in the Event Report. Each time the event occurs thereafter, an Event Report is generated and sent.

### 3.4.2 Exceptions

Exceptions notify the client when errors, warnings, or alarms occur on the equipment. The Exception Report optionally includes data, but the set of included data is fixed by the equipment supplier and listed in the equipment metadata model. Clients cannot choose which parameters are included with Exception Reports like they can with Event Reports.

### 3.4.3 Traces

Traces provide a convenient mechanism for continuously or intermittently polling data at a constant rate. A Trace Request includes a data gathering frequency, a set of E125 data parameters to collect, a start condition, and a stop condition. This configures the equipment to poll the data at the specified frequency and send it to the client. Start and stop conditions, also called start and stop triggers, are optional. When specified, they include a set of one or more events and/or exceptions. A start trigger determines when to start collecting the trace data at the specified frequency. A stop trigger determines when to stop collecting the trace data. Otherwise, the trace data collection begins as soon as the Data Collection Plan is activated by the client and ends when deactivated by the client or when the specified number of reports have been generated and sent.

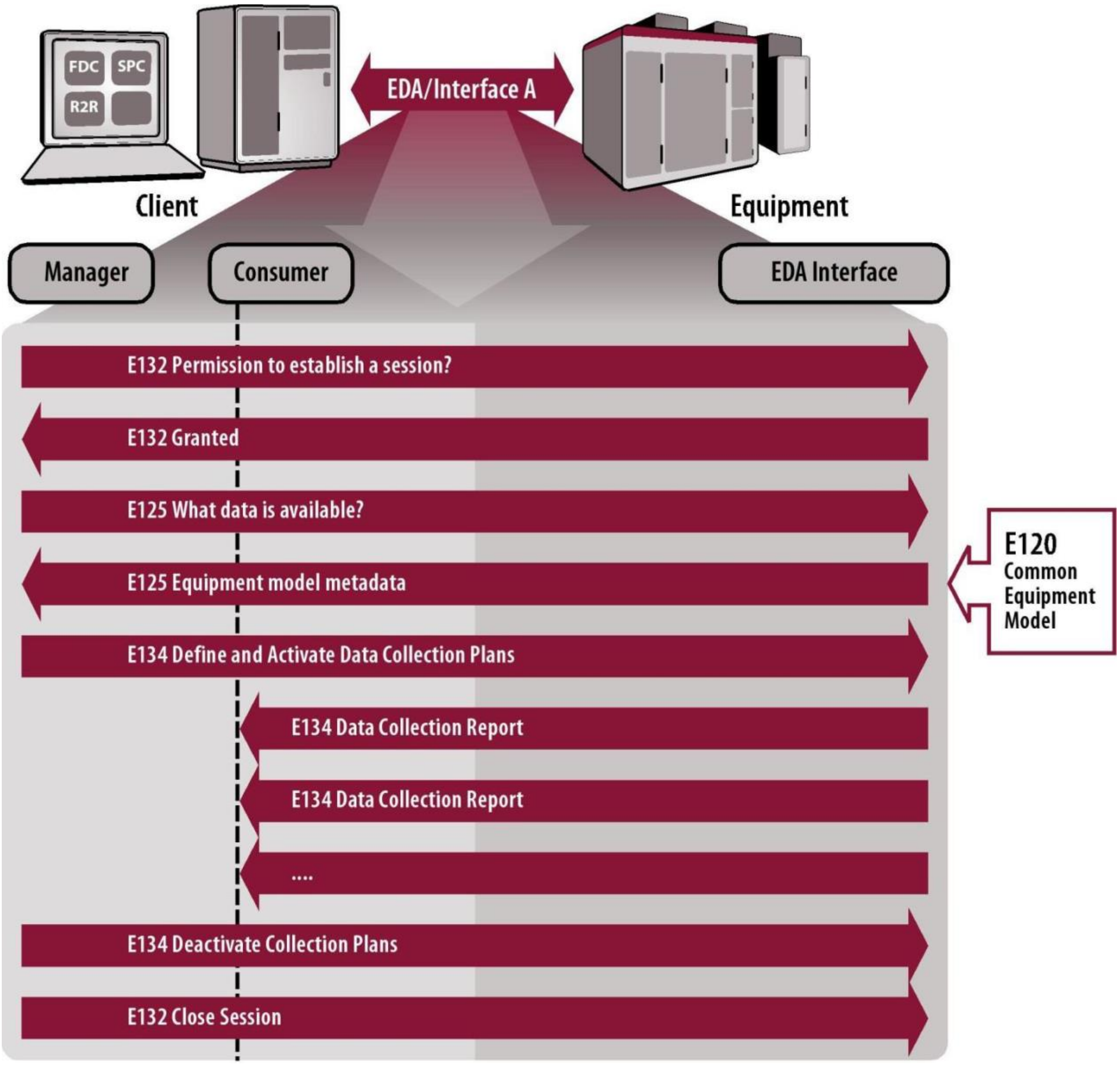
In the 0710 version (Freeze II), start and stop triggers can also include a conditionValue, which includes a parameter ID, comparison operator, and a parameter value for comparison. When the start or stop trigger's event or exception occurs, the specified parameter's current value is compared with the specified value. If the comparison results in a true statement, then the start or stop trigger is in effect. Otherwise if the comparison results in a false statement, then the start or stop trigger is not in effect. The conditionValue feature is particularly effective when implemented with SEMI standard E157 (Specification for Module Process Tracking). For example, this conditionValue feature could be used to start trace data collection when recipe step is greater than 10 and stop trace data collection when the recipe step is greater than 40.

## 3.5 E164 Specification for EDA Common Metadata

Unlike the other EDA standards in which most of the features are generally applicable to any manufacturing equipment, the E164 standard targets equipment that implement the GEM300 SEMI standards. The E120 and E125 standards are very flexible and allow equipment models to be implemented in many different styles that can vary tremendously. E164 specifies additional requirements on how to represent manufacturing equipment with E120 and E125 features so that equipment modeling styles are more consistent. Additionally, E164 includes XML files that define exactly how the SEMI GEM300 standards should be modeled in EDA, including E30, E40, E87, E90, E94, E116, E148, and E157. This describes exactly how to represent the state models from the standards and exactly what the event, parameter, and exception names should be. This additional standardization in E164 makes it much easier for end users to browse equipment models and set up data collection by making EDA interfaces from different equipment suppliers more consistent.

## 3.6 EDA Client Operations

The following diagram demonstrates the basic operational flow using the individual EDA standards together.




Initially, each client must establish an authenticated session. Next the client can ask for the Equipment Model metadata information to see what data is available through EDA. With this information, the client can define and activate Data Collection Plans. The equipment will then continue to generate Data Collection Reports until the client deactivates the plans or becomes unavailable. When shutting down properly, the client should close the session properly.

# 4. Frequently Asked Questions (FAQ)

## Where can I get more information?

More information is available from the following websites:

SEMI: www.semi.org

Cimetrix: www.cimetrix.com

## Will the EDA standards change over the next few years?

The EDA standards have been modified on a regular basis as IC makers have used them in production operations and identified areas for improvement. The standards will continue to adapt to the needs of the semiconductor manufacturing community and evolve over time. Anyone who implements the EDA standards must plan to dedicate resources to follow the standards and migrate their implementations as the inevitable changes and improvements are approved.

## Why Use EDA Instead of SECS/GEM for data collection?

Here are a few of the reasons. SECS/GEM has its merits, too, but they are listed elsewhere.

| SECS/GEM | EDA |
|---|---|
| SECS/GEM implementations typically support only one client (host) connection. IC makers cannot run several data gathering applications at the same time without an infrastructure to share the data. | EDA supports multiple, independent concurrent clients. |
| SECS/GEM is only partially self-describing and therefore relies on good documentation. IC makers have complained that the documentation is often not well maintained. | EDA is self-describing through the E125 metadata standard. |
| SECS/GEM data is relatively flat and unorganized. The IC Maker must study the documentation, hardware, software, and processing to understand how to organize the data. | EDA presents the data in a hierarchy, organized by the major hardware components. It is relatively easy to determine data context and meaning from the E120 and E125 metadata model. |
| Data in a SECS/GEM message is highly structured and relatively inflexible. | Because EDA uses XML, it is inherently designed to accommodate additional metadata. |
| SECS/GEM is used extensively in only a few industries, which limits the number of experts in the world. | SOAP/XML and HTTP are the backbone of most Internet and Intranet applications. There are many programmers worldwide that are familiar with this technology. Most software development tools easily handle SOAP messaging, such as Java and Microsoft's Visual Studio. |

| | |
|---|---|
| In most factories, the SECS/GEM interface is used for equipment configuration and control, and therefore subject to extensive oversight and lengthy change procedures. It can take several weeks or longer for any data collection changes to be approved and implemented. | EDA is independent of the GEM interface and therefore immediately available for the dynamic data collection needs of the process engineers. New data collection requirements can be implemented as they are identified. |
| Data collection through a SECS/GEM interface tends to be set up such that the targeted data is collected all of the time. | With start triggers, stop triggers and conditionValue features in a Trace Request definition, trace data collection can be focused on collecting the targeted data only when it is meaningful to do so.<br><br>Collecting all the data all the time becomes increasingly difficult and unmanageable as data is collected at faster rates and increases in volume and granularity. |

EDA uses many of the same concepts as SECS/GEM. Here is a mapping between the comparable concepts and technologies in these two standards.

| SECS/GEM | EDA |
|---|---|
| Status Variables, Equipment Constants, and Data Variables | Parameters |
| Alarms | Exceptions |
| Collection Events (S6,F11) | Event Requests in Data Collection Plans |
| E39 Objects | SEMIObjType and Instance IDs |
| Trace Data Collection | Trace Requests in Data Collection Plans |
| Reports | Data Collection Reports |
| State Machines | State Machines |
| Enable/Disable Collection Events | ActivatePlan/DeactivatePlan |
| Define and Link Reports (S2,F33 & S2,F35) | DefinePlan |

## Is the SOAP/XML over HTTP fast enough?

Yes, if it is implemented effectively. Cimetrix has run multiple tests and prototypes that demonstrated excellent performance that exceeds industry requirements. The results have been presented in industry public forums.

## What data should be made available through EDA?

Ultimately, each equipment supplier must negotiate the requirements from the IC makers to determine exactly what data they need. Here are some guidelines for equipment suppliers:

- As many sensors and actuators as possible

- Processing or inspection setup information, states, sub-states and results
- Material tracking information at the individual substrate and material location level
- Provide all data as nearly as possible directly from the source. This may require some software and hardware architectural changes.
- Predefine some Data Collection Plans that include the most useful data gathering configurations (called "Built-in Plans in E134). This will establish bandwidth capacity guidelines for the IC makers and help them to use the equipment most productively.

The most recent equipment purchase specifications from leading IC makers are very prescriptive about the data required, so the mantra going forward should be "when in doubt about the usefulness of a particular data item or event, include it in the model."

## How fast will data be collected through Trace Requests?

IC Makers have set an expectation that most data will be collected as fast as 10 Hz. However, some critical data might need to be collected at faster rates such as 20-50 Hz. Just because data will be collected at these rates under some conditions does not mean that it will be collected all the time at that rate. EDA allows for trace data collection to begin and end using the start and stop triggers. This allows for focused data collection where the client only collects data when the data is important to analyze.

## What should be reported when the requested data is not available?

E134 defines a NoValue class with a ValueNotAvailable value for the NoValueReasonEnum enumeration to handle this situation.

## How can a client gather SEMIObjType information, since the object IDs are not in the equipment metadata model?

SEMIObjType objects can be created and deleted dynamically or exist statically on the equipment. Either way, EDA clients manage them using the same operations. Dynamic objects include carriers and substrates. Static objects include substrate locations not in the carrier and Equipment Performance Tracking modules.

First, a client should use E125 operation GetSEMIObjTypes to query the list of available SEMIObjType classes. These are part of the Equipment Model. Then use E134 operation GetObjTypeInstanceIds to get full instance ID information for the current set of objects. With these instance IDs, the client can use any of the data gathering operations such as DefinePlan or GetParameterValues to query the object's attribute data. If the object is somehow deleted, then the client will receive a NoValue with a NoSuchParameter reason enumeration.

## How does an EDA Client receive data from the Equipment?

The Client must implement a web server with a single URL that implements the equipment-initiated E125, E132 and E134 operations. During the EstablishSession operation, a client passes its web server URL to the Equipment. When data collection plans are activated, the Equipment uses the NewData operation to pass the Data Collection Reports to the client.

## How can I Implement EDA?

The most cost-effective approach is to purchase a solution from a third-party supplier like Cimetrix. Cimetrix offers a state-of-the-art EDA development package called CIMPortal Plud which is for equipment suppliers and IC makers.

It is important to consider the following technical and commercial challenges when choosing a solution.

### Evolving Standards

There is no question that the standards will change several times over the next few years. A reliable solution not only complies with today's standards, but also is designed to anticipate and accommodate future changes. The equipment supplier's software team must be committed to become experts in the standards, actively track the changes, and provide timely upgrades.

### Data Source Management

You must ask these crucial questions; "Where does the data reside?" and "What is the most efficient way to get the data into an EDA interface?" All data will not be in one location. The EDA solution must have the ability to channel data directly from different sources.

### Data Integrity

The EDA solution must carefully manage data integrity. Inevitably, there is a time lag between data gathering in the equipment and data reporting to the interested consumers. Some data must be refreshed in every Data Collection Report. Other data can be cached for limited periods of time. Still other data can be perpetually cached. To maximize EDA performance, the solution must be able to manage data integrity and make correct data-caching decisions.

### Performance

Many IC makers expect EDA to achieve data rates of 100+ variables per chamber at rates up to 10 Hz. In EDA purchase specifications, leading IC makers have even requested the ability to gather 1,000 to 2,000 parameters for some chambers using data collection rates from 5 to 20 Hz, with a total interface bandwidth approaching 50K parameters per second. These rates are only possible with a software architecture designed for high performance.

### Software Quality

EDA has now become a valuable feature for the IC makers because it is the principal data source for their mission critical manufacturing applications, such as Fault Detection and Classification (FDC). In fact, a robust EDA interface will likely become one of the critical features that IC makers will validate before making a decision to purchase a particular unit of equipment. The EDA solution must achieve the highest software quality and performance standards. By contrast, a second-rate solution will jeopardize equipment sales.

### Experience

A professional EDA solution requires experience. Experience means running on real machines in production for long periods of time. Experience means having prototypes and demonstrations with simulators and real machines. Experience gives a solution the maturity to implement not only the stated requirements, but to understand and support the intent of the standards with features that go beyond the IC makers' expectations.

### Customer Dedication

Commercial solution providers must be committed to providing effective customer support. Check references and review past experiences. Ensure that the company is truly dedicated to its customers.

## What are EDA/Interface A Freeze Versions?

Because the EDA standards continue to evolve and change, SEMI initiated the concept of Freeze Versions. A Freeze Version enables equipment suppliers to develop new systems that can be installed into factories while IC makers develop new data collection client applications and SEMI continues to evolve the EDA standards. Each freeze identifies a specific version of the individual EDA standards (E120, E128, E125, E132, E134, and E138) that equipment suppliers and IC makers can use as a unified, known-compatible set to implement the interface. The industry adopted the initial 1105 Freeze Version (Freeze I) in 2006, and then, four years later, the next 0710 Freeze Version (Freeze II) was announced that included many improvements and some new capabilities.

SEMI realized the Freeze Version concept by creating two downloadable bundles of standard schema files: one for the 1105 freeze version and another for the 0710 freeze version. These are available on the SEMI website www.semi.org.

## Which version of EDA should we implement?

Unless the client applications and Equipment are implementing the same version of the EDA standards, they cannot fully communicate and might be incompatible. This is the simply nature of SOAP/XML Web Services technology. Moreover, since individual equipment suppliers have different development roadmaps, they may not all support the same Freeze Version at the same time, so most factory systems will need to support multiple Freeze Versions. However, the best policy for an equipment supplier is to always implement the latest approved Freeze Version (at this writing, Freeze II). Consult the table below for the specifics of which version of each standard are part of which Freeze Version.

Freeze I (1105)

| Standard | Document Revision | Schema Namespace Version | WSDL Namespace Version |
|---|---|---|---|
| E120 Common Equipment Model | 1104 | E120-1.V1104 | N/A |
| E125 Equipment Self Description | 1105 | E125-1.V0305 | E125-1-V0305 |
| E132 Client Authentication and Authorization | 1105 | E132-1.V0305 | E132-1-V0305 |
| E134 Data Collection Management | 1105 | E134-1.V1105 | E134-1-V1105 |
| E138 Common Components | 0305 | V0305 | N/A |

Freeze II (0710)

| Standard | Document Revision | Schema Namespace Version | WSDL Namespace Version |
|---|---|---|---|
| E120 Common Equipment Model | 0310 | E120-1-V0310 | N/A |
| E125 Equipment Self Description | 0710 | E125-1-V0710 | E125-1-V0310 |
| E128 Specification for XML Message Structures | 0310 | E128-1-V0706 | N/A |
| E132 Client Authentication and Authorization | 0310 | E132-1-V0310 | E132-1-V0310 |
| E134 Data Collection Management | 0710 | E134-1-V0710 | E134-1-V0710 |
| E138 Common Components | 0709 | E138-1-V0305 | N/A |

Be sure to discuss with your customer which version of the standard they are implementing on the client side.

## What Are the EDA Standards Guidance Documents?

The two principal guidelines to assist in the development of the EDA standards on manufacturing equipment are SEMI standard E147 (Guide for Equipment Data Acquisition) and E164 (EDA Common Metadata).

Be sure to discuss with your customer what additional guidelines they may require for a specific factory implementation, especially regarding system performance and equipment metadata model content.

## What products are available for EDA testing?

Several software products are available for evaluating and testing an EDA interface, including the ECCE Plus, the EDATester and the Metadata Conformance Analyzer (MCA). The Equipment Client Connection Emulator (ECCE) was originally funded by SEMATECH and developed by Cimetrix early in the EDA standards development process and has since been significantly enhanced by Cimetrix to support both Freeze Versions I and II. The latest version of the EDATester also supports Freeze I and II and includes not only automated compliance testing for the EDA standards but also full interface performance characterization features. as well.

EDA developers may download and use the EDA 0710 freeze of the Metadata Conformance Analyzer at no charge. To obtain the 0710 MCA, visit https://www.cimetrix.com/metadata-conformance-analyzer-v1.2.

The ECCE Plus and EDATester are available for purchase from Cimetrix by contacting sales@cimetrix.com.

Note that some factories have also developed additional custom test applications that reflect their specific requirements and performance expectations.