Introduction to the SEMI Standards: GEM

Table of Contents

1. **Overview** 3

2. **Background** 4

2.1 Publication 4
2.2 Protocol Layer 4
2.3 Message Layer 4
2.4 Host Architecture 5
2.5 Widespread Industry Adoption 6
2.6 Publish/Subscribe Design Pattern with a Message Broker 6
2.7 GEM Scalability 7
2.8 Data Classifications 8
2.9 Data Types 8

3. **Fundamental Requirements** 8

3.1 State Models 8
3.2 Processing State Model 9
3.3 Collection Event Notification 9
3.4 Identification 10
3.5 Error Messages 10
3.6 Documentation 10

4. **Additional Capabilities** 12

4.1 Dynamic Collection Event Reports 12
4.2 Variable, Trace, and Status Data Collection 12
4.3 Self-Description 13
4.4 Alarms 13
4.5 Remote Commands 13
4.6 Equipment Constants 14
4.7 Recipe (Process Program) Management 14
4.8 Material Movement 14
4.9 Terminal Services 14
4.10 Clock 15
4.11 Spooling 15

5. **Frequently Asked Questions** 15

6. **GEM Terminology** 19

- NOTICE -

- COPYRIGHT -

- TRADEMARKS -

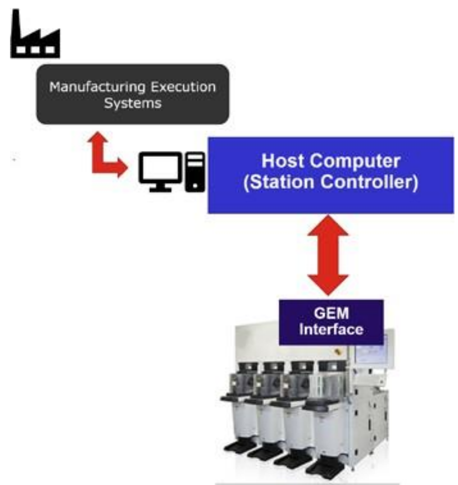
Cimetrix®
6979 High Tech Dr
Midvale, UT 84047
Phone: 801.256.6500
www.cimetrix.com

1 Overview

The GEM standard defines a communications interface that runs on manufacturing equipment and other intelligent devices. Factories use the GEM interface to remotely monitor and control the equipment or devices. For this reason, GEM is a key enabling technology for the global Smart Manufacturing and Industry 4.0 initiatives.

The GEM interface serves as a broker between the factory information and control software (host) and the manufacturing equipment embedded control system. Because the GEM standard is an open standard, anyone can develop a GEM-capable host or equipment software. If the interface software developers follow the GEM standard specifications correctly, all GEM host software packages are compatible with all GEM equipment implementations. A GEM-capable host can connect to any equipment GEM interface very quickly, start collecting data and use the GEM functionality.

Figure 1 Factory Host Computer connected to a single GEM Interface



A host computer might be dedicated to a single equipment's GEM interface, as depicted in Figure 1 above. When remotely controlling the equipment, the host will be integrated with the factory's manufacturing execution system (MES).

Alternatively, a host computer might be connected to multiple equipment as depicted in the Figure 2 below.

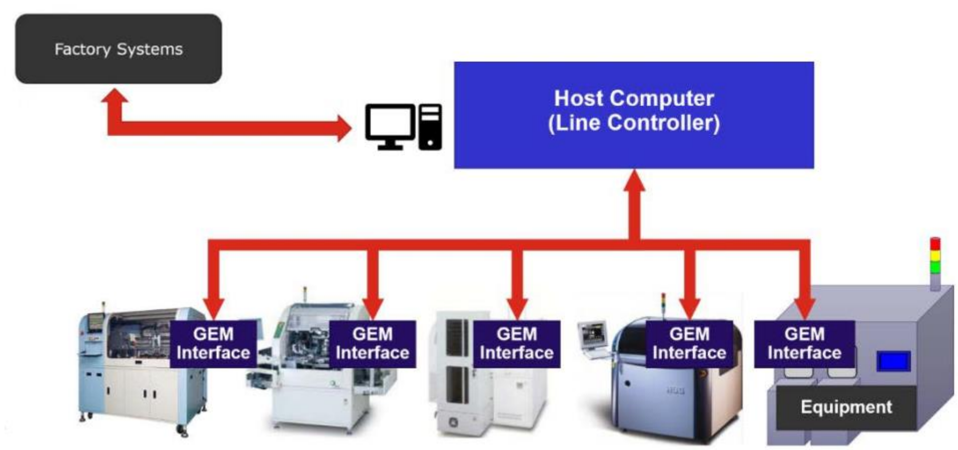


Figure 2 Factory Host Computer connected to multiple GEM Interface

GEM is the adopted technology by factories worldwide because it is mature and supports all the command and control features they require today and may need in the future. GEM allows them to use the same technology and software to integrate multiple equipment and process types, independent of supplier. The efficiency gained by implementing GEM technology allows factories to spend more time and money improving production operations and process capability, rather than developing custom integration for each type of equipment.

When equipment suppliers must support multiple protocols and external interfaces, focus is shifted from core equipment development. By using the same GEM interface for all end users, equipment suppliers can spend more time and money improving equipment quality and functionality. This time savings enables equipment suppliers to ensure they implement a complete, fully functional GEM interface.

2 Background

2.1 Publication

The GEM standard is published and maintained by the SEMI international standards organization (see http://www.semi.org) based in Milpitas, CA, USA. SEMI uses the standard designation "E30" to identify the GEM standard with the publication month and year appended as four numbers to designate a specific version. For example, E30-0418 identifies the version of the GEM standard published in April of 2018.

2.2 Protocol Layer

The GEM/SECS-II standards are protocol independent. Today, there are two protocols defined by SEMI: SECS-I (E4) for serial communication and HSMS (E37) for network communication. Not surprisingly, most systems today are using the HSMS standard. HSMS does not specify the physical layer. Any physical layer supported by TCP/IP can technically be used, but typically everyone uses an Ethernet network interface controller (NIC) with an RJ45 connector. When using the SECS-I standard, the messages size is limited to 7,995,148 bytes (about 8MB). HSMS maximum message size is limited to 4,294,967,295 bytes (about 4.3 GB).

2.3 Message Layer

The GEM standard is built on top of SEMI standard SECS-II (E5). The SECS-II standard defines a generic message layer to transmit any data structure and defines a of set of standard messages each with a specific ID, purpose and format. Every message sent and received is identified by a stream number (0-255) and function number (0- 255). For example, message Stream 1 Function 13, called S1F13, establishes communication. An odd-numbered function identifies a primary message. An even-numbered function identifies a secondary (reply) message. Together, the primary and secondary messages define a transaction. A reply message is expected to be returned before a timeout expires. This timeout is called the T3 timer. A late reply message is ignored.

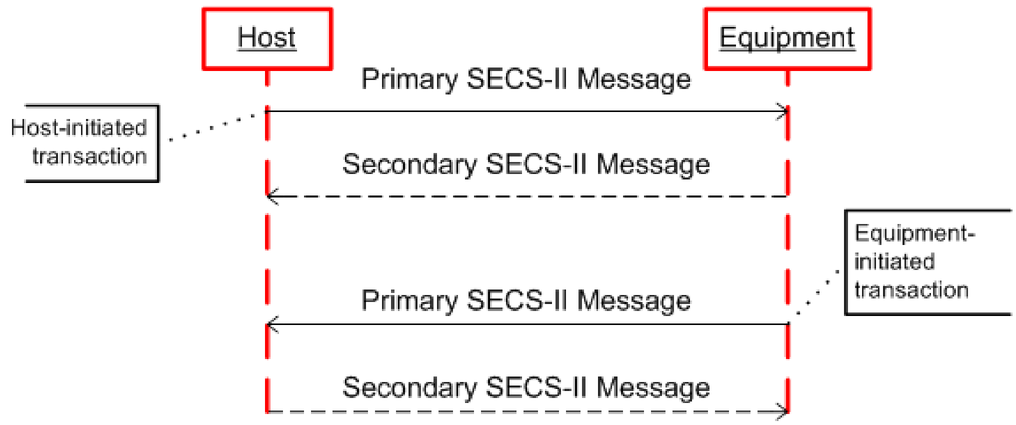


Figure 3 SECS-II Message Transactions

To minimize network bandwidth used, SECS-II messages are always sent as binary data. The structure of each standard SECS-II message is defined by the SEMI E5 standard. As specified in the message definition, some messages are only allowed to be sent by the host. Some can only be sent by the equipment. And some can be sent by the host or the equipment.

A message may be a simple data element, such as a binary response or an ASCII string. A message may\\\\ also be a complex list structure with multiple levels of lists in the hierarchy. A few me no data. The standard limits a single element within a SECS-II message to 16,777,215 bytes (about 16.5 MB).

For a SECS-II message to be valid, it must be initiated by the correct party and have the correct message format (i.e., the structure defined by SEMI E5 standard). The host and equipment can agree to support custom messages to implement custom features. The format of these messages is not defined in SEMI E5. However, this practice is highly discouraged when standard messages are sufficient.

The GEM standard specifies a subset of the E5 messages and defines several state machines and message scenarios (sequences) to ensure consistent behavior across equipment types.

2.4 Host Architecture

When using GEM technology, the software that talks to the equipment's GEM interface is called the host. The host software in often integrated with the factory's manufacturing execution system (MES). The host software is usually developed by the factory or purchased from a third party, but it can also be developed by an equipment supplier. In some instances, a factory will have a dedicated host computer for each equipment to interact with the equipment's GEM interface. This is typical when the equipment is extremely expensive and complex. In other systems a single host system will communicate with multiple equipment at the same time.

In traditional implementations of GEM, an equipment GEM interface can talk to one and only one host. However, many GEM implementations and products today support multi-hosting, in which the equipment supports simultaneous communication with more than one host. However, this is best viewed as one equipment having multiple GEM interfaces, because the data collection subscriptions are independent for each host. Multi-host functionality is particularly useful when the factory wants to run more than one host package. For example, they might have one host integrated with the MES system to control and configure the equipment, and another host (or more) that implement data collection specifically for fault detection or statistical process control (SPC).

A host does not have to comply with the GEM standard because GEM only sets the expectations for the equipment implementation. However, to make full use of the GEM interface, a host must implement the host side of the communication and send correctly formatted messages. The GEM standard defines clear equipment behavior expectations for each possible host message. Nevertheless, factories should strongly encourage GEM compliance by writing host software that expects GEM compliance and implements the host side to complement what is in the GEM standard. Factories should never demand features that violate or contradict the GEM standard.

2.5 Widespread Industry Adoption

The GEM standard is heavily used in numerous manufacturing industries across the world. The semiconductor front end, semiconductor backend (assembly/test), photovoltaic, electronics assembly, surface mount technology (SMT), high-brightness LED (HB-LED), flat panel display (FPD), printed circuit board (PCB) and small parts assembly industries all use GEM technology. The adaptability of the GEM standard allows it to be applied in almost any manufacturing industry.

Virtually every semiconductor manufacturing company around the world currently uses the GEM standard on all manufacturing equipment, and has done so for many years. This includes 300mm, 200mm and 150mm wafer production. GEM was successful enough early on that is was used as a foundation for a number of complementary factory automation standards. These additional standards are referred to as the GEM 300 standards, so named because of their widespread adoption by the factories dedicated to the manufacturing integrated circuits on 300mm wafers.

In 2008 the photovoltaic (solar cell) industry officially adopted GEM technology, denoting it as the SEMI PV2 standard (Guide for PV Equipment Communication Interfaces). This standard directly references and requires an implementation of the GEM standard. In 2013 high-brightness LED industry created a similar SEMI standard HB4 (Specification of Communication Interfaces for High-Brightness LED Manufacturing Equipment). Recently, the printed circuit board industry has followed suit, approving the SEMI PCBECI standard (Printed Circuit Board Equipment Communication Interfaces). All three standards similarly define implementations of the SEMI standard that increase GEM's plug-and-play characteristics and mandate only a subset of GEM functionality to facilitate GEM development on both the equipment and factory hosts.

Several additional SEMI standards have been created over the years to enhance GEM implementations and are applicable to any industry and equipment. For example, SEMI E116, Specification for Equipment Performance Tracking, defines a method for measuring equipment utilization, both at the overall equipment level and for major components within the equipment. SEMI E157, Specification for Module Process Tracking, allows an equipment to report the progress of recipe steps while processing. SEMI E172, Specification for SECS Equipment Data Dictionary, defines an XML schema for documenting the features implementing a GEM interface. Finally, SEMI E173, Specification for XML SECS-II Message Notation, defines an XML schema for logging and documenting messages.

2.6 Publish/Subscribe Design Pattern with a Message Broker

The GEM interface acts as a message broker between the host and equipment by using a publish/subscribe pattern with dynamic subscriptions so that the host only receives notification for desired messages. This applies to GEM data collection features, namely trace data collection, collection event reports and alarms. Even though there is typically only one publisher (the equipment) and one subscriber (the host), this design pattern is very powerful. It allows the equipment development engineers to design and create a single GEM interface that publishes everything required for all potential customers. It also allows the customer to set up their subscriptions dynamically based on changing needs and current issues. The equipment supplier can even upgrade a GEM interface with additional features without affecting the existing host system.

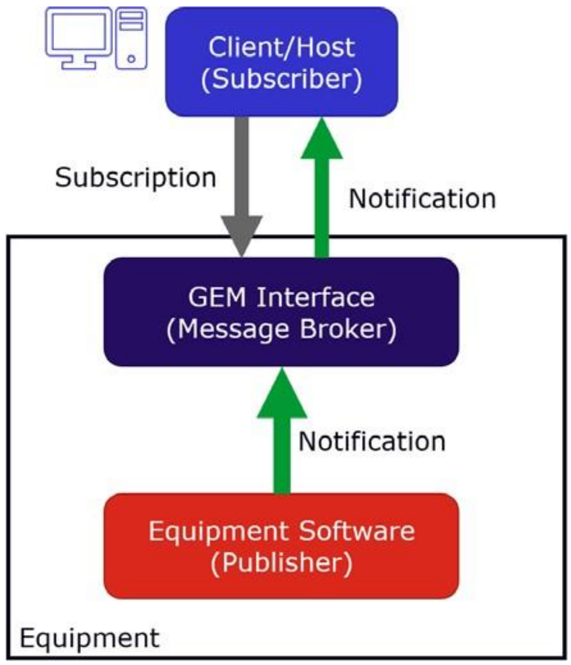


Figure 4 Message Broker

One issue with a publish/subscribe design pattern in an industrial application is that the publisher typically must send everything to the message broker all the time, regardless of the subscriptions. If the message broker is located on an external entity, then you need enough protocol bandwidth between the publisher and message broker to handle all available subscriptions. For a simple device or equipment, this might not be a problem because there are fewer messages and they are typically small. However, for medium-to-complex equipment which may include a tremendous amount of available data, this can become a huge resource problem. Since a GEM interface allows the host to select not only the messages to be used but also the data content of those messages, the throughput requirements for a GEM interface are difficult to predict. Fortunately, a GEM interface is typically integrated into the equipment software design and runs on one of the equipment's computers. This allows the equipment software to set up the publisher/message broker communication on an internal network that is isolated from the factory's network.

2.7 GEM Scalability

GEM requirements are divided into two groups: Fundamental Requirements and Additional Capabilities. Any equipment that implements GEM is expected to support all the Fundamental Requirements. Additional Capabilities are optional and therefore are only implemented when applicable. This makes the GEM standard inherently flexible so that both simple devices and complex equipment can implement GEM. GEM naturally scales with the complexity of any system. A simple device need only implement the minimum functionality to serve its purpose, whereas a complex equipment can implement a fully featured GEM interface to allow the factory to fully monitor and control its functionality.

Note again that the requirements in the GEM standard only apply to the equipment and not the host. This means that equipment behavior is predictable, but the host can be creative and selective, choosing to use whatever features from the equipment's GEM interface are most appropriate to attain it goals.

2.8 Data Classifications

A GEM interface is capable of publishing three classifications of data: status variables, data variables and equipment constants. A status variable is status information about the equipment hardware or software—data that can be queried at any time. For example, sensor data, other I/O, and the state of the hardware are published as status variables. A data variable is information that pertains to a specific collection event. Therefore, a data variable can only be collected as part of a collection event report. For example, if there is a collection event "barcodeScanned," there might also be associated data variables such as "barcode" and "materialLocation" to convey the scanned barcode and the location where it was scanned. Finally, an equipment constant is a setting to configure the equipment's behavior. Values of equipment constants can be changed by the operator and by the host through the GEM interface.

2.9 Data Types

Each status variable, data variable and equipment constant is declared as a specific data type. The following table describes the available data types.

Boolean	true or false
Binary	A value 0-255, often used as enumerations and in arrays for raw data and images
Unsigned Integer	1, 2, 4, or 8-byte unsigned integer
Signed Integer	1, 2, 4, or 8-byte signed integer
Floating Point	4 or 8-byte floating point
Array	Array of Boolean, Binary, Unsigned Integer, Signed Integer and Floating Point is allowed
ASCII	ASCII string The standard allows supports 2-byte character strings and JIS-8, which are rarely used
List	Structured list of any type, including another List

3 Fundamental Requirements

The Fundamental Requirements should always be supported on any device or equipment implementing the GEM standard.

3.1 State Models

The Communication State Model defines how to establish communication between the host and equipment. This allows the host or equipment to initiate communication and handles simultaneous attempts from both parties. Communication is established with a successful host or equipment initiated S1F13/S1F14 message exchange.

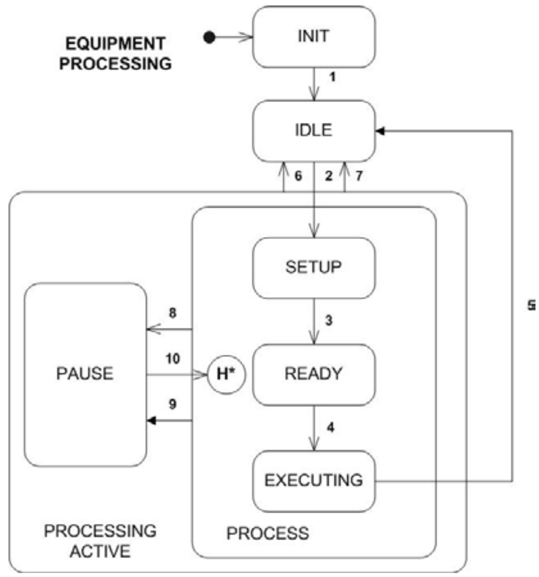
The Control State Model manages whether the equipment is online or offline, whether the host is online or offline, and most importantly, whether the host or local operator is in control of the equipment. At the equipment, the operator/technician can configure the GEM interface control state. For safety considerations, the local operator can always take over control of the equipment and stop the equipment. The states and transitions are reported using the GEM status variable and collection event features.

By requiring consistent implementations of the communication and control state models, the GEM standard makes it easier to manage communication and manage the factory's level of equipment control. Every equipment shows its communication and control status and allows the state to be changed in the equipment's Human Machine Interface (HMI), typically a Graphical User Interface (GUI).

3.2 Processing State Model

Every equipment must implement and report a processing state model, but the model can and should be customized for each equipment type. The main intention is to report basic equipment utilization, such as when the equipment is initializing, idle and processing. The state and transitions are reported using the GEM status variable and collection event features.

Figure 5 GEM Example Processing State Model



The processing state model ensures basic reporting to calculate production throughput and equipment utilization statistics. It also provides state information for coordinating remote control commands and other controlling features.

3.3 Collection Event Notification

A GEM collection event is a notification from the equipment to the host. The notification reports when something of possible interest to the host occurs. An equipment might support only a few collection events or many thousands of collection events. GEM requires certain standard collection events, but most are unique to an equipment's GEM interface. A collection event subscribed to by the host is deemed as "enabled." A collection event without a host subscription is deemed "disabled." Subscriptions are persisted so that the same collection event notification occurs after restarting the equipment.

Collection event notifications are always sent using the S6F11 collection event report message. A host does not subscribe to the S6F11 message itself, but subscribes instead to specific collection event notifications using message S2F37. The host can also enable or disable all collection event notification with a simple S2F37 message. The collection event report message always includes the collection ID number to identify the collection event. It can also include one or more reports with additional data about the collection event and equipment status information. The equipment supplier can pre-define reports and link them to collection events, but this is not necessary when using the optional feature Dynamic Event Report Configuration (described later as an Additional Capability).

Collection events are a key feature that many other GEM features utilize. They allow the factory to track what the equipment is doing. Collection events also serve as triggers to the host to initiate host actions and other activities such as material delivery, consumable replenishment and equipment maintenance. Because collection event notification is subscription based, an equipment supplier can publish the same collection event to all customers even though each customer is interested in a unique set of collection events.

An end user must consult the equipment's GEM interface documentation to understand a collection event's full meaning. A common strategy is to create an "equipment characterization" process whereby the host subscribes to all collection event notifications, runs the equipment normally, logs the collection event occurrences and correlates this information to equipment operation.

3.4 Identification

Each equipment is required to publish its model number (MDLN) and software revision (SOFTREV) as a form of identification. Both are strings up to 20 characters long. Both are used in the S1F13/S1F14 messages to establish communication, in the heartbeat message S1F1/S1F2, and are available as status variables for general data collection.

Immediately upon establishing communication with a GEM interface, the end user knows the model and software version of the equipment. Additionally, the end user knows if the software has been upgraded since the last communication and is therefore aware of potential behavior changes and new features.

3.5 Error Messages

GEM defines seven standard error messages that equipment are required to support. If the host sends an unsupported message, a message with a bad format, or forgets to send an expected message, the equipment will report this to the host with a stream 9 error message. This is in addition to the unique error codes returned by each message. A host does not send stream 9 error messages.

3.6 Documentation

An equipment supplier must provide GEM documentation with a GEM interface implementation. Typically, this is provided in PDF format or similar electronic format. The GEM documentation must include a list of supported messages, the format of each message and the list of supported remote commands, status variables, collection events, alarms, equipment constants and data variables. The documentation can also include message scenarios to explain normal and abnormal operation. An equipment's GEM documentation allows the end user to use the GEM interface for whatever purpose without additional information.

In the GEM documentation an equipment must declare which GEM features are and are not implemented as well as what is and is not fully compliant to the GEM standard. The declaration looks like this:

GEM Compliance Statement		
FUNDAMENTAL GEM REQUIREMENTS	**IMPLEMENTED**	**GEM COMPLIANT**
State Models	☐Yes ☐No	☐Yes ☐No
Equipment Processing States	☐Yes ☐No	
Host-Initiated S1 = F13/F14 Scenario	☐Yes ☐No	

Event Notification	☐Yes ☐No	
On-Line Identification	☐Yes ☐No	
Error Messages	☐Yes ☐No	
Documentation	☐Yes ☐No	
Control (Operator Initiated)	☐Yes ☐No	
ADDITIONAL CAPABILITIES	**IMPLEMENTED**	**GEM COMPLIANT**
Establish Communications	☐Yes ☐No	☐Yes ☐No
Dynamic Event-Report Configuration	☐Yes ☐No	☐Yes ☐No
Data and Collection Event Namelist Requests	☐Yes ☐No	☐Yes ☐No
Variable Data Collection	☐Yes ☐No	☐Yes ☐No
Trace Data Collection	☐Yes ☐No	☐Yes ☐No
Status Data Collection	☐Yes ☐No	☐Yes ☐No
Alarm Management	☐Yes ☐No	☐Yes ☐No
Remote Control	☐Yes ☐No	☐Yes ☐No
Equipment Constants	☐Yes ☐No	☐Yes ☐No
Process Program Management	☐Yes ☐No	☐Yes ☐No
Material Movement	☐Yes ☐No	☐Yes ☐No
Equipment Terminal Services	☐Yes ☐No	☐Yes ☐No
Clock	☐Yes ☐No	☐Yes ☐No
Limits Monitoring	☐Yes ☐No	☐Yes ☐No
Spooling	☐Yes ☐No	☐Yes ☐No
Control (Host-Initiated)	☐Yes ☐No	☐Yes ☐No

Non-compliant behavior is required to be documented so that the factory is alerted appropriately.

SEMI standard E172 (Specification for SECS Equipment Data Dictionary (SEDD)) defines an XML schema for GEM interface documentation so that an equipment supplier can provide an XML file to fully describe its GEM interface. The XML file is called an SEDD file and describes the list of support GEM messages, remote commands, collection events, data variables, status variables, equipment constants, alarms, and supported standards. This documentation method is superior to the traditional GEM documentation because it is consumable by an intelligent software application.

4 Additional Capabilities

Although the additional capabilities are optional, most equipment should implement many of these features.

4.1 Dynamic Collection Event Reports

Dynamic Collection Event Reports augment the fundamental requirement Collection Event Notification described above to provide extremely powerful, flexible and efficient data collection. The same S6F11 collection event report message that provides collection event notification also allows zero to many reports, where each report can have one to many data variables, status variables and equipment constants. Moreover, each variable value can be any data type: simple, scalar, array or even a structure of information. When an equipment implements a collection event, it can also associate zero to many data variables for that collection event. The data variables associated with a collection event are not automatically transmitted to the host with the collection event report. The host is expected to define reports (message S2F33) with the data it wants, possibly including associated data variables, any status variables and any equipment constants. The host then links these reports to collection events (message S2F35) and enables the collection events (message S2F37). When a report includes one or more data variables, the host should only link a report to collection events which are documented to be associated with those data variables. All dynamic event report configuration is persistent so that the same data collection continues after restarting the equipment.

The publish/subscribe pattern implemented by dynamic event report data collection is far superior to a traditional publish/subscribe pattern. In a typical publish/subscribe design the subscriber simply chooses what messages it wants to receive, and the content and structure of the messages are fixed. But in GEM, not only does the host subscribe to receive specific collection events, it also subscribes to the specific data to be received with each of those collection events. This two-layered subscription in the dynamic collection event report feature is a key design pattern in the GEM standard that makes it particularly popular and useful for a tremendous variety of factory applications. The host can not only track what the equipment is doing, but it can receive the data and status information it wants along with the collection event report notifications. This feature makes the GEM standard's data collection extremely flexible and efficient.

4.2 Variable, Trace, and Status Data Collection

Variable Data Collection, Trace Data Collection and Status Data Collection allow a host to collect status information (called status variables) from the equipment in various ways. The equipment makes the status variable data available for various forms of data collection, but it is up to the host to request the desired data. There are several ways to accomplish this. A host can define a report with a set of status variable IDs and ask for their current values at any time using the report ID with message S6F19. Similarly, a host can directly ask for the current values of any set of status variables using message S1F3. Most important and useful, a host can subscribe to trace data collection using message S2F23. A trace subscription requests the GEM interface to periodically send a trace report with the current values for the requested status variables using an S6F1 message. In the subscription definition, the host chooses the desired set of status variables, the update rate (for example 1 second or 100 milliseconds), and the reporting group size. The GEM standard allows the frequency to be as fast as 100 Hz, but the equipment can choose what frequency it chooses to support. A group size of 2 or more caches the data in the GEM interface and then sends a group of trace data reports in a single S6F1 message. Trace subscriptions are also persistent so that they continue to be reported even after restarting the equipment.

Trace data collection allows the host to receive a constant stream of status information from the equipment. This can include sensor input, actuator commanded and current values, process parameters, state information and any other information published as status variables.

Trace data collection offers an implementation far better than the traditional publish/subscribe pattern. In a typical publish/subscribe the subscriber is simply choosing what messages it wants to receive where the content of the messages and the rate of publication are fixed and defined by the publisher. But with GEM trace data collection, the subscriber decides what data to receive and the publication frequency. This makes GEM extremely useful and efficient.

4.3 Self-Description

Through the GEM interface, a host can ask the equipment what information is available for data collection. For example, message S1F11 retrieves the list of available status variables. Message S1F21 retrieves the list of available data variables. Message S1F23 retrieves the list of available collection events and the associated data variables for each collection event. Message S2F29 retrieves the list of available equipment constants. Finally, message S5F5 retrieves the list of available collection events. Together, these messages make GEM data collection plug and play. An intelligent host software package can dynamically configure itself to facilitate the data collection setup.

Upon connecting, the host can find out what is available to help engineers choose the right data to collect. Some aspects of equipment characterization can be automated.

4.4 Alarms

GEMs alarms report to the host when something happens in the equipment that could be dangerous to the equipment, material and/or operator. Each alarm has two states: "set" and "clear." Each alarm has two associated collection events: one for each state transition. These associated collection events allow the host to collect any available data both when an alarm occurs and when it is resolved. No specific alarms are required by GEM. The equipment must document the list of all alarms available in the GEM interface. Through the GEM interface the host can ask what alarms are available.

Alarms use a publish/subscribe pattern. When a host wants to be notified each time an alarm changes state, it subscribes to that alarm using message S5F3. It is also very simple to enable or disable all alarm notification. Any time the alarm becomes set or clear, the equipment notifies the host using message S5F1. The subscription is persisted so that the same alarm notification occurs after restarting the equipment. Additionally, the host can query the state of all alarms at any time. This is useful after a host first connects to the GEM interface and needs to know the alarm states.

The Alarm feature allows the factory to implement a variety of fault detection applications. A host can disable notification for insignificant alarms and thereby ignore them. This is especially useful when the equipment supplier's alarm classification differs from that of the factory.

4.5 Remote Commands

A GEM remote command is identified by its command name. A host can send commands to the equipment using message S2F41 or S2F49. Optionally, a remote command can also have one-to-many name/value pairs. In the name/value pair each name is a string, but the value can be of any data type, including a structured list. The name/value pair allows the host to give data to the equipment related to the remote command.

When using message S2F49, a remote command can also include an object name. The object name is a string to identify the target for the command such as "chamber3" or "conveyor2."

The GEM standard defines the expected behavior for the following commands: START, STOP, PAUSE, RESUME, ABORT and PPSELECT. PPSELECT is used for selecting a recipe where the recipe is passed using a name/value pair. However, the equipment can define any number of additional remote commands. An equipment's GEM documentation must identify the list of supported remote commands and the available name/value pairs associated with each command.

Remote commands allow an equipment to be controlled remotely by numerous means. Remote commands can pass data to the equipment such as a lot number, material tracking ID, material list or even a substrate map. The uses for remote commands are limitless.

4.6 Equipment Constants

An equipment constant is an equipment configuration setting. GEM allows equipment constant values to be set by the host using message S2F15. If a setting is changed at the equipment by an operator or engineer, the host is notified with a collection event and data to identify the equipment constant changed and its new value. All equipment constant values are persisted so that the equipment behavior remains the same after restarting the equipment. The host can also use message S2F13 to request the current equipment constant values.

Equipment constants allow the factory to configure equipment behavior remotely. When there are multiple equipment of the same type, the ability to check and set equipment constants remotely ensures consistent behavior across the equipment. Equipment suppliers can support unique factory requirements and use the GEM interface to configure the equipment behavior correctly for each customer, such as enabling or disabling an operator assist during processing or disabling a barcode scanner that needs servicing.

4.7 Recipe (Process Program) Management

A "recipe," also called a "process program," is a set of instructions that the equipment executes to process material or otherwise provide its intended function. A recipe should be transferrable to any equipment of the same make and model to produce the same results. The instructions may be in an ASCII, binary or structured format as designated by the equipment supplier. Recipe management allows the host to get the list of available recipes, upload recipes from the equipment, download recipes to the equipment, delete recipes, and subscribe to be notified when a recipe is changed, created or deleted at the equipment. The equipment operator can download recipes from the host and upload recipes to the host.

GEM recipe management allows the factory to ensure that the correct recipe is run with the correct material. effectively minimizing scrap and human error. It allows the factory to protect the golden recipes by storing them on a backed-up host computer in case of system failure and to distribute recipes to other equipment of the same type. Notably, the cost savings across the industry attributed to this feature alone was one of the primary initial drivers and key success factors for GEM adoption.

4.8 Material Movement

Material Movement requires the equipment to use collection events when material arrives at the equipment and departs from the equipment. When applicable, the equipment should also provide data variables to identify the port, material IDs, and any other relevant data.

Material Movement allows the factory to track when material is delivered to the equipment and removed from the equipment, whether this is done by an operator or by an automated material handling system. This is useful for basic production logistics to log the exact material history. The factory can avoid accidental misprocessing by double-checking factory floor operation and making sure that material is delivered to the right equipment at the right time.

4.9 Terminal Services

Terminal services allow the host to send a text message to the equipment operator and for the operator to acknowledge receipt of the message. It also allows the operator to send a message to the host system. When a message arrives for the operator the equipment alerts the operator.

Terminal services can be useful in interesting situations. For example, if the host software detects an issue with the equipment, such as loading the wrong consumables or product material, the host can report specific details about a detected error to the operator so he/she can identify the exact nature of the problem and resolve the issue.

4.10 Clock

The GEM clock feature allows the host computer's clock to synchronize with the equipment computer's clock. It also requires that a clock status variable be available for all data collection. The clock feature allows the factory to correlate the GEM data collection from the equipment with the data from other equipment as well as with data collected from other sources by the host system.

Synchronized clocks allow the host to implement meaningful data analysis by comparing data from disparate sources, such as other equipment, devices and systems. For example, a factory might use external dedicated systems to measure and report gas flow rates into some equipment. Synchronized clocks allow for data from these systems to be merged and stored in a single database without timestamp correction. This is absolutely essential for advanced troubleshooting and data analysis.

4.11 Spooling

The Spooling feature allows the equipment to save messages that would otherwise be lost when communication with the host is interrupted. When the host reestablishes communication, the host must decide to request or purge the spooled messages. Spooling is primarily for preserving trace data collection, alarm and event report data collection messages.

The Spooling feature allows factories to set up equipment data collection so that no data is lost. This is vital for some applications, such as when the factory has regulatory requirements for preserving data, when the equipment data is required for later processing steps, or when the factory must take action based on the received data.

5 Frequently Asked Questions

Where can I get a copy of the GEM standard?	Official copies must be obtained through SEMI. Standard documents can be ordered or downloaded for a fee at the SEMI website: http://www.semi.org/
How does a system become GEM certified?	There is no official GEM certification. GEM compliance is self-proclaimed. Software programs are available for testing equipment to validate its GEM compliance, EquipmentTest. Note that GEM compliance does not require all GEM features to be implemented. For example, some equipment may not implement Remote Commands and Process Program Management, yet they can still be GEM compliant if they correctly implement all the GEM Fundamental Capabilities.

Should the host or equipment implement message logging?	When using a GEM interface, the messages sent between the host and equipment define the behavior of the GEM interface and the equipment. Anytime there is a problem, it is invaluable to have a message log to determine whether the equipment has a defect in its GEM interface or not. The best notation to use for GEM message logging is described in SEMI standard E173, Specification for XML SECS-II Message Notation (SMN). SMN defines an XML schema for logging messages at the SECS-II layer as well as at the protocol layer. SMN can also be used for documentation. Prior to SMN, numerous variations of SECS Message Language (SML) were used. But SML was never standardized, well documented or consistently implemented. Additionally, a company claims to own a copyright on SML. For these reasons, GEM users are strongly encouraged to use SMN, which is far superior in every way.
Can more than one host simultaneously establish communication with a single piece of equipment?	Yes, but not all GEM interface software products support this capability. Cimetrix CIMConnect software product has a built-in multiple-host feature that simplifies the process of communicating with more than one SECS/GEM host at a time using HSMS-SS or SECS-I communication. When using HSMS-SS, each host uses a unique port and all data collection subscriptions are unique that that host.
How long does it take to implement a SECS/GEM interface on an equipment?	This depends on the complexity of the GEM interface and the equipment. A simple GEM interface may only take a few weeks. Some freeware libraries exist to support this development. It is usually more cost-effective and lower risk to purchase a commercial software product. There are several commercial GEM software products available worldwide such as the Cimetrix CIMConnect product, which many consider to be the best product on the market. With CIMConnect you can implement a complex GEM interface in a few weeks and a simple one in a few days.
Is GEM difficult to implement?	No more difficult that the alternatives. With modern software development languages and supporting IDEs, it is easy to implement the GEM standard.
How fast is a SECS/GEM interface?	Trace data collection is limited by the GEM standard to a minimum data reporting interval of 1 centisecond, allowing for up to 100 Hz data collection by this means. Many equipment limit the data collection to a lower frequency, such as 1 Hz for older implementations and 10 Hz for more modern implementations. Other forms of data collection, such as collection event reports and data polling using the S1F3, are not limited by the standard but instead are limited only by the networking and computer resources of the equipment and host systems. Because the combined SECS-II and HSMS message format is very efficient, a lot of data can be transferred using little network bandwidth. The precise data rates

	depend on many factors such as the network, the GEM software in both the host and equipment systems, and the computer hardware. Older versions of the GEM standard were limited to 1 Hz trace data collection.
Can you implement custom messages?	Yes, but this is highly discouraged. In most cases, existing GEM messages can accommodate the needs of the equipment and factory end users. The use of custom messaging naturally requires custom development both on the equipment and by the factory, increasing cost, integration time, and risk.
Do factories ever require the equipment to violate the GEM standard?	A small number of factories exist with old or poorly written host software that demand the equipment to violate the GEM standard in one way or another. Sometimes factories will use their position of power as "the customer" to demand compliance to their requirements instead of the industry standard. This is extremely bad practice and does the entire industry a disservice. A few small changes in the host software can usually remedy their issues.
What advantages does GEM have over other interfaces/protocols?	There are several advantages. 1. Message Broker Location With a GEM interface, the message broker is located on the equipment rather than on the factory network. Most other competing interfaces/protocols position the message broker on the factory network. This means that the equipment or device must publish all available data to the message broker which can require extensive network bandwidth. With simple devices and equipment this might be tolerable, but it cannot be scaled to complex equipment or to devices and equipment that publish extensive amounts of data. With a GEM interface, all message subscription management is handled on the equipment itself; therefore, only requested data is published on the factory network. 2. Flexible Data Subscriptions Many other protocols/interfaces define a different message for each report and hardcode the expected data and message structure. In such cases, the host/client is only subscribing to the message but cannot choose what data is in the message. As a result, you might not get the data you really need, while getting data you don't really want and wasting resources in the process. Moreover, it is difficult to ask for additional data. In GEM, the host subscription includes the list of requested data (for both collection event reports and trace data collection) and chooses the data frequency (trace data collection). With a GEM interface, the messages structures are flexible to adapt to requests. A host can use the same S1F3 message or trace data collection to request 1 or thousands of status variables

	from the equipment. A host can set up collection event reports to receive any number of data reports, and where each report can include one to many data variables, status variables and equipment constants. Reusing the same messages for various purposes simplifies message deserialization. 3. Flexible Subscription Frequency Many other protocols/interfaces fix the data report frequency. The client is only subscribing to the message but with little control and at the mercy of the publisher. The data might be published too slowly for the intended purpose. Or it might be published too frequently, thereby wasting resources. GEM allows the subscriber to dictate the trace report frequency. 4. Industry Proof and Momentum The GEM standard is a low-risk, high performance solution already proven to work in hundreds of highly sophisticated factories around the world collection event reports and trace data collection) and chooses the data frequency (trace data collection). With a GEM interface, the messages structures are flexible to adapt to requests. A host can use the same S1F3 message or trace data collection to request 1 or thousands of status variables from the equipment. A host can set up collection event reports to receive any number of data reports, and where each report can include one to many data variables, status variables and equipment constants. Reusing the same messages for various purposes simplifies message deserialization. 5. Flexible Subscription Frequency Many other protocols/interfaces fix the data report frequency. The client is only subscribing to the message but with little control and at the mercy of the publisher. The data might be published too slowly for the intended purpose. Or it might be published too frequently, thereby wasting resources. GEM allows the subscriber to dictate the trace report frequency. 6. Industry Proof and Momentum The GEM standard is a low-risk, high performance solution already proven to work in hundreds of highly sophisticated factories around the world..

What are the most important features in a GEM product?	There are many important features, but here are some of the key ones: 1. Technical Support The GEM interface is often a mission-critical capability for production. There are many details in the SEMI standards. Before selecting a product, make sure that the product is backed by a solid company with a responsive, experienced technical support team. 2. Performance Some products use much less CPU than others for the same set of tasks. A product that uses less CPU can achieve higher data collection rates. As factories increasingly optimize their manufacturing processes they rely on more and more data collection from the equipment. Select a product that can use computing and networking resources most efficiently and can meet both today's and tomorrow's throughput requirements. 3. Supporting Multiple Hosts In recent years, the importance of supporting multiple client applications has increased. For example, PV manufacturers documented the need for an "IT interface of the equipment that allows an arbitrary number of clients to connect to the equipment to gather data from the equipment (all kinds of data collection) and to interact with the equipment (remote control, etc.)." Choose a product that has multiple client access as a built-in feature such as CIMConnect. 4. Client-Server Architecture A GEM interface interacts with all the components within the equipment. Purchase a product with a client-server architecture so that all the components can interact directly with the GEM software. This foundation will reduce the time, cost, and complexity of software development.

6 GEM Terminology

The table below deceibes common terms related to the GEM standard.

Term	Description
Alarm	"An alarm is related to any abnormal situation on the equipment that may endanger people, equipment, or material being processed" [SEMI E30, 2]. GEM allows the host to be notified when alarm conditions are both detected and cleared.
Collection Event	A collection event is a "detectable occurrence significant to the equipment" that "is considered to be significant to the host." [SEMI E30, 2] GEM allows the host to be notified when a collection event occurs. This allows the host to track the equipment's activity.
Data Variable	Data variables "…may only be valid upon the occurrence of a particular event." [SEMI E5, 6.6]. The host can gather data variable values from the GEM equipment. The data variable values provide information specifically related to the event.

Equipment Constant	Equipment Constants are "settable by the Host." [SEMI E5 6.6] The host can gather equipment constant values from the GEM equipment. The host can also set equipment constant values on the GEM equipment to control the equipment's behavior.
GEM Equipment	An "intelligent system which communicates with a host" [SEMI E4, 2.1] and complies with the GEM standard.
Host	"An intelligent system which communicates with the equipment." [SEMI E4, 2.1]. GEM does not intend to define how the host should behave. The GEM standard defines the set of messages a host must use when interacting with GEM equipment. A host can communicate with multiple GEM equipment. Sometimes a host may serve as a bridge between the Manufacturing Execution System (MES). Sometimes a host is dedicated to specific data collection applications, such troubleshooting, predictive maintenance, advanced process control (APC), or fault detection and classification (FDC). Sometimes a host completely controls equipment processing and material flow remotely for full automation. Sometimes a host does all these things and more.
HSMS-SS	SEMI standard High-Speed Message Service Single Selected-Session, which defines TCP/IP network communication used by GEM for host/equipment communication. It has effectively replaced the SECS-I standard. Only one host client can use a specific port at a time.
PCBECI	Specification for Printed Circuit Board Equipment Communication Interfaces, an approved SEMI standard for PCB manufacturing equipment suppliers and their customers.
PV2 (PVECI)	Guide for PV Equipment Communication Interfaces, an approved SEMI standard specifically for photovoltaic equipment suppliers.
Process Program	(see Recipe)
ecipe	A set of instructions for the equipment that serve some specific purpose (wafer processing, defect inspection, calibration, equipment test, etc.).

Report	"A set of variables predefined by the equipment or defined by the host…." The host uses reports to gather status variable, data variable, and equipment constant values. The host can request a report explicitly or attach a set of reports to a collection event.
Status Variable	"Status variables may include any parameters that can be sampled over time such as temperature or quantity of a consumable." [SEMI E5, 6.5] "Status values … always contain valid information." [SEMI E5, 6.6] The host can gather status variable values from the GEM equipment.
SECS-I	SEMI Equipment Communications Standard 1 Message Transfer - defines RS-232 serial communication used by GEM for host/equipment communication. It has been phased out due to inherent speed limitations and replaced by the HSMS standard.
SECS-II	SEMI Equipment Communications Standard 2 Message Content. GEM is a specific implementation of the SECS-II standard. SECS-II defines most concepts and functionality used in the GEM standard. Many SECS-II capable systems are not GEM compliant.
SECS-II Message	All GEM equipment and host communication is accomplished using SECS-II messages. Each unique SECS-II message is identified by its stream number (S) and function number (F). The SECS-II standard defines a large set of SECS-II messages, specifying each message's purpose, content, and usage. The GEM standard defines how to use a subset of these SECS-II messages, while allowing other SECS-II messages to be used in addition to this subset.