Virtual Factory Equipment Interface: Version 2.2

SEMATECH
Technology Transfer 95113016A-TR

Virtual Factory Equipment Interface: Version 2.2

Technology Transfer # **95113016A-TR**
SEMATECH
November 30, 1995

Abstract: This document describes version 2.2 of the Virtual Factory Equipment Interface (VFEI) for semiconductor manufacturing systems. This release contains updates and corrections to the prior version. Sections 1.3 and 1.4 of this document summarize the changes from version 2.1 to version 2.2. The document includes descriptions of SEMATECH Baseline Requirements, VFEI concepts, VFEI commands, VFEI predefined events, VFEI predefined data variables, and VFEI predefined error codes. Syntax is provided for implementing VFEI commands when used with ASCII messaging systems.

Keywords: Strategic Cell Controller, Virtual Factory Equipment Interface, Control Systems

Approvals: Frank Yeager, Author
Pete Whelan, Project Manager
Dan McGowan, Technical Information Transfer Team Leader

Table of Contents

List of Figures

List of Tables

Acknowledgements

The author wishes to thank the members of the VFEI Technical Advisory Forum for their contributions to this release of VFEI:

Jeff Acklen	SEMATECH	Mike McDonald	SEMATECH
Charles Baylis	FASTech	Kevin Nguyen	Mitta Technology
Paola Benassi	Mitta Technology	CarolPergola-Koch	AT&T
Ashish Chona	Radiance Systems	Margaret Pratt	SEMATECH
Peter Dube	National Semiconductor	Bing Pu	Rockwell
Yolanda Habraken	BDM Technologies	Glen Stefanski	IBM
David Jacobs	SEMATECH	Greg Thomson	FASTech
Jim King	Digital	Bob Umstead	BDM Technologies
Fred Mast	Hewlett-Packard	Pete Whelan	SEMATECH
Richard Mangieri	AT&T	Joe White	SEMATECH

Also, special thanks are due to the following persons for their contributions to this document:

Dave Francis	Auto-Soft
Toni Guckert	FASTech
Simon Hu	Motorola
Larry Kurfis	IBM
Jon Milan	Motorola
Doug Raymond	Praxis Technologies
Greg Vorhis	AT&T
Frank Weger	EDS/Philips Semiconductors

1 EXECUTIVE SUMMARY

1.1 Overview

This document describes version 2.2 the Virtual Factory Equipment Interface (VFEI) for semiconductor manufacturing systems. VFEI defines a messaging interface between cell controller applications and “machine drivers.” Machine drivers, like device drivers on PC's, provide an interface to semiconductor equipment. The requirements for VFEI messages are derived from the SEMATECH Baseline Requirements process.

1.2 Background

The VFEI Technical Advisory Forum first convened on May 27, 1993, with the charter to enhance and extend the VFEI specification. Work groups were formed to address specific areas of the VFEI specification. The first work group was concerned with the specification of the VFEI syntax for ASCII busses. A feature work group addressed semantic changes and additions to VFEI. A policy group was chartered with defining policy for development process and standardization. Release 2.1 represented the final recommendation of these work groups.

The VFEI Technical Advisory Forum reconvened on May 9, 1995, more than a year after the release of the 2.1 specification. The charter of this group was to enhance and extend the VFEI specification based on experience gained using the current 2.1 release.

1.3 Policy Changes from VFEI 2.1

Policy changes between versions are as follows:

- Baseline Requirements, which are referenced throughout this document, are no longer confidential. They are available from SEMATECH as *Strategic Cell Controller (SCC) Baseline Requirements and Design Scenarios 2.0,* Technology Transfer #92071186A-ENG.
- Obsolescence policy. If a feature from the previous revision has been removed, it is indicated by an ***OBSOLETE*** notation. Features marked OBSOLETE are no longer functional. It is the VFEI Advisory Forum’s policy to mark the obsolete item[s] in the specification for one release. This will allow for industry review and comment. The Forum strives to maintain backwards compatibility with previous VFEI releases. Features are made obsolete only when unanimous experience of the Forum members suggest such changes are warranted.

1.4 Changes from VFEI 2.1

Content changes between versions include the following:

- The order of appendixes in the back of the manual was changed.
- Sections 4 (VFEI Events), 5 (VFEI Variables), and 6 (VFEI Error Codes) were moved in front of Section 3 (VFEI Commands).
- A new section has been added on using the SCC Baselines.
- Two new TYPE values (CASSETTE_MAP and WAFER_MAP) have been added to the RESOURCE commands.
- The INITIALIZE command has been changed; primarily, the NOEVENT flag has been marked for obsolescence and Baseline events are no longer automatically enabled.

- New error code designations have been added. Error explanation fields that were previously blank now have explanations.
- The Backus Naur Form (BNF) in the ASCII and FILE syntax sections has been enhanced to better illustrate list and array information.
- VFEI Predefined Variables has been broken into three new sections.
- VFEI Predefined Events has been broken into two new sections.
- Use of MACHINE COMMAND has been clarified; especially when dealing with already defined VFEI commands (i.e., ABORT, PAUSE, START).
- Section 4.4 has been expanded to clarify the different data types.
- DISPLAY MESSAGE command has now been made bi-directional.
- EVENT SETUP now allows multiple EVENT_IDs, and the tie-in with INITIALIZE's NOEVENT flag has been removed.
- EVENT_SETUP can no longer selectively disable reporting of any variables previously enabled.
- The MOVE command was created for material handling, along with new events and variables.
- The "undefined optional name" data item was changed to "user defined name".
- Data item ALARM_CATEGORY was added to the ALARM REPORT command.
- The use of user defined variables in STATUS LIST and TRACE REPORT was clarified.
- The standard data items no longer have to be provided in a pre-described order.

Feedback on this information should be sent to the following:

VFEI Advisory e-mail account
vfei@sematech.org

E-mail will be distributed to all current members of the VFEI Advisory team.

Requests for additional copies of the VFEI Specification should be sent to:

SEMATECH
Technology Transfer
2706 Montopolis Drive
Austin, TX 78741-6499
(512) 356-SEMA

2 INTRODUCTION

Cell controllers integrate factory shop floor control and business systems with semiconductor equipment. To achieve this integration, cell controllers include

- A shop floor interface to provide communication between shop floor control systems and cell controllers
- An optional user interface to provide communication between users and cell controllers
- An equipment interface to provide communication between equipment and cell controllers
- Applications to furnish cell-level functionality and control

The following diagram provides a view of how a cell controller and VFEI might fit into a semiconductor factory.

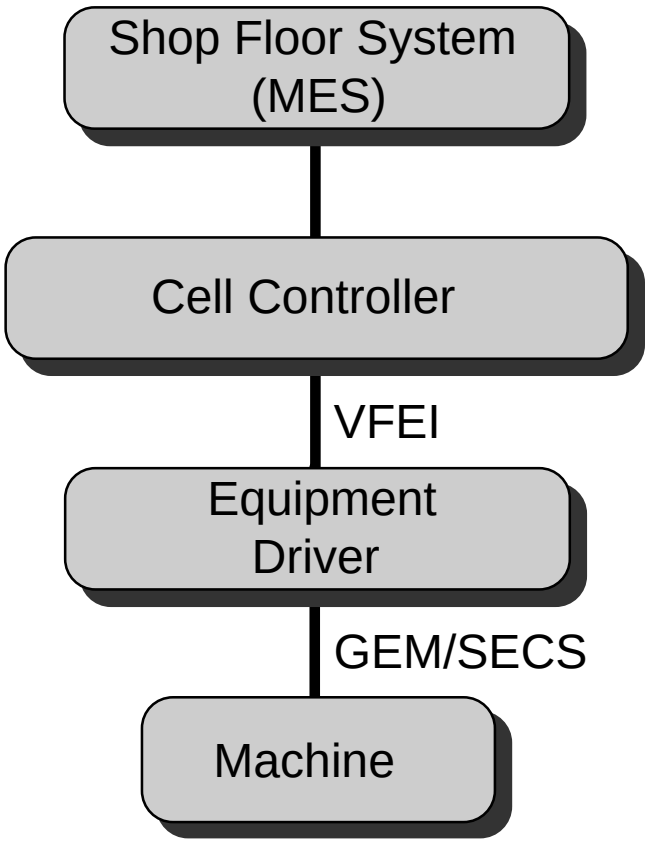


Figure 1 Sample Cell Controller Configuration

Historically, communication between semiconductor equipment and cell controllers has been equipment-specific, causing equipment control logic in cell applications to incorporate cell-level functionality as well as equipment-specific rules.

Several standards have been developed for communication between semiconductor equipment and host systems, including the Semiconductor Equipment Communications Standard (SECS), Generic Equipment Model (GEM), and SECS Message Service (SMS). These standards have lessened the need for equipment-specific communication but are based on semiconductor equipment capability and not on the behavior of host systems. As such, there has been no

semiconductor manufacturing operational model for host systems that can be applied to all equipment within a company, let alone all manufacturing companies within the semiconductor industry.

To remedy this discrepancy, SEMATECH created Baseline Requirements, a generic operational model for host systems that can be applied to all semiconductor manufacturing companies. The Baseline Requirements focus on cell controller responsibility.

A key assumption of Baseline Requirements is that a common set of requirements and high level designs can meet the needs of semiconductor manufacturing for all SEMATECH member companies. Baseline Requirements use scenarios to describe these common requirements and high level designs. The scenarios detail operational interactions involving the four major pieces of cell controllers.

Baseline Requirements also assume that there is no industry-wide definition for cell control, so that no company's manufacturing operation is considered wrong. The variation is in the management of the business or business rules.

Baseline Requirements advocate that cell controller applications incorporate all company-specific business rules, while equipment communication applications incorporate all equipment-specific rules. The interface between the two applications should be "atomic" to provide all functionality needed for cell applications to incorporate specific member company's business rules and yet be generic to all companies. The term "atomic" describes an activity that must either finish or, if interrupted, be resubmitted with no ill effects, so that the activity appears either "finished" or "never started."

NOTE: In some cases, semiconductor equipment companies have incorporated business rules into their operations. For example, Baseline Requirements define whether recipes are downloaded to the equipment or always stored on the equipment as a business rule. Some equipment require all recipes to be stored on the equipment with no download capability. Discrepancies like this may cause a business rule to be equipment-specific instead of company-specific. Baseline Requirements recognizes this inconsistency but still strives to encapsulate all business rules within the cell applications.

VFEI includes the message set and behavior characterization that describes Baseline Requirements logical interface between cell controller applications and equipment drivers. Baseline Requirements uses a machine driver to provide the physical interface between the cell controller application and the equipment. The machine driver maps the logical interface to the physical interface.

This interface uses atomic messages in an attempt to be equipment-independent as well as company independent. Using SEMATECH's Baseline Requirements to define VFEI as a consistent equipment interface to cell controller application provides the best approach to reusability within all companies.

3 USING THE STRATEGIC CELL CONTROLLER (SCC) BASELINES

Prior to VFEI 2.1, the SCC Baseline Requirements and Design Scenarios remained SEMATECH Confidential. Prior to the release of VFEI 2.2, the SCC Baseline Requirements were made public. Following is a brief introduction to the SCC Baseline Requirements.

All VFEI messages were derived during the creation of the Baseline Requirements. The Baselines provide the context for all VFEI messages. Before additional VFEI messages are adopted, the Baseline Requirements are visited to identify the requirements for the new message.

The process follows three steps:

1. Identify the context of the activity.
2. Describe implementation options.
3. Select specific technologies.

3.1 Identify Context in Activity Flow Charts

The Activity Flow Chart (AFC) is a diagram describing the manufacturing operational process.

There is an Activity Flow Chart for each of a number of points of view. For example, the manufacturing process from the point of view of a Machine is described in A300, Machine Life Cycle. This describes, at a high level, the life of a machine from commissioning through decommissioning, including machine processing and servicing.

There is also a life cycle for “Manufactured Goods” (a generic term for what everyone in a Wafer Fab knows is really just a “wafer”). Since Manufactured Goods travel in “groups” (a group of one is just a special case), there is an Activity Flow Chart, A037, Manufactured Goods Collection Operation.

Clearly, there is a fair amount of overlap among the Activity Flow Charts, in that activities are shared among the variety of objects whose point of views are described using AFCs. Machines and Manufactured Goods come together during most Machine Runs. Rather than running amok and creating Activity Flow Charts for everything in a business, the SCC Program focused primarily on those AFCs where Machines and business systems come together. The central Activity Flow Chart is A205, Machine Run. A205 is where materials, machines, resources and users combine to add value to manufactured goods.

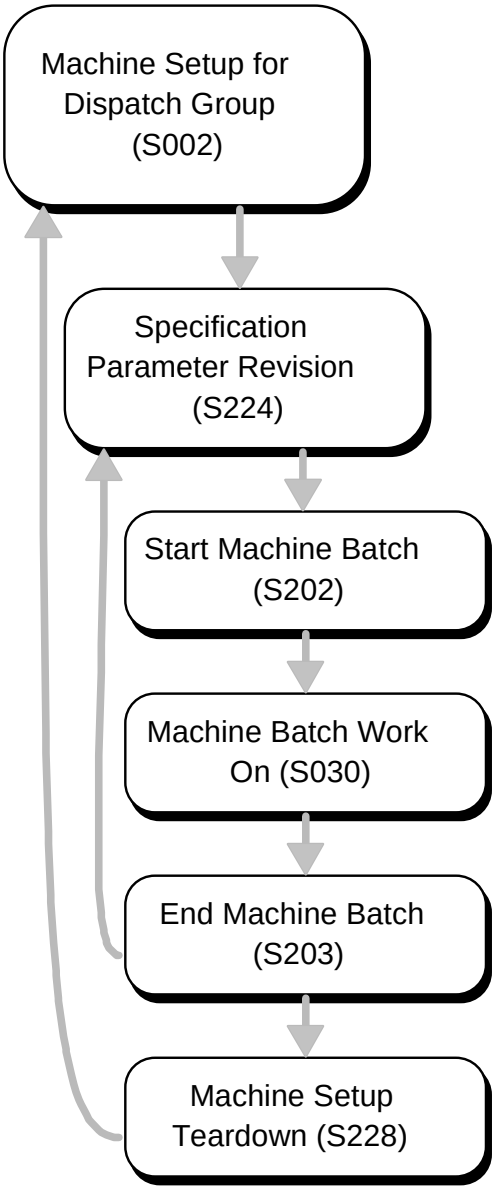


Figure 2 A205, Machine Run

The "bubbles" in A205, Machine Run are "Scenarios" and also are given identifiers, prefixed with the letter "S." The scenarios in A205 are as follows:

S002, Machine Setup for Dispatch Group This scenario actually contains another set of scenarios, such as *Tools Setup*, *Recipe Setup*, *Components Setup*, *Consumables Setup*, and *Display Specification*. These scenarios may be in any order, or may even be skipped, based on business rules.

A "Dispatch Group" is another group of Manufactured Goods (think "wafer" if you're a wafer fab type) that, by definition, all require the same machine setup.

S224, Specification Parameter Revision After a machine is set up for a run and before the first machine cycle starts, there may be an opportunity for some modification of the machine setup parameters, perhaps based on a previous machine cycle or material history. Scenario S224 includes two other scenarios to accomplish this, *Data Collection Mechanism* and *Perform Data Analysis*, as well as some steps specific to *S224, Specification Parameter Revision*. There is another Activity Flow Chart for data collection called *A001, Data Collection and Analysis.*

S202, Start Machine Batch Each machine cycle pulls material from a load port and processes the material in a group called a "Machine Batch." Each Machine Batch may be any number of goods (including none), and all goods in the Machine Batch receive processing simultaneously. Hence this is the smallest unit of processing typically found under control of a cell controller. Scenario S202 uses *Verify Setup* and *Select Machine Batch from Machine Batch Collection* scenarios. (A "Machine Batch Collection" is what is sitting on the load port.)

S030, Machine Batch Work On Finally, we're processing the machine batch. This scenario uses *In-Situ Process Control* and *Data Collection Mechanism*. Note that *Data Collection Mechanism* was also used in *S224, Specification Parameter Revision*. Yes, this is the same scenario.

S203, End Machine Batch When the batch is complete, the batch is evaluated using *Evaluate Machine Batch.*

S228, Machine Setup Teardown When all batches in a dispatch group are finished, the machine may be reset into an "idle" state using scenarios *Tool Removal, Recipe Removal, Components Removal* and *Consumables Removal*.

From this example, it becomes clear that the Activity Flow Charts do not concern themselves with any aspect of "who" or "what" performs any of the activities mentioned. Also, scenarios may be referenced in many places in the AFCs. This exemplifies the concept of "requirements reuse" into the SCC Baseline Requirements.

When a request for a new VFEI message is discussed, the first place to look, then, is the operational activity driving the new message. When is the message to be used? What is going on operationally when the message would be generated? Is the activity already documented in an Activity Flow Chart? If not, how does this activity fit? (Be careful with this question; as in most complex operational systems, operational stability must be maintained.)

If the activity is already documented in an Activity Flow Chart, how is the scenario implemented? This question leads the process into the next step.

3.2 Scenario Implementation Options

The scenarios described in the Activity Flow Charts are seen as what the Baseline Requirements calls “Level 0 (zero)” scenarios. No responsibilities are assigned to components in the system. As a result, the Level 0 scenarios are quite general and apply to a variety of implementation architectures.

The implementation architecture of interest to SEMATECH for the SCC Program is a three-level architecture containing four system components.

The lowest level is the Machine, the next higher level is the “Cell Controller” and the highest level is the MES (called “SFC” for “Shop Floor Controller” in the Baselines). The fourth system component is the User, who can touch each of the previous three components.

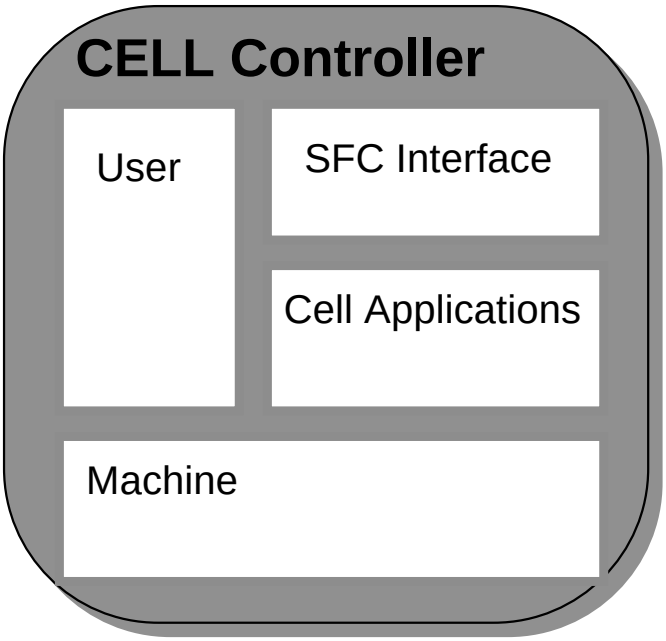


Figure 3 Four Level 0 System Components

Given these four system components, a set of implementation options can be described for each scenario. These options are called “Level 1 (one)” scenarios, and any number of them may be described for each Level 0 scenario, given the different roles the components may play.

As an example, *S193, Recipe Setup* , as a Level 0 scenario, simply states that to perform a recipe setup, one must identify the recipe, find the recipe, verify the recipe syntax, activate the recipe, and update our status. The Baseline Requirements and Design Scenarios — here is where the “design” begins — lists three implementation options:

1. S193.1, Recipe Setup (Recipe located on SCC)
2. S193.2, Recipe Setup (Recipe located on SFC)
3. S193.3, Recipe Setup (Recipe located on Machine)

Each of these Level 1 scenarios is described as a series of message flows among the four system components in the form of a ladder diagram.

For *S193.1, Recipe Setup (Recipe located on SCC),* the SCC identifies the recipe, checks if the recipe is on the machine, finds the recipe, and transfers it to the machine The machine then verifies the syntax, and then SCC activates the recipe on the machine and updates the status.

Messages begin to appear in this option, such as the request to activate the recipe on the machine from the SCC. In the course of a hundred or so such scenarios and implementation options, a core set of messages appear among the various system components.

By focusing on the messages between SCC and the machine, VFEI is created.

At this point, no specific technologies are considered (such as what protocol is used to communicate to the machine from the SCC). Nor is any variation taken into account based on machine type or operational specifics of the machine. Herein lies the true promise of VFEI—the "Virtual Factory Equipment" part of VFEI does not contain any machine or communication protocol specifics.

3.3 Technology Selection

In any true implementation, specific technologies must be chosen, and the Baseline Requirements provide an example of a single technology selection—the "Generic Equipment Model" for equipment communications, based on the SEMI SECS-II protocol.

The final step in scenario analysis is the "Level 2 (two)" scenario. In the Baselines, an example of a GEM-compliant machine is described for many (but not all) Level 1 scenarios.

For example, *S193, Recipe Setup* is broken further to Level 2 scenarios:

1. S193.1.1, Recipe Setup (Recipe located on SCC) (GEM Machine)
2. S193.2.1, Recipe Setup (Recipe located on SFC) (GEM Machine)
3. S193.3.1, Recipe Setup (Recipe located on Machine) (GEM Machine)

In reality, the only part of the Level 1 scenario that changes with this particular form of Level 2 scenario is that the "Machine" component is split into two parts, the *Machine Interface,* and the *GEM Machine.* These two parts communicate using SEMI SECS II GEM-compliant messages. A *Machine Interface* remains on the host and is implemented as a "Machine Driver."

3.4 Special Considerations

The step in the process described above that is most pertinent to VFEI is the creation of the Level 1 scenarios. When allocating responsibilities to system components, some guidelines are warranted to maintain some consistency in the application of the process.

Reuse is a special consideration when deriving the Baseline Requirements. When a cell controller is moved from one site to another, it is useful to look at what varies during the move.

The first variable is the machine being connected. If the cell controller is running the same business using the same technology, only a single system component should be changed. This component is the *Machine Interface* identified in the Level 2 scenarios.

The second variable is the business operation implemented. If the same machine interface is implemented at two different companies, the rules of the business will change, and these business rules should be isolated to simplify this change. The *Machine Interface* should not embody any

of these business rules, or it would not be reusable from business to business for a given machine.

The third variable considered by the Baselines is the system technology used to implement the scenarios. While this does not directly affect VFEI, it is the primary reason why the VFEI specification text is arranged as a “semantic” specification, with implementation options listed as appendices.

4 VFEI CONCEPTS

The following is a set of concepts that describe the intent of VFEI. These concepts provide a framework to enhance interface reuse, make explicit the interface structure, and assist developers in VFEI use.

4.1 Design Concepts

VFEI design concepts include the following:

- SEMATECH’s Baseline Requirements define VFEI messages. VFEI provides the interface between cell controller applications and equipment in the Level 1 Baseline Requirements diagrams (logical partition between Shop Floor System, Cell Application, User, and Equipment).
- Any changes to VFEI should be reflected in Baseline Requirements. There are some known inconsistencies between VFEI and the Baselines (e.g., bidirectional DISPLAY MESSAGE capability and the new MOVE command). A supplement to the Baselines is planned to address these issues.
- VFEI isolates equipment-dependent software (the machine driver) from cell controller applications. Equipment-dependent software should not include the machine state model.
- Baseline Requirements implies a machine state model. In simplest terms, Baseline Requirements expect machines will have INIT (uninitialized), IDLE, EXECUTING, and PAUSE states. Note that there is no requirement that the machines implement the specified states. However, it is expected the actual machine states will have common aspects to the Baseline Requirements' machine states.
- VFEI is equipment communications protocol independent. VFEI is compatible with SECS and GEM, but SECS and GEM are not required for a VFEI interface.

4.2 Messaging Concepts

- Message routing is simple. VFEI presumes a hierarchy of cell controller applications over a machine. The machine driver will have a known address through which it receives command and reply messages from cell controller applications. The machine driver also will have knowledge of an external address to which all unsolicited messages and command messages from the equipment will be sent.
- This hierarchy maps to client/server architecture. The cell controller application and the machine driver can be both a client and a server.
- VFEI supports bi-directional command/reply messages. Commands from cell controller applications are sent to the machine driver's address. The reply message is sent to the sender's

return address. Commands from the machine are sent to the defined external address and replies are expected at the machine driver's address.

- VFEI supports unsolicited messages from the machine driver to the cell controller applications. All unsolicited messages from a machine are routed to a single defined address. Unsolicited messages do not receive a response message. Typically, unsolicited messages just pass information. The most common use of unsolicited messages is an event that notifies changes in the state model.
- VFEI does not provide subscription services. The cell application must provide subscription service or further routing of unsolicited or machine commands, if desired.
- VFEI treats the two message models independently. That is, VFEI does not have a contextual association to relate a command/reply to the later unsolicited message. Any contextual association is within the domain of the Baseline Requirements. NOTE: VFEI treats the message models independently because state changes may occur even if no command is received. Hence, events cannot be treated unilaterally as replies.
- VFEI requires a reliable transport mechanism to implement the messaging between it and cell controller applications. This implies that issues such as guaranteed delivery, message loss detection, and time-outs are handled by the messaging system.

4.3 Message Concepts

Message concepts include the following:

- VFEI messages are available for each piece of equipment. They have the same names, types, and sizes and semantic meaning for each piece of equipment.
- Some equipment does not support capabilities implemented by VFEI commands. In these cases, the VFEI message is still defined but will result in a reply message with a 'not supported' error code.
- Command messages result in reply messages with error codes. An error code of SUCCESS indicates command success. All other error codes indicate a failure of some kind. Reply messages also return error text. This document defines common error codes sent in the reply messages. However, a returned error code may be a specialized error code for the equipment. Cell applications should gracefully handle such error codes.
- The rest of the VFEI messages are kept at a minimum. Wherever possible, data modeling techniques are used to reduce several similar commands to a single command.
- Messages that change the implied machine states are uniquely identified. These command IDs are the following: INITIALIZE, START, ABORT, PAUSE, and RESUME.
- Additional commands may be sent than those specified. The additional commands are considered equipment-specific and use the MACHINE COMMAND message.
- Equipment commands that can result in a state change may receive an event in addition to the reply message. If the reply to the command is success, then the equipment may return an event in an EVENT REPORT message. The event indicates that the actual command has been activated on the equipment, since these commands are interpreted as "request action be initiated" rather than "do action."
- Equipment may send additional events than those specified in the Baseline Requirements. cell controller applications should gracefully handle these events.

4.4 Message Data Items

VFEI messages may contain several types of data items. These consist of standard data items, predefined data items, and user defined data items. All data items are self-defining; that is, they are identified by a name, type, size, and value. Each type is described below.

4.4.1 Standard Data Items

VFEI messages contain standard data items. The standard data items are used by all VFEI messages and are required to be present. A standard data item has the same name, type, size, and semantic meaning for each piece of equipment. Additional data items (either predefined or user defined) can appear in any order after the standard fields. VFEI message standard fields are listed in the following tables for each message type. A description of each field follows.

***** OBSOLETE ***** In the previous VFEI specifications, the standard data items had to be provided in a pre-described order. This is considered to be an implementation issue.

Table 1 VFEI Command Message Standard Data Items

Name	Type
COMMAND_ID	ASCII
MACHINE_ID	ASCII
MSG_TYPE	ASCII
TID	UNSIGNED 4

Table 2 VFEI Reply Message Standard Data Items

Name	Type
COMMAND_ID	ASCII
MACHINE_ID	ASCII
MSG_TYPE	ASCII
TID	UNSIGNED 4
ERROR_CODE	UNSIGNED 4
ERROR_TEXT	ASCII

Table 3 VFEI Unsolicited Message Standard Data Items

Name	Type
COMMAND_ID	ASCII
MACHINE_ID	ASCII
MSG_TYPE	ASCII
ERROR_CODE	UNSIGNED 4
ERROR_TEXT	ASCII

COMMAND_ID	Identifies the semantic content of the message—that is, which data item fields, according to the VFEI definition, exist in the message. Command ID is required for all messages.
MACHINE_ID	Identifies the equipment associated with the message. Machine ID is required for all messages.
MSG_TYPE	Identifies the message type of the command. Valid values are the following: command messages, reply messages, and unsolicited messages. Message Type is required for all messages.
TID	Transaction ID. Identifies a specific instance of a message. It is used to associate a specific reply with its respective command. Transaction ID is required for command messages and reply messages. Transaction ID is set by the sender of the command message. Reply message echoes the transaction ID of the command message. Transaction ID is not applicable for unsolicited messages.
ERROR_CODE	Status return value. Indicates success by a value of SUCCESS. Error code is required in reply messages and unsolicited messages. Error code is not applicable for command messages.
ERROR_TEXT	Status text describing the error. Applies to reply and unsolicited messages. Error text is not applicable to command messages.

4.4.2 Predefined Data Items

Predefined data items are those data items beyond standard message fields. VFEI commands that make use of predefined elements will indicate the data item name, size, and brief description. Optional predefined data items are common across all equipment. For example, the **ABORT** VFEI command has the following predefined data item defined:

[LEVEL]	UNSIGNED 2	Abort level to perform

The brackets ([]) around LEVEL indicate it is an optional data item that may appear zero or one time.

4.4.3 User Defined Data Items

User defined data items provide a VFEI command with the ability to handle equipment-specific capabilities. User defined items are described in VFEI commands in the following manner:

[<user defined name>]*

The brackets ([]) indicate an optional data item. The asterisk (*) means the item may appear zero or more times.

In some instances, a specific equipment implementation may require user defined data items with a message. For example, the START command for equipment A might require a user defined data item MBC_ID. Equipment B may require the user defined data items MBC_ID and PPID. In order to properly function with either tool, the host system may want to use both MBC_ID and PPID in the START command.

No VFEI error message should be generated if extra user defined data items are included with a command. In the above example, if equipment A was being used and a START command with both MBC_ID and PPID was sent from the host, then only MBC_ID would be passed along to the equipment. PPID would be ignored and no error for this extra item would be generated.

4.4.4 VFEI Message Data Item Types and Values

The data item types and values are described in Table 4.

5 VFEI EVENTS

Tables 5 and 6 provide the list of VFEI events and their recommended naming designation.

Table 5 lists those events required in VFEI implementations. It is realized not all equipment may provide these required events. VFEI drivers must provide these events using whatever means Table 6 lists events that are not required, but are commonly used. If the equipment supplies an equivalent event, then the VFEI implementation should use the VFEI events. VFEI implementations may define additional events relevant to specific equipment.

The "Used By" column indicates whether the Baselines (B), GEM (G; SEMI E30-95), or both require the event.

Table 4 VFEI Message Data Item Types and Values

Data Item Type	Data Item Values
FLOAT 8	Used for a data item whose value requires the IEEE-754 format.
FLOAT 4	Used for a data item whose value requires the IEEE-754 format.
UNSIGNED 1	Used for a data item whose value is between 0 and 255 inclusively.
UNSIGNED 2	Used for a data item whose value is between 0 and 65535 inclusively.
UNSIGNED 4	Used for a data item whose value is between 0 and 4294967295 inclusively.
UNSIGNED 8	Used for a data item whose value is between 0 and 18446744073709551615 inclusively.
SIGNED 1	Used for a data item whose value is between -128 and 127 inclusively.
SIGNED 2	Used for a data item whose value is between -32768 and 32767 inclusively.
SIGNED 4	Used for a data item whose value is between -2147483648 and 2147483647 inclusively.
SIGNED 8	Used for a data item whose value is between -9223372036854775808 and 9223372036854775807 inclusively.
ASCII	Used for a data item whose value is a sequence of ASCII characters.
BOOLEAN	Used for a data item whose value is either TRUE or FALSE.
BINARY	Used for a data item whose value is between 0 and 255 inclusively.
LIST	Used for a data item whose value is list of items.

Table 5 Required VFEI Events

Event Designation	Event Description	Used By
ABORT	Processing aborted	B
COM_DISABLE	Communications with the equipment has been terminated	
COM_ENABLE	Communications with the equipment has been established	
MB_COMPLETE	Processing completed for a machine batch	B
MB_START	Machine batch processing started	B
MBC_COMPLETE	Processing completed for a machine batch collection	B
MBC_START	Processing started for a machine batch collection	B
MESSAGE_RECEIVE	Terminal message sent is acknowledged by operator	B, G
PAUSE	Processing suspended	B
RESOURCE_ACTIVATE	Resource activate complete. Same as the GEM required PP_SELECTED	B, G
RESUME	Processing resumed	B

Table 6 Common VFEI Events

Event Designation	Event Description	Used By
ALARM_DETECT	Alarm has been set	G
ALARM_CLEAR	Alarm has been cleared	G
COMMAND_ISSUE	Indicates operator activity while in the remote state	G
COMPLETE	Processing has completed normally	G
CTRL_STATE_LOCAL	Equipment is in the local control state	G
CTRL_STATE_REMOTE	Equipment is in the remote control state	G
EC_CHANGE	An equipment constant has been changed by the operator	G
EQUIPMENT_OFFLINE	The equipment has transitioned to the offline state	G
ID_RECEIVE	The ID of the Transport Group has been obtained	
MATERIAL_RECEIVE	Material was received	G
MATERIAL_REMOVE	Material was removed	G
PP_CHANGE	Indicates the operator has performed a process program change	G
PROC_STATE_CHANGE	A processing state change has occurred on the equipment	G
RESOURCE_VERIFY	Equipment has verified the downloaded resource	
START	Equipment has entered the EXECUTING state	G
STATE_CHANGE	A state change has occurred on the equipment	G
STOP	Processing on the equipment has stopped in response to a STOP command	G
TG_ARRIVE	The Transport Group has arrived at its destination location	
TG_SENSE	Transport Group is sensed at a port	

6 VFEI VARIABLES

Tables 7, 8, and 9 provide the list of VFEI variables and their recommended naming designation.

Tables 7 and 8 are required in VFEI implementations. Table 7 lists those variables required for reporting from the equipment and Table 8 lists variables the VFEI driver must accept. The variables in Table 8 are used to configure the equipment driver and do not pass through to the equipment. It is realized not all equipment may support these required variables. VFEI drivers must provide these variables using whatever means possible.

Table 9 lists variables that are not required, but are commonly used. If the equipment supplies an equivalent variable, then the VFEI implementation should use the VFEI variables.

The VFEI commands that make use of the VARIABLE_ID data item, i.e., EVENT SETUP, STATUS QUERY, and TRACE SETUP, must support these variables.

EVENT REPORT, STATUS LIST, and TRACE_REPORT, also must support these variable names in the [<user defined name>] * data items. Other commands may also use these variables for any [<user defined name>]* data items.

The “Used By” column indicates whether the Baselines (B), GEM (G; SEMI E30-95), or both require the variable.

Table 7 Required VFEI Equipment Variables

Variable ID	Variable Description	Used By
CLOCK	Equipment’s date/time of day clock	B, G
COM_STATE	Communications state (communicating or not-communicating)	
CONTROL_STATE	Current control state (local or remote) of the equipment	G
EQ_MODEL_ID	Equipment model ID	
EQ_SOFT_VER	Equipment software version number	
MB_ID	Machine batch ID	
MBC_ID	Machine batch collection ID	
MBC_SIZE	Machine batch collection size	
PP_EXEC_NAME	Process program in the execute state	B, G
PROCESS_STATE	Current process state of the equipment	B, G

Table 8 Required VFEI Driver Variables

Variable ID	Variable Description	Used By
VFEI_CMDS	List of VFEI commands supported	
VFEI_REL	VFEI driver release (e.g. 0.5)	
VFEI_VER	VFEI specification version (e.g. 2.2)	

Table 9 Common VFEI Variables

Variable ID	Variable Description	Used By
ABORT_LEVEL	Level of the most recent abort	
ALARM_ID	ID of the most recent alarm that went ON or OFF	G
ALARMS_ENABLED	Contains the list of alarms enabled for reporting	G
ALARMS_SET	A list of alarms currently in the set state	G
COM_TIMER	Time interval, in seconds, between attempts by the equipment to establish communications. Same as the GEM specified EstablishCommunicationsTimeout variable	G
DIR_UPLOAD	Where to put (i.e. directory path) an unsolicited resource transfer	
EVENTS_ENABLED	The list of events enabled for reporting	
HEART_BEAT	Time interval, in seconds, between attempts by the VFEI driver to verify the communications link is operating	
INIT_COMM_STATE	Equipment's initial communication state (enabled or disabled)	
INIT_CTRL_STATE	Equipment's initial control state (local or remote)	
LAST_EVENT	Event ID of the most recent event	
LOCATION	Name of the physical position on a piece of equipment at which a transfer object may reside	
PORT	A point on equipment at which a change of ownership of a transfer object occurs.	
PP_CHANGE_NAME	Process program ID of the most recently created, edited, or deleted process program	G
PP_CHANGE_STATUS	How the process program "PP_CHANGE_NAME" was changed (created, edited, or deleted)	G
PREV_PROC_STATE	Previous process state of the equipment	G
STATION	An addressable sub-assembly of a machine ID	
TG_ID	Transport group ID. Identifier for material that is being moved from a source to destination location.	
TG_TYPE	Transport group type. Type of material that is being moved from a source to destination location.	
UPLOAD_TYPE	How to store (i.e., ASCII, binary.) an unsolicited resource transfer	
VERBOSITY	Debug level. Effects verbosity of reporting or logging	

7 VFEI PREDEFINED ERROR CODES

The following tables in this section provide a list of predefined error codes defined in VFEI messages. Additional error codes may be added to the following lists as necessary to meet specific application needs.

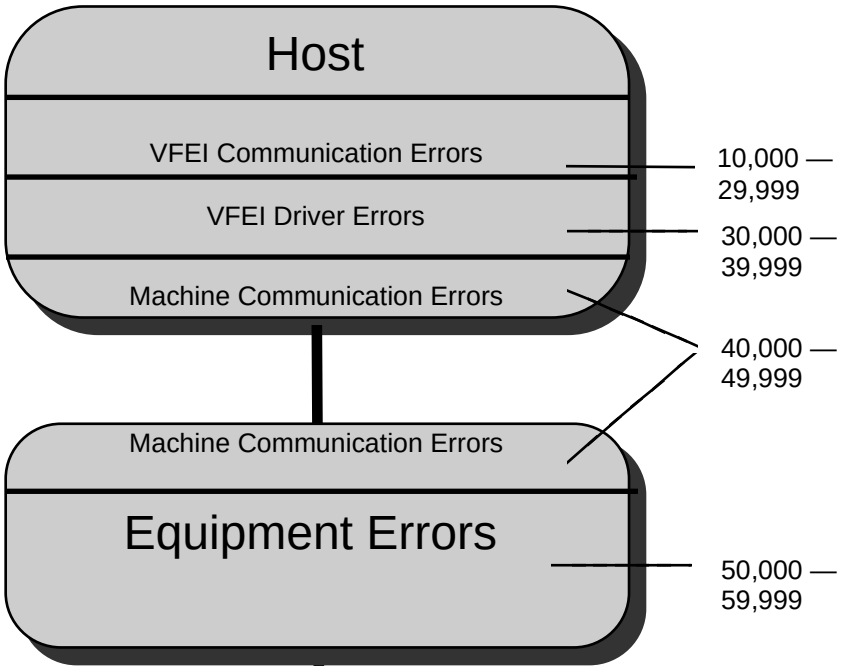
Error codes are grouped into four different categories to reflect errors that occur at different levels in a VFEI application. The four categories are described below.

Error Description	Error Number Range
VFEI Communication Errors	10000-29999
VFEI Driver Errors	30000-39999
Machine Communication Errors	40000-49999
Equipment Errors	50000-59999

Definitions for each category are as follows:

VFEI Communication Errors:	Any error in the format or syntax of a VFEI message. The errors are NOT returned by the VFEI driver, but are generated by the bus or Protocol driver that interfaces with the VFEI driver.
VFEI Driver Errors:	Any error generated by the VFEI driver. These errors may be execution or logic errors. This also may be used to indicate that a particular VFEI command is not supported by that particular VFEI driver.
Machine Communication Errors:	Any error generated at the machine communication level. These may be communication syntax errors, communication driver errors, communication link errors, etc.
Equipment Errors:	Any error generated by the tool. These errors indicate that the message was received by the tool, but rejected by the tool for the reason indicated in the response message.

The following diagram provides an architectural view of where the four error categories would fit in a typical VFEI implementation.



NOTE: In all error categories, an Error Code Value of 0 indicates Success.

Figure 4 Error Code Configuration

Table 10 VFEI Communication Error Codes

Error Designation	Error Code Value	Error Explanation
VFEI_MSG_SYNTAX_ERROR	10026	Error receiving VFEI Command message.
VFEI_PROTOCOL_ERROR	10027	Error sending VFEI Command message.
VFEI_TIMEOUT	10028	Error receiving VFEI Command reply message.
VFEI_FILE_SYNTAX_ERROR	10029	Error sending VFEI Command reply message.

Table 11 VFEI Driver Error Codes

Error Designation	Error Code Value	Error Explanation
VFEI_UNKNOWN_ERROR	30000	Unknown error has occurred.
VFEI_INVALID_FORMAT	30007	The parameter or message format is incorrect.
VFEI_VID_NOT_EXISTS	30009	A VID has been specified that does not exist.
VFEI_EVENT_NOT_EXISTS	30011	An Event has been specified that does not exist.
VFEI_UNEXPECTED_EVENT	30013	An Event that was not expected was received from the machine.
VFEI_MISSING_PARAM	30014	Parameter required to perform the command is missing.
VFEI_REQUEST_DENIED	30018	Command request has been denied.
VFEI_CMD_NOT_SUPPORTED	30019	This command is recognized but not supported by the equipment.
VFEI_UNEXPECTED_VALUE	30025	The value of one of the parameters was unexpected or out of range.
VFEI_FILE_IO_ERROR	30030	VFEI File I/O Error.

Table 12 VFEI Machine Communication Error Codes

Error Designation	Error Code Value	Error Explanation
COMM_MSG_SYNTAX_ERROR	40026	Syntax of the message to the equipment is incorrect.
COMM_PROTOCOL_ERROR	40027	A low level communication error has occurred.
COMM_TIMEOUT	40028	A low level communication timeout has occurred.

Table 13 VFEI Equipment Error Codes

Error Designation	Error Code Value	rror Explanation
EQUIP_UNKNOWN_ERROR	50000	Unknown error has occurred.
EQUIP_INVALID_COMMAND	50001	Equipment cannot perform command
EQUIP_CANNOT_DO_NOW	50002	Command cannot be performed at this time.
EQUIP_INVALID_PARAM	50003	Invalid parameter specified with the command.
EQUIP_INVALID_C_STATE	50004	Equipment is in an invalid control state for accepting the specified command. Equipment must be in a control state that accepts the remote command.
EQUIP_INVALID_P_STATE	50005	Equipment is in an invalid process state for accepting the specified command. The current process state does not accept the remote command.
EQUIP_NOT_ENOUGH_SPACE	50006	Not enough room to store the resource.
EQUIP_INVALID_FORMAT	50007	Message format is not correct.
EQUIP_REPORT_EXISTS	50008	The equipment already has a report defined.
EQUIP_VID_NOT_EXISTS	50009	At least one VID specified does not exist.
EQUIP_LINK_EXISTS	50010	At least one event link is already defined on the equipment.
EQUIP_EVENT_NOT_EXISTS	50011	At least one CEID specified does not exist.
EQUIP_REPORT_NOT_EXISTS	50012	At least one report id specified does not exist.
EQUIP_UNEXPECTED_EVENT	50013	*****OBSOLETE***.** This is now represented by VFEI_UNEXPECTED_EVENT.
EQUIP_MISSING_PARAM	50014	Parameter required to perform the command is missing.
EQUIP_ALREADY_HAVE	50015	Current command has already been specified.
EQUIP_WILL_NOT_ACCEPT	50016	The equipment will not accept the current command.
EQUIP_INVALID_TYPE	50017	Type specified is not valid for the equipment.
EQUIP_REQUEST_DENIED	50018	Command request has been denied.
EQUIP_CMD_NOT_SUPPORTED	50019	*****OBSOLETE***.** This error is already supported by EQUIP_INVALID_COMMAND
EQUIP_INCOMPLETE_RESPONSE	50020	Used to indicate incomplete data reported in STATUS LIST or RESOURCE LIST.
EQUIP_LENGTH_ERROR	50021	*****OBSOLETE*****
EQUIP_MATRIX_OVERFLOW	50022	*****OBSOLETE*****
EQUIP_RESOURCE_NOT_FND	50023	Resource specified was not found.
EQUIP_MODE_UNSUPPORTED	50024	*****OBSOLETE*****
EQUIP_UNEXPECTED_VALUE	50025	The value of one of the parameters was unexpected or out of range.

Returning the specific equipment error codes in the ERROR_TEXT field may be used to provide more information about errors in the Equipment Error category.

8 VFEI COMMANDS

This section describes the commands available in VFEI. Each command description is formatted as shown below. It includes a message data item list, message type, reply message if applicable, data item explanation, Baseline reference, command explanation, examples, reply, and error codes.

8.1 Command Synopsis (Message Direction)

Data Item Name	Type (from Table 4)	ASCII syntax Value for Data Item
COMMAND_ID	ASCII	*VFEI COMMAND*
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	*C for command* *R for reply* *E for unsolicited*
TID	UNSIGNED 4	<transaction_id>
[pre defined data item name]	predefined data item format	<value>
[<user defined name>]*	user defined format	<value>

The message data items are listed in a tabular form. Where there are multiple forms of the same command, several separate tables are presented.

The notation used within the tables is as follows:

<value> Angle brackets indicate a value that must be supplied.

<data item>+ A plus indicates a data item that must appear one or more times.

[<data item>] Square brackets indicate an optional data item that may appear zero or one time.

[<data item>]* Square brackets with an asterisk indicate an optional data item that may appear zero or more times.

| OR: indicates a set of mutually exclusive values that are valid.

Message Type: *Command* Reply Message: *VFEI REPLY*

Message type is in terms of command, reply, and unsolicited messages as described in the VFEI rules. If the message type is command, then the reply message is listed.

8.1.1 Predefined Data Items

Data item explanations provide information on all predefined data items beyond standard message fields that are available within the message. Data item name and size are provided with a brief description.

NAME Description of data item.

8.1.2 Baseline Reference

This is a reference to a section in the SEMATECH Baseline document. The Baseline reference is provided so the command's usage within the manufacturing context can be explored.

8.1.3 Description

Command information describes how the message is used. This includes a detailed description of logic performed within equipment communications.

8.1.4 Examples

Examples are provided for each message. Examples show message content, which includes standard fields and any predefined optional data items. Examples use the syntax of an ASCII bus, as defined in Appendix C.

8.1.5 Reply

For command messages, this section explains a reply. Reply explanation describes the reply message expected and standard error codes returned. The normal reply message for most VFEI commands is COMMAND ACK.

This section only provides applicable Equipment Error Codes and VFEI Driver Error Codes. VFEI Communication Errors, Machine Communication Errors, and implementation specific error codes may also be returned in addition to the listed applicable error codes.

Error codes are logical error codes. Error codes are described by symbolic reference. Refer to the section on predefined error codes.

VFEI REPLY returns the following data items:

ERROR_CODE

ERROR_TEXT

8.2 ABORT

Abort Processing (H->E)

Data Item	Type	Value
COMMAND_ID	ASCII	ABORT
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C
TID	UNSIGNED 4	<transaction_id>
[LEVEL]	UNSIGNED 2	<value>
[<user defined name>]*	<format>	<value>

Message Type: Command Reply Message: COMMAND ACK

8.2.1 Predefined Data Items

LEVEL Abort level to perform.

8.2.2 Baseline Reference

This command is documented in Baseline scenario S253, Abort Machine Run.

8.2.3 Description

ABORT terminates the current activity prior to its completion. Abort immediately stops the process, generally because of abnormal conditions. Abort makes no guarantee about the subsequent condition of the material.

Some equipment can terminate the current cycle at various points. One point may be at the next “safe break point,” at which the equipment retrieves all material, stops in a safe condition, and returns to the idle state. Abort level determines this capability. LEVEL is a general equipment-specific capability, and therefore LEVEL is a predefined optional parameter.

The user-defined data items give the ABORT message any equipment-specific capability.

8.2.4 Examples

CMD/A="ABORT" MID/A="STP1" MTY/A="C" TID/U4=1

CMD/A="ABORT" MID/A="TRK1" MTY/A="C" TID/U4=2 LEVEL/U2=3 MG/A="WAFER1"

8.2.5 Reply

The machine driver returns a COMMAND ACK message for the ABORT command. This return indicates whether the equipment has accepted or rejected the ABORT command. When the equipment's current activity has terminated, the equipment sends an EVENT REPORT with an event type of ABORT.

COMMAND ACK returns the following error codes:

EQUIP_INVALID_COMMAND
ABORT not valid for this equipment.

EQUIP_CANNOT_DO_NOW
ABORT cannot be performed at this time. Try again later.

EQUIP_INVALID_PARAM
Invalid parameter specified with the abort.

EQUIP_INVALID_C_STATE
The equipment is in an invalid control state for accepting ABORT. The equipment must be in a control state that accepts the remote abort command.

EQUIP_INVALID_P_STATE
The equipment is in an invalid process state for accepting ABORT. The current process state does not accept an ABORT command.

8.3 ALARM REPORT

Alarm Report (E->H)

Data Item	Type	Value
COMMAND_ID	ASCII	ALARM_REPORT
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	E
ERROR_CODE	UNSIGNED 4	<error_code>
ERROR_TEXT	ASCII	<error_text>
ALARM_ID	SIGNED 4	<value>
ALARM_STATE	UNSIGNED 1	0 | 1
[ALARM_TEXT]	ASCII	<value>
[ALARM_CATEGORY]	UNSIGNED 1	<value>
[<user defined name>]*	<format>	<value>

Message Type: Unsolicited

8.3.1 Predefined Data Items

ALARM_ID	Alarm ID
ALARM_STATE	Alarm state: 0 = ALARM CLEAR 1 = ALARM SET
ALARM_TEXT	Text associated with the alarm
ALARM_CATEGORY	Category of the alarm (e.g., personal safety, equipment safety)

8.3.2 Baseline Reference

This command is documented in Baseline scenario S212, Machine Event and Alarm Management.

8.3.3 Description

ALARM REPORT reports changes in alarm state for alarms enabled with the ALARM SETUP command. The equipment manufacturer is responsible for documenting all alarms on their equipment. Cell application is responsible for enabling alarm reports with the ALARM SETUP command.

Notification occurs when the alarm changes state. Alarms have two states, ALARM CLEAR or ALARM SET. Transition from ALARM CLEAR to ALARM SET sends the ALARM REPORT message with alarm state in the “set” state. Transition from ALARM SET to ALARM CLEAR sends the ALARM REPORT message with alarm state in the "cleared" state.

ALARM_TEXT is provided for equipment that sends an alarm text with alarm notification.

ALARM_CATEGORY is provided for equipment that sends the alarm category (e.g., personal safety, equipment safety) with alarm notification.

The user-defined data items give the ALARM REPORT message any equipment-specific capability.

8.3.4 Examples

CMD/A="ALARM_REPORT" MID/A="STP1" MTY/A="E" ECD/U4=0 ETX/A="" ALARM_ID/I4=353 ALARM_STATE/U1=1 ALARM_TEXT/A="WAFER ARM JAMMED"

8.3.5 Error Codes

None defined.

8.4 ALARM SETUP

Enable/Disable Alarm Reporting (H->E)

Data Item	Type	Value
COMMAND_ID	ASCII	ALARM_SETUP
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C
TID	UNSIGNED 4	<transaction_id>
ENABLE	UNSIGNED 1	<value>
[ALARM_ID]*	SIGNED 4	<value>
[<user defined name>]*	<format>	<value>

Message Type: Command Reply Message: COMMAND ACK

8.4.1 Predefined Data Items

ENABLE Enable flag: 0 = DISABLE
1 = ENABLE

ALARM_ID Alarm ID

8.4.2 Baseline Reference

This command is documented in Baseline scenario S255, Enable Alarm Collection.

8.4.3 Description

ALARM SETUP enables or disables reporting transitions of alarm states on equipment. ALARM REPORT command contains the alarm reports. Enabling or disabling a given alarm impacts communication of both set and clear messages. The alarm itself is not being enabled or disabled, but the reporting of the alarm is enabled or disabled. See the ALARM REPORT command for information on alarm reporting.

This command has the ability to enable/disable more than one alarm. If no alarm IDs are listed, then all alarm reporting on the equipment is enabled/disabled.

The user defined data items give the ALARM SETUP message any equipment-specific capability.

8.4.4 Examples

CMD/A="ALARM_SETUP" MID/A="STP1" MTY/A="C" TID/U4=1
ENABLE/U1=0

CMD/A="ALARM_SETUP" MID/A="STP1" MTY/A="C" TID/U4=2
ENABLE/U1=0 ALARM_ID/I4[3]=[1 2 3]

8.4.5 Reply

The equipment returns a COMMAND ACK message for the ALARM SETUP. This return indicates whether the ALARM SETUP command was accepted or rejected.

COMMAND ACK returns the following error codes:

EQUIP_INVALID_PARAM
Invalid parameter specified with the alarm setup command.

8.5 COMMAND ACK

Command Acknowledgment (E->H)

Data Item	Type	Value
COMMAND_ID	ASCII	CMD_ACK
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	R
TID	UNSIGNED 4	<value>
ERROR_CODE	UNSIGNED 4	<error_code>
ERROR_TEXT	ASCII	<error_text>

Message Type: Reply

8.5.1 Predefined Data Items

None

8.5.2 Baseline Reference

This command has no Baseline reference.

8.5.3 Description

The equipment sends the COMMAND ACK as the generic command acknowledgment in response to command message types. This reply message is used only if the reply can return only a status. If the reply message can return variables, then a message with a unique command ID is used. These unique reply messages are identified in the message description for those messages that have a reply with variables.

Error text is returned only if the error code is not success. Success is indicated by a value of SUCCESS. See the command messages for error codes that are sent in response.

8.5.4 Examples

CMD/A="CMD_ACK" MID/A="STP1" MTY/A="R" TID/U4=1 ECD/U4=0 ETX/A=""

CMD/A="CMD_ACK" MID/A="STP1" MTY/A="R" TID/U4=1 ECD/U4=50000 ETX/A="EQUIP_UNKNOWN_ERROR"

8.6 DISPLAY MESSAGE

Display Message on Equipment (H<->E)

Data Item	Type	Value
COMMAND_ID	ASCII	DISPLAY_MSG
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C | E
TID	UNSIGNED 4	<transaction_id>
TERMINAL	BINARY 1	<value>
[MESSAGE]*	ASCII	<value>
[<user defined name>]*	<format>	<value>

Message Type: Command Reply Message: COMMAND ACK (H<->E)

8.6.1 Predefined Data Items

TERMINAL Terminal to display message

MESSAGE Message to display

8.6.2 Baseline Reference

This command is documented in Baseline scenario S225, Display Specification.The capability of receiving this command from the equipment is not currently documented in the Baseline scenarios.

8.6.3 Description

DISPLAY MESSAGE allows the cell controller to display information on the equipment's display device. The equipment must be capable of displaying information passed to it by the cell controller. The equipment has no responsibility for interpreting any information passed from the host in DISPLAY MESSAGE.

NOTE: Some equipment provides an event for message recognition, which indicates that the operator has actually viewed the information. The event has to be enabled with the EVENT SETUP message for the equipment to send the EVENT REPORT message with EVENT_ID of MESSAGE_RECEIVE to the cell controller.

DISPLAY MESSAGE also allows the cell controller to receive information that was entered at the equipment's display device.

The user defined data items give the DISPLAY MESSAGE message any equipment-specific capability.

8.6.4 Examples

CMD/A="DISPLAY_MSG" MID/A="TRK1" MTY/A="C" TID/U4=2 TERMINAL/B=0x00 MESSAGE/A="SETUP EQUIPMENT"

CMD/A="DISPLAY_MSG" MID/A="TRK1" MTY/A="C" TID/U4=2 TERMINAL/B=0x01 MESSAGE/A="START"

CMD/A="DISPLAY_MSG" MID/A="TRK1" MTY/A="E" TID/U4=2 TERMINAL/B=0x01 MESSAGE/A="THIS IS A MESSAGE FROM THE EQUIPMENT"

8.6.5 Reply

The equipment/host returns a COMMAND ACK message for the DISPLAY MESSAGE command. This return indicates whether the DISPLAY MESSAGE command was accepted or rejected. When the message is displayed (H->E), then the equipment can send an EVENT REPORT with an EVENT_ID of MESSAGE_RECEIVE.

COMMAND ACK returns the following error codes:

EQUIP_CANNOT_DO_NOW
Terminal not available.

EQUIP_REQUEST_DENIED
Message will not be displayed.

8.7 EVENT REPORT

Event Occurs on Equipment (E->H)

Data Item	Type	Value
COMMAND_ID	ASCII	EVENT_REPORT
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	E
ERROR_CODE	UNSIGNED 4	<error_code>
ERROR_TEXT	ASCII	<error_text>
EVENT_ID	ASCII	<value>
[FILE]	ASCII	<value>
[<user defined name>]*	<format>	<value>

Message Type: Unsolicited

8.7.1 Predefined Data Items

EVENT_ID Event ID

FILE Full file specification for file that contains the data.

8.7.2 Baseline Reference

This command is documented in Baseline scenario S007, Data Collection Mechanism.

8.7.3 Description

The equipment sends to the host EVENT REPORT as the data collection report when an event occurs. Event and data must have been enabled by the EVENT SETUP transaction.

By default, the format of the returned data is in the format returned by the equipment. It is valid for the implementation of a driver to modify the format of the returned data.

Data linked to the event report will be reported as a series of user defined data items. It is possible to enable an event without specifying any data to be reported. If there is no data “linked” to the event, then a "null" report is returned. That is, the message will not contain any user defined data items.

If EVENT SETUP specified a directory, then the event’s data will be stored in a file in that directory. The name of the file will be the event’s name, and the file extension will be a unique extension. The format of the data in the file will be the same format described in Appendix B. The machine driver must have the necessary permission to access the file system.

8.7.4 Examples

CMD/A="EVENT_REPORT" MID/A="STP1" MTY/A="E" ECD/U4=0 ETX/A="" EVENT_ID/A="MBC_START" TIME/A="1992021115010000"

CMD/A="EVENT_REPORT" MID/A="STP1" MTY/A="E" ECD/U4=0 ETX/A="" EVENT_ID/A="MB_COMPLETE" TIME/A="1992021115010000"

CMD/A="EVENT_REPORT" MID/A="STP1" MTY/A="E" ECD/U4=0 ETX/A="" EVENT_ID/A="MB_COMPLETE" FILE/A="\DIR\MB_COMPLETE.001"

CMD/A="EVENT_REPORT" MID/A="STP1" MTY/A="E" ECD/U4=30013 ETX/A=" VFEI_UNEXPECTED_EVENT"

8.7.5 Error Codes

VFEI_UNEXPECTED_EVENT
Event received that was not expected. There is not enough knowledge available to properly send information to the cell application.

8.8 EVENT SETUP

Enable/Disable Data Collection by Event (H->E)

Data Item	Type	Value
COMMAND_ID	ASCII	EVENT_SETUP
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C
TID	UNSIGNED 4	<transaction_id>
ENABLE	UNSIGNED 1	0 | 1
[EVENT_ID]*	ASCII	<value>
[VARIABLE_ID]*	ASCII	<value>
[DIR]	ASCII	<value>
[<user defined name>]*	<format>	<value>

Message Type: Command Reply Message: COMMAND ACK

8.8.1 Predefined Data Items

ENABLE Enable flag: 0 - DISABLE
1 - ENABLE

EVENT_ID Event ID

VARIABLE_ID Variable to report when event occurs

DIR Directory specification for file containing variable data associated with the event

8.8.2 Baseline Reference

This command is documented in Baseline scenario S245, Enable Data Collection.

8.8.3 Description

EVENT SETUP enables or disables data collection associated with an event. When event reporting is enabled, an EVENT REPORT message is sent when the event occurs. VARIABLE IDs specified for the event are returned in the EVENT REPORT message.

*****OBSOLETE***.** There is no longer any correlation between the setting of the NOEVENT optional parameter used in the INITIALIZE command and its effect on EVENT SETUP.

This event-based approach to data collection provides automatic notification to cell controller applications of equipment activities and is useful in monitoring the equipment and in maintaining synchronization with the equipment.

The equipment manufacturer is responsible for defining events that can collect data. Refer to Section 5 for a list of events required by VFEI in support of the SCC baselines. Note that VFEI

uses the event designation as specified in the section. The driver implementation makes any necessary equipment-specific translation.

The equipment manufacturer is required to provide unique IDs for the various variable data that can be collected in the equipment. Refer to Section 6 for a list of common Variable IDs. Note that VFEI uses the variable designation as specified in this section. The driver implementation makes any necessary equipment-specific translation.

ENABLE set to DISABLE (0) results in disabling reporting of specified event(s). If EVENT_ID is not specified, then reporting of all events is disabled.

When the ENABLE field is enabled, variables that are in the EVENT SETUP message are the variables reported in the EVENT REPORT message next time the event occurs. Specifying multiple EVENT_IDs in the EVENT SETUP message results in EVENT REPORTs with the same set of variables (if specified).

EVENT REPORT message reports only those variables enabled in the most recent EVENT SETUP message. That is, if an event had been enabled with one set of variables and the cell sent the EVENT SETUP message with another set of variables, then only those variables in the EVENT SETUP message are reported when an event next occurs. Enabling an event without specifying variables will result in a null report.

If the ENABLE flag is set to ENABLE (1) and no EVENT_ID is specified, then a VFEI_MISSING_PARAM error will be returned. For example,

CMD/A=“EVENT_SETUP” MID/A=“STP1” MTY/A=“C” TID/U4=5 ENABLE/U1=1 will result in the following returned message:

CMD/A=“CMD_ACK” MID/A=“STP1” MTY/A=“R” TID/U4=5 ECD/U4=30014 ETX/A=“VFEI_MISSING_PARAM”

If any of the specified events in the EVENT SETUP command do not exist then no action is taken with respect to the EVENT REPORT and the COMMAND ACK message will contain the error code EVENT_NOT_EXISTS.

*****OBSOLETE***.** There is no longer the capability to selectively disable reporting of any variables previously enabled.

If any of the specified variables in the EVENT SETUP command do not exist, then no action is taken with respect to the EVENT REPORT and the COMMAND ACK message will contain the error code EQUIP_INVALID_PARAM.

EVENT SETUP can specify that the data associated with an event be put into a file by giving a directory specification in the DIR field. The machine driver must have the necessary permission to create this file. The file name will consist of the event name with a unique file extension. The cell controller application is responsible for any cleanup of old files in the directory specified in the DIR field.

The data in the file will contain annotated fields but will be implementation-specific on how the annotation is done (see Appendix B).

The user-defined data items give the EVENT SETUP message any equipment-specific capability.

8.8.4 Examples

CMD/A="EVENT_SETUP" MID/A="STP1" MTY/A="C" TID/U4=5 ENABLE/U1=1 EVENT_ID/A="MB_START" VAR_ID/A="CLOCK"

CMD/A="EVENT_SETUP" MID/A="STP1" MTY/A="C" TID/U4=6 ENABLE/U1=0

CMD/A="EVENT_SETUP" MID/A="STP1" MTY/A="C" TID/U4=6 ENABLE/U1=1 EVENT_ID/A[2]=[“MB_START” “MBC_START”]

CMD/A="EVENT_SETUP" MID/A="STP1" MTY/A="C" TID/U4=6 ENABLE/U1=0 EVENT_ID/A="MB_START"

CMD/A="EVENT_SETUP" MID/A="STP1" MTY/A="C" TID/U4=6 ENABLE/U1=1 EVENT_ID/A="MB_START" VAR_ID/A="CLOCK" DIR/A="\usr\stp1\temp"

8.8.5 Reply

The equipment returns a COMMAND ACK message for the EVENT SETUP. This return indicates whether the EVENT SETUP command was accepted or rejected.

COMMAND ACK returns the following error codes:

EQUIP_EVENT_NOT_EXISTS
Event specified does not exist on the equipment.

EQUIP_INVALID_PARAM
Variable specified does not exist on the equipment.

EQUIP_MISSING_PARAM
Parameter required to perform the command is missing.

8.9 INITIALIZE

Initialize Logical Communications (H->E)

Data Item	Type	Value
COMMAND_ID	ASCII	INITIALIZE
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C
TID	UNSIGNED 4	<transaction_id>
[NOEVENT]	UNSIGNED 1	0 | 1 *****OBSOLETE*****
[<user defined name>]*	<format>	<value>

Message Type: Command Reply Message: COMMAND ACK

8.9.1 Predefined Data Items

NOEVENT *****OBSOLETE***** No event setup 0= NOT SET

1 = SET

8.9.2 Baseline Reference

This command is documented in Baseline scenario S250, Machine Initialize.

8.9.3 Description

INITIALIZE performs a logical communication connection with the equipment. This message could be issued as the cell's request to initialize logical communication or as a response to a RESTART, which is the equipment request to initialize logical communication.

NOEVENT is now considered obsolete. Previously, if the NOEVENT parameter was not present or was present and had a value of 0, then the following events specified by the Baseline Requirements (S250) were enabled for event data collection.

MBC_START	MBC_COMPLETE
MB_START	MB_COMPLETE
ABORT	PAUSE
RESUME	MESSAGE_RECEIVE
RESOURCE_ACTIVATE	

The enabling of these events should be achieved by following the INITIALIZE command with an EVENT SETUP and specifying these nine events in the [EVENT_ID]* data item.

INITIALIZE does not enable any events for data collection. Cell applications must use EVENT SETUP to enable data collection for all events.

INITIALIZE performs the following tasks in the order indicated:

1. Perform a logical communication connection with the equipment and verify software version and model. Implementation business rules will determine what to do if expected version or model are not received.
2. Request that the equipment go ONLINE.
3. Synchronize the equipment's clock with the machine driver's clock.
4. Set equipment constants. This is limited to the equipment constants that control the equipment driver's behavior and should not be changed by the cell controller.
5. Disables all event reporting and deletes all report definitions.

Alarms are not enabled or disabled. The host must accept any alarm it gets until it sends down the appropriate ALARM SETUP messages. Alarms are treated differently than events because of their capability to alert situations that may endanger people, equipment, or material.

8.9.4 Examples

CMD/A="INITIALIZE" MID/A="TRK1" MTY/A="C" TID/U4=7

8.9.5 Reply

The equipment returns a COMMAND ACK message for the INITIALIZE command. This return indicates whether the INITIALIZE command was accepted or rejected.

COMMAND ACK returns the following error codes:

EQUIP_CANNOT_DO_NOW
The equipment cannot perform the initialize at the current time.

EQUIP_INVALID_PARAM
Invalid parameter specified with the initialize or in an attempt to set up equipment constants.

EQUIP_INVALID_FORMAT
Error in deleting reports or linking reports.

EQUIP_UNKNOWN_ERROR
For any unexpected responses.

8.10 MACHINE COMMAND

Remote Equipment Command (H->E)

Data Item	Type	Value
COMMAND_ID	ASCII	MACH_CMD
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C
TID	UNSIGNED 4	<transaction_id>
COMMAND_TYPE	ASCII	<value>
[<user defined name>]*	<format>	<value>

Message Type: Command Reply Message: COMMAND ACK

8.10.1 Predefined Data Items

COMMAND_TYPE Equipment-specific command type.

8.10.2 Baseline Reference

This command has no Baseline reference.

8.10.3 Description

MACHINE COMMAND provides the cell applications with a level of control over equipment operations beyond the remote control described in the Baseline Requirements. This allows cell applications to have additional capability, such as performing material movement.

Do not use MACHINE COMMAND to perform those VFEI commands that are explicitly defined (e.g., ABORT, PAUSE, RESUME, START). MACHINE COMMAND is intended to support only those additional remote commands the equipment manufacturer has made available.

Command parameters are not specified in this document but are left to equipment manufacturers to define. User defined data items provide MACHINE COMMAND the capability for command parameters.

The equipment may interpret remote commands as “request action be initiated” rather than “do action.” In this case, the equipment would respond with the acknowledgment as the command “is going to be performed.” Completion of the action initiated by the machine command is usually indicated by the equipment triggering an event. Cell applications have the responsibility to send an EVENT SETUP transaction to have the event reported.

8.10.4 Examples

CMD/A="MACH_CMD" MID/A="TRK1" MTY/A="C" TID/U4=2 CMD_TYPE/A="LOOPBACK"

CMD/A="MACH_CMD" MID/A="TRK1" MTY/A="C" TID/U4=8 CMD_TYPE/A="HELLO"

8.10.5 Reply

The equipment returns a COMMAND ACK message for the MACHINE COMMAND command. This return indicates whether the MACHINE COMMAND command was accepted or rejected.

COMMAND ACK returns the following error codes:

EQUIP_INVALID_COMMAND
The equipment cannot perform the command requested.

EQUIP_CANNOT_DO_NOW
The equipment cannot do the request command at this time.

EQUIP_INVALID_PARAM
Invalid parameter specified with the requested command.

EQUIP_INVALID_C_STATE
The equipment is in an invalid control state for accepting the requested command. The equipment must be in a control state that accepts the remote command request.

EQUIP_INVALID_P_STATE
The equipment is in an invalid process state. The current process state does not accept the requested command.

8.11 MOVE COMMAND

Request Transfer of a Transfer Group (H->E)

Data Item	Type	Value
COMMAND_ID	ASCII	MOVE
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C
TID	UNSIGNED 4	<transaction_id>
[TG_TYPE]	ASCII	<value>
[TG_ID]	ASCII	<value>
[SRC_LOCATION]	ASCII	<value>
[DEST_LOCATION]	ASCII	<value>
[SRC_PORT]	ASCII	<value>
[DEST_PORT]	ASCII	<value>
[<user defined name>]*	<format>	<value>

Message Type: Command Reply Message: COMMAND ACK

8.11.1 Predefined Data Items

TG_TYPE	Transport Group Type
TG_ID	Transport Group ID
SRC_LOCATION	Source location
DEST_LOCATION	Destination location
SRC_PORT	Source port
DEST_PORT	Destination port

8.11.2 Baseline Reference

This command will be documented in the supplemental Baseline scenarios S227, Storage and S029, Transfer.

8.11.3 Description

MOVE commands the equipment to transfer material. Material may be transferred to/from either a location and/or port. Ports and Locations are discussed in SEMI E32-94 Material Movement Management Section 3.2. Specific implementations may require specification of any combination of SRC_LOCATION, DEST_LOCATION, SRC_PORT, and DEST_PORT.

TG_TYPE is currently unspecified. The VFEI Task Force may define valid TG_TYPE's in future releases.

The user defined data items give the MOVE message any equipment specific capability.

8.11.4 Examples

CMD/A="MOVE" MID/A="STOCKER1" MTY/A="C" TID/U4=2 TG_TYPE/A=“POD” TG_ID/A=“POD1” SRC_LOCATION/A=“LOC123” DEST_LOCATION/A=“STOCKER2”

CMD/A="MOVE" MID/A="MCS1" MTY/A="C" TID/U4=2 TG_TYPE/A=“” TG_ID/A=“12345-00000” SRC_LOCATION/A=“STOCKER1” DEST_LOCATION/A=“STOCKER2” SRC_PORT/A=“INPUT-PORT” DEST_PORT/A=“OUTPUT-PORT”

CMD/A="MOVE" MID/A="TRACK" MTY/A="C" TID/U4=2 TG_TYPE/A=“CAR” TG_ID/A=“CAR1”

8.11.5 Reply

The equipment returns a COMMAND ACK message for the MOVE COMMAND command. This return indicates whether the MOVE COMMAND command was accepted or rejected. When the equipment completes the move, an EVENT REPORT with an EVENT_ID of TG_ARRIVE will be generated if the event had been previously enabled.

COMMAND ACK returns the following error codes:

EQUIP_INVALID_COMMAND
The equipment cannot perform the command requested.

EQUIP_CANNOT_DO_NOW
The equipment cannot do the request command at this time.

EQUIP_INVALID_PARAM
Invalid parameter specified with the requested command.

8.12 PAUSE

Pause Processing (H->E)

Data Item	Type	Value
COMMAND_ID	ASCII	PAUSE
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C
TID	UNSIGNED 4	<transaction_id>
[<user defined name>]*	<format>	<value>

Message Type: Command Reply Message: COMMAND ACK

8.12.1 Predefined Data Items

None

8.12.2 Baseline Reference

This command is documented in Baseline scenario S251 Pause Machine Run.

8.12.3 Description

PAUSE suspends the current activity prior to its completion. Pause intends to suspend the process at the next "safe break point." Normal processing resumes with the RESUME command.

The user defined data items give the PAUSE command equipment-specific capability.

8.12.4 Examples

CMD/A="PAUSE" MID/A="TRK1" MTY/A="C" TID/U4=2

CMD/A="PAUSE" MID/A="TRK1" MTY/A="C" TID/U4=2
STATION/U1=1

8.12.5 Reply

The equipment returns a COMMAND ACK message for the PAUSE command. This return indicates whether the PAUSE command was accepted or rejected. When the equipment's current activity has been suspended, the equipment sends an EVENT REPORT with an EVENT_ID of PAUSE.

COMMAND ACK returns the following error codes:

EQUIP_INVALID_COMMAND
PAUSE not valid for this equipment

EQUIP_CANNOT_DO_NOW
PAUSE cannot be performed at this time.

EQUIP_INVALID_PARAM
Invalid parameter specified with the PAUSE.

EQUIP_INVALID_C_STATE
The equipment is in an invalid control state for accepting PAUSE. The equipment must be in a control state that accepts the remote PAUSE command.

EQUIP_INVALID_P_STATE
The equipment is in an invalid process state for accepting PAUSE. The current process state does not accept a PAUSE command.

8.13 RESOURCE ACTIVATE

Resource Activate (H->E)

Data Item	Type	Value
COMMAND_ID	ASCII	RES_ACTIVATE
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C
TID	UNSIGNED 4	<transaction_id>
TYPE	ASCII	RECIPE | TOOL | COMPONENT | CONSUMABLE | CASSETTE_MAP | WAFER_MAP
RESOURCE_ID+	ASCII	<value>
[<user defined name>]*	<format>	<value>

Message Type: Command Reply Message: COMMAND ACK

8.13.1 Predefined Data Items

TYPE Resource type RECIPE
TOOL (e.g., reticles, jigs, fixtures)
COMPONENT (e.g., manufactured goods - wafer)
CONSUMABLE (e.g., process gases, developer)
CASSETTE_MAP
WAFER_MAP

RESOURCE_ID Resource ID

8.13.2 Baseline Reference

This command is documented in Baseline scenario, S193 Recipe Setup.

8.13.3 Description

RESOURCE ACTIVATE activates resources while setting up equipment for a processing run. RESOURCE_IDs specifies which resources to activate.

RESOURCE ACTIVATE handles all resource types in the setup: recipes, tools, components, consumables, cassette maps, and wafer maps. Each resource type to be set up requires a separate RESOURCE ACTIVATE.

The user defined data items give a RESOURCE ACTIVATE message any equipment-specific capability.

8.13.4 Examples

CMD/A="RES_ACTIVATE" MID/A="TRK1" MTY/A="C" TID/U4=10 TYPE/A="RECIPE" RES_ID/A="110232"

8.13.5 Reply

The equipment returns a COMMAND ACK message for the RESOURCE ACTIVATE command. This return indicates whether the RESOURCE ACTIVATE command was accepted or rejected. When the equipment's resource has been activated, the equipment sends an EVENT REPORT with an event type of RESOURCE_ACTIVATE.

COMMAND ACK returns the following error codes:

EQUIP_INVALID_TYPE
Type specified is not valid for the equipment.

EQUIP_INVALID_COMMAND
The equipment cannot perform the resource activation.

EQUIP_CANNOT_DO_NOW
Resource activation cannot be performed at this time.

EQUIP_INVALID_PARAM
Invalid parameter specified with the resource activation.

EQUIP_INVALID_C_STATE
The equipment is in an invalid control state for accepting the resource activation. The equipment must be in a control state that accepts the remote resource activation command.

EQUIP_INVALID_P_STATE
The equipment is in an invalid process state for accepting the resource activation. The current process state does not accept a resource activation command.

8.14 RESOURCE DEACTIVATE

Resource Deactivate (H->E)

Data Item	Type	Value
COMMAND_ID	ASCII	RES_DEACTIVATE
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C
TID	UNSIGNED 4	<transaction_id>
TYPE	ASCII	RECIPE | TOOL | COMPONENT | CONSUMABLE | CASSETTE_MAP | WAFER_MAP
RESOURCE_ID+	ASCII	<value>
[<user defined name>]*	<format>	<value>

Message Type: Command Reply Message: COMMAND ACK

8.14.1 Predefined Data Items

TYPE Resource type RECIPE
TOOL (e.g., reticles, jigs, fixtures)
COMPONENT (e.g., manufactured goods - wafer)
CONSUMABLE (e.g., process gases, developer)
CASSETTE_MAP
WAFER_MAP

RESOURCE_ID Resource ID

8.14.2 Baseline Reference

This command is documented in Baseline scenario, S237 Recipe Deactivate.

8.14.3 Description

RESOURCE DEACTIVATE deactivates resources for shutting down equipment after a processing run. Resources in the parameter list specify which resources to deactivate.

RESOURCE DEACTIVATE handles all resource types in the shutdown: recipes, tools, components, consumables, cassette maps, and wafer maps. Each resource type to be shutdown requires a separate RESOURCE DEACTIVATE message.

The user defined data items give RESOURCE DEACTIVATE message any equipment-specific capability.

8.14.4 Examples

CMD/A="RES_DEACTIVATE" MID/A="TRK1" MTY/A="C" TID/U4=10
TYPE/A="RECIPE" RES_ID/A="110232"

8.14.5 Reply

The equipment returns a COMMAND ACK message for the RESOURCE DEACTIVATE command. This return indicates whether the RESOURCE DEACTIVATE command was accepted or rejected.

COMMAND ACK returns the following error codes:

EQUIP_INVALID_TYPE
Type specified is not valid for the equipment.

EQUIP_INVALID_COMMAND
The equipment cannot perform the resource deactivate.

EQUIP_CANNOT_DO_NOW
Resource deactivate cannot be performed at this time.

EQUIP_INVALID_PARAM
Invalid parameter specified with the resource deactivate.

EQUIP_INVALID_C_STATE
The equipment is in an invalid control state for accepting the resource deactivate. The equipment must be in a control state that accepts the remote resource deactivate command.

EQUIP_INVALID_P_STATE
The equipment is in an invalid process state for accepting the resource deactivate. The current process state does not accept a resource deactivate command.

8.15 RESOURCE DELETE

Resource Delete (H->E)

Data Item	Type	Value
COMMAND_ID	ASCII	RES_DELETE
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C
TID	UNSIGNED 4	<transaction_id>
TYPE	ASCII	RECIPE | TOOL | COMPONENT | CONSUMABLE | CASSETTE_MAP | WAFER_MAP
RESOURCE_ID+	ASCII	<value>
[<user defined name>]*	<format>	<value>

Message Type: Command Reply Message: COMMAND ACK

8.15.1 Predefined Data Items

TYPE	Resource type	RECIPE TOOL (e.g., reticles, jigs, fixtures) COMPONENT (e.g., manufactured goods - wafer) CONSUMABLE (e.g., process gases, developer) CASSETTE_MAP WAFER_MAP
RESOURCE_ID	Resource ID	

8.15.2 Baseline Reference

This command is documented in Baseline scenario S282 Delete Specification.

8.15.3 Description

RESOURCE DELETE deletes resources from the equipment. RESOURCE_IDs specify which resources to delete.

RESOURCE DELETE handles all resource types: recipes, tools, components, consumables, cassette maps, and wafer maps. Each resource type requires a separate RESOURCE DELETE message.

The user defined data items give the RESOURCE DELETE message any equipment-specific capability.

8.15.4 Examples

CMD/A="RES_DELETE" MID/A="TRK1" MTY/A="C" TID/U4=10 TYPE/A="RECIPE" RES_ID/A="110232"

8.15.5 Reply

The equipment returns a COMMAND ACK message for the RESOURCE DELETE command. This return indicates whether the RESOURCE DELETE command was accepted or rejected.

COMMAND ACK returns the following error codes:

EQUIP_INVALID_TYPE
Type specified is not valid for the equipment.

EQUIP_INVALID_COMMAND
The equipment cannot perform the resource delete.

EQUIP_CANNOT_DO_NOW
Resource delete cannot be performed at this time.

EQUIP_INVALID_PARAM
Invalid parameter specified with the resource delete.

EQUIP_INVALID_C_STATE
The equipment is in an invalid control state for accepting the resource delete. The equipment must be in a control state that accepts the remote resource delete command.

EQUIP_INVALID_P_STATE
The equipment is in an invalid process state for accepting the resource delete. The current process state does not accept a resource delete command.

8.15.6 RESOURCE LIST

Resource List (E->H)

Data Item	Type	Value
COMMAND_ID	ASCII	RES_LIST
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	R
TID	UNSIGNED 4	<value>
ERROR_CODE	UNSIGNED 4	<error_code>
ERROR_TEXT	ASCII	<error_text>
TYPE	ASCII	RECIPE | TOOL | COMPONENT | CONSUMABLE | CASSETTE_MAP | WAFER_MAP
[RESOURCE_ID]*	ASCII	<value>
[FILE]	ASCII	<value>

Message Type: Reply

8.15.7 Predefined Data Items

TYPE	Resource type	RECIPE TOOL (e.g., reticles, jigs, fixtures) COMPONENT (e.g., manufactured goods - wafer) CONSUMABLE (e.g., process gases, developer) CASSETTE_MAP WAFER_MAP
RESOURCE_ID	Resource ID	
FILE	Full file specification containing resource list	

8.15.8 Baseline Reference

This command is documented in Baseline scenario, S193 Recipe Setup.

8.15.9 Description

RESOURCE LIST is the reply message to RESOURCE QUERY. RESOURCE QUERY requests existence of resources on the equipment. RESOURCE LIST replies with the resources that exist on the equipment.

RESOURCE QUERY either sends a resource ID list or no resource IDs. If RESOURCE QUERY contains a resource ID list, then RESOURCE LIST contains a resource ID list of all resources that are in the RESOURCE QUERY resource ID list that exist on the equipment. If RESOURCE QUERY contains no resource IDs, then RESOURCE LIST contains a resource ID list of all resources that exist on the equipment.

If the RESOURCE QUERY specifies a file, then the resource ID list is returned in the file. The machine driver will override any information that is in the file and put in the returned resource IDs. The machine driver must have the necessary permission to access to the file.

8.15.10 Examples

CMD/A="RES_LIST" MID/A="STP1" MTY/A="R" TID/U4=14 ECD/U4=0 ETX/A="" TYPE/A="RECIPE" RES_ID/A[2]=["RECIPE1" "RECIPE2"]

CMD/A="RES_LIST" MID/A="STP1" MTY/A="R" TID/U4=15 ECD/U4=50020 ETX/A="Not all resource IDs exist" TYPE/A="RECIPE" RES_ID/A="RECIPE1"

8.15.11 Error Codes

EQUIP_INCOMPLETE_RESPONSE
All of the data requested is not available.

EQUIP_INVALID_TYPE
Type specified is not valid for the equipment.

8.16 RESOURCE QUERY

Resource Query (H->E)

Data Item	Type	Value
COMMAND_ID	ASCII	RES_QUERY
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C
TID	UNSIGNED 4	<transaction_id>
TYPE	ASCII	RECIPE | TOOL | COMPONENT | CONSUMABLE | CASSETTE_MAP |WAFER_MAP
[RESOURCE_ID]*	ASCII	<value>
[FILE]	ASCII	<value>
[<user defined name>]*	<format>	<value>

Message Type: Command Reply Message: RESOURCE LIST

8.16.1 Predefined Data Items

TYPE	Resource type	RECIPE TOOL (e.g., reticles, jigs, fixtures) COMPONENT (e.g., manufactured goods - wafer) CONSUMABLE (e.g., process gases, developer) CASSETTE_MAP WAFER_MAP
RESOURCE_ID	Resource ID	
FILE	Full file specification to put resource list	

8.16.2 Baseline Reference

This command is documented in Baseline scenario, S193 Recipe Setup.

8.16.3 Description

RESOURCE QUERY queries equipment for existence of resources within a given resource type. RESOURCE QUERY requests a search criteria based on the resource IDs that are in the command. If RESOURCE QUERY contains no resource IDs, then RESOURCE QUERY requests all resource IDs. If RESOURCE QUERY contains a resource ID list, then RESOURCE QUERY requests resources in this list that are on the equipment.

RESOURCE QUERY can request that the list of resources be stored in a file by giving a file specification in the FILE field. The machine driver must have the necessary permission to access this file.

The user defined data items give the RESOURCE QUERY message any equipment-specific capability.

8.16.4 Examples

CMD/A="RES_QUERY" MID/A="STP1" MTY/A="C" TID/U4=12 TYPE/A="RECIPE"

CMD/A="RES_QUERY" MID/A="STP1" MTY/A="C" TID/U4=12 TYPE/A="RECIPE" RES_ID/A[2]=["7892" "RECIPE1"]

8.16.5 Reply

RESOURCE LIST is the reply message that returns a set of RESOURCE_IDs for resources that exist on the equipment.

8.17 RESOURCE REQUEST

Resource Transfer Request (H<->E)

Data Item	Type	Value
COMMAND_ID	ASCII	RES_REQUEST
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C
TID	UNSIGNED 4	<transaction_id>
TYPE	ASCII	RECIPE | TOOL | COMPONENT | CONSUMABLE | CASSETTE_MAP | WAFER_MAP
RESOURCE_ID	ASCII	<value>
[RESOURCE_FILE]	ASCII	<value>
[<user defined name>]*	<format>	<value>

Message Type: Command Reply Message: RESOURCE TRANSFER

8.17.1 Predefined Data Items

TYPE Resource type RECIPE
TOOL (e.g., reticles, jigs, fixtures)
COMPONENT (e.g., manufactured goods - wafer)
CONSUMABLE (e.g., process gases, developer)
CASSETTE_MAP
WAFER_MAP

RESOURCE_ID Resource ID to transfer

RESOURCE_FILE Full file specification for file that will contain the resource

8.17.2 Baseline Reference

This command is documented in Baseline scenario S211, Recipe Transfer.

8.17.3 Description

RESOURCE REQUEST requests a transfer of a resource. The sender is requesting that the receiver transfer the resource to the sender. The equipment or cell application can be the sender of the command.

RESOURCE REQUEST can request that the resource be stored in a file by giving a file specification in the RESOURCE_FILE field. The driver must have the necessary permission to access this file.

If RESOURCE REQUEST does not specify a file, then the resource will be returned within the reply message.

The user defined data items give RESOURCE REQUEST any equipment-specific capability.

8.17.4 Examples

CMD/A="RES_REQUEST" MID/A="STP1" MTY/A="C" TID/U4=11 TYPE/A="RECIPE" RES_ID/A="RECIPE1"

CMD/A="RES_REQUEST" MID/A="STP1" MTY/A="C" TID/U4=12 TYPE/A="RECIPE" RES_ID/A="BIG_RECIPE" RES_FILE/A="/usr/recipes/big_recipe.dat"

8.17.5 Reply

RESOURCE TRANSFER is the reply message that contains the resource from the equipment.

8.18 RESOURCE TRANSFER

Resource Transfer (H<->E)

Data Item	Type	Value
COMMAND_ID	ASCII	RES_TRANSFER
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C
TID	UNSIGNED 4	<transaction_id>
TYPE	ASCII	RECIPE | TOOL | COMPONENT | CONSUMABLE | CASSETTE_MAP | WAFER_MAP
RESOURCE_ID	ASCII	<value>
RESOURCE_LENGTH	UNSIGNED 4	<value>
RESOURCE_BODY	<format>	<value>
[<user defined name>]*	<format>	<value>

Data Item	Type	Value
COMMAND_ID	ASCII	RES_TRANSFER
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C
TID	UNSIGNED 4	<transaction_id>
TYPE	ASCII	RECIPE | TOOL | COMPONENT | CONSUMABLE | CASSETTE_MAP | WAFER_MAP
RESOURCE_ID	ASCII	<value>
RESOURCE_FILE	ASCII	<value>
[<user defined name>]*	<format>	<value>

Data Item	Type	Value
COMMAND_ID	ASCII	RES_TRANSFER
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C
TID	UNSIGNED 4	<transaction_id>
TYPE	ASCII	RECIPE | TOOL | COMPONENT | CONSUMABLE | CASSETTE_MAP | WAFER_MAP
RESOURCE_ID	ASCII	<value>
SOURCE	ASCII	<value>
DESTINATION	ASCII	<value>
[<user defined name>]*	<format>	<value>

Message Type: Command Reply Message: COMMAND ACK

Data Item	Type	Value
COMMAND_ID	ASCII	RES_TRANSFER
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	R
TID	UNSIGNED 4	<transaction_id>
ERROR_CODE	UNSIGNED 4	<error_code>
ERROR_TEXT	ASCII	<error_text>
TYPE	ASCII	RECIPE | TOOL | COMPONENT | CONSUMABLE | CASSETE_MAP | WAFER_MAP
RESOURCE_ID	ASCII	<value>
RESOURCE_LENGTH	UNSIGNED 4	<value>
RESOURCE_BODY	<format>	<value>
[<user defined name>]*	<format>	<value>

Data Item	Type	Value
COMMAND_ID	ASCII	RES_TRANSFER
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	R
TID	UNSIGNED 4	<transaction_id>
ERROR_CODE	UNSIGNED 4	<error_code>
ERROR_TEXT	ASCII	<error_text>
TYPE	ASCII	RECIPE | TOOL | COMPONENT | CONSUMABLE | CASSETTE_MAP | WAFER_MAP
RESOURCE_ID	ASCII	<value>
RESOURCE_FILE	ASCII	<value>
[<user defined name>]*	<format>	<value>

Message Type: Reply

8.18.1 Predefined Data Items

TYPE Resource type RECIPE
TOOL (e.g., reticles, jigs, fixtures)
COMPONENT (e.g., manufactured goods - wafer)
CONSUMABLE (e.g., process gases, developer)
CASSETTE_MAP
WAFER_MAP

RESOURCE_ID Resource ID to download

RESOURCE_LENGTH Resource length in resource body

RESOURCE_BODY Resource to download

RESOURCE_FILE Full file specification for file that contains the resource

SOURCE Source of transfer

DESTINATION Destination of transfer

8.18.2 Baseline Reference

This command is documented in Baseline scenario S193 Recipe Setup.

8.18.3 Description

RESOURCE TRANSFER transfers a resource between the cell application and the equipment. RESOURCE TRANSFER can be either a command or a reply.

If RESOURCE TRANSFER is a command, then the sender is transferring the resource to the receiver. The equipment or cell application can be the sender of the command. The equipment is responsible for verifying and validating all resources.

If RESOURCE TRANSFER is a reply, which is in response to a RESOURCE REQUEST message, then the resource is transferred from the receiver of the RESOURCE REQUEST message to the sender of the RESOURCE REQUEST message. The sender of the RESOURCE REQUEST message can be the cell application or the equipment.

This command handles all resource types to be transferred: recipes, tools, components, consumables, cassette maps, and wafer maps. Each resource type transferred requires a separate RESOURCE TRANSFER command.

The resource transferred can be specified either in RESOURCE_BODY or RESOURCE_FILE. If specified in RESOURCE_BODY, then RESOURCE_BODY contains the resource and RESOURCE_LENGTH contains the length. If specified in RESOURCE_FILE, then the file specified contains the resource. The equipment communications must have the permission necessary to read the transfer and the resource to the equipment and to write the transfer from the resource to the cell application. The cell application is responsible for any cleanup of the file after the transfer.

The parameters SOURCE and DESTINATION give the RESOURCE TRANSFER command the ability to allow a sender to request transfer of the resource within the domain of the receiver. For example, some equipment will require a resource (e.g., recipe) be transferred from the equipments' storage location/SOURCE to a processing/DESTINATION location before processing can be started.

The user defined data items give the RESOURCE TRANSFER command any equipment-specific capability.

8.18.4 Examples

CMD/A="RES_TRANSFER" MID/A="STP1" MTY/A="C" TID/U4=11
TYPE/A="RECIPE" RES_ID/A="RECIPE1" RES_LENGTH/U4=9
RES_BODY/B[9]=[0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09]

CMD/A="RES_TRANSFER" MID/A="STP1" MTY/A="R" TID/U4=12 ECD/U4=0 ETX/A=""
TYPE/A="RECIPE" RES_ID/A="BIG_RECIPE" RES_FILE/A="/usr/recipes/big_recipe.dat"

8.18.5 Reply

COMMAND ACK message is the reply for the RESOURCE TRANSFER command. This return indicates whether the RESOURCE TRANSFER command was accepted or rejected.

COMMAND ACK returns the following error codes:

EQUIP_INVALID_TYPE
Type specified is not valid.

EQUIP_MISSING_PARAM
Parameter to perform the transfer is missing.

EQUIP_INVALID_PARAM
File specified could not be found.

EQUIP_ALREADY_HAVE
Resource already exists.

EQUIP_NOT_ENOUGH_SPACE
There is not enough space to receive the resource.

EQUIP_CANNOT_DO_NOW
Resource transfer cannot be performed at this time.

EQUIP_WILL_NOT_ACCEPT
Resource cannot be accepted.

RESOURCE TRANSFER reply returns the following error codes:

EQUIP_INVALID_TYPE
Type specified is not valid for the equipment.

EQUIP_REQUEST_DENIED
Resource transfer request has been denied

8.19 RESTART

Restart Logical Communications (E->H)

Data Item	Type	Value
COMMAND_ID	ASCII	RESTART
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	E
ERROR_CODE	UNSIGNED 4	<error_code>
ERROR_TEXT	ASCII	<error_text>
[<user defined name>]*	<format>	<value>

Message Type: Unsolicited

8.19.1 Predefined Data Items

None defined.

8.19.2 Baseline Reference

This message is documented in Baseline scenario, S250 Initialize.

8.19.3 Description

The equipment sends a RESTART to restart logical communications with the cell applications. This is not a low level connectivity issue, but rather a logical application issue used by the equipment to notify the cell applications that they may need to perform synchronization activities with the equipment. RESTART provides a means of formally establishing communications following power-up or loss of communications.

If the cell applications is ready to restart logical communications, then the cell applications respond with the INITIALIZE message.

8.19.4 Examples

CMD/A="RESTART" MID/A="STP1" MTY/A="E" ECD/U4=0 ETX/A=""

8.19.5 Error Codes

None defined.

8.20 RESUME

Resume Processing (H->E)

Data Item	Type	Value
COMMAND_ID	ASCII	RESUME
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C
TID	UNSIGNED 4	<transaction_id>
[<user defined name>]*	<format>	<value>

Message Type: Command Reply Message: COMMAND ACK

8.20.1 Predefined Data Items

None defined.

8.20.2 Baseline Reference

This command is documented in Baseline scenario S252, Resume Machine Run.

8.20.3 Description

RESUME continues the current activity that had been suspended prior to its completion. Resume intends to resume processing from a "safe break point." Normal processing reaches the safe break point with the PAUSE command.

The user defined data items give the RESUME command equipment-specific capability.

8.20.4 Examples

CMD/A="RESUME" MID/A="TRK1" MTY/A="C" TID/U4=2

CMD/A="RESUME" MID/A="TRK1" MTY/A="C" TID/U4=2 STATION/U1=1

8.20.5 Reply

The equipment returns a COMMAND ACK message for the RESUME command. This return indicates whether the RESUME command was accepted or rejected. When the equipment's current activity has been resumed, the equipment sends an EVENT REPORT with an event type of RESUME.

COMMAND ACK returns the following error codes:

EQUIP_INVALID_COMMAND
RESUME not valid for this equipment.

EQUIP_CANNOT_DO_NOW
RESUME cannot be performed at this time.

EQUIP_INVALID_PARAM
Invalid parameter specified with RESUME.

EQUIP_INVALID_C_STATE
The equipment is in an invalid control state for accepting RESUME. The equipment must be in a control state that accepts the remote RESUME command.

EQUIP_INVALID_P_STATE
The equipment is in an invalid process state for accepting RESUME. The current process state does not accept a RESUME command.

8.21 SET PARAMETER

Set Resource Parameters (H->E)

Data Item	Type	Value
COMMAND_ID	ASCII	SET_PARAM
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C
TID	UNSIGNED 4	<transaction_id>
TYPE	ASCII	RECIPE | TOOL | COMPONENT | CONSUMABLE | MACHINE | VFEI
[<user defined name>]*	<format>	<value>

Message Type: Command Reply Message: COMMAND ACK

8.21.1 Predefined Data Items

TYPE Resource type RECIPE
TOOL (e.g., reticles, jigs, fixtures)
COMPONENT (e.g., manufactured goods - wafer)
CONSUMABLE (e.g., process gases, developer)
MACHINE
VFEI

8.21.2 Baseline Reference

This command is documented in Baseline scenario, S224 Specification Parameter Revision.

8.21.3 Description

SET PARAMETER sets parameter values associated with different types of resources. The equipment is responsible for checking upper and lower bounds on the parameter values.

If the resource type is RECIPE, then SET PARAMETER changes parameters associated with the recipe in the executable space.

If the resource type is MACHINE, then SET PARAMETER sets equipment constants that may affect equipment behaviors.

If the resource type is VFEI, then SET PARAMETER sets parameters that affect or control the operation of the VFEI driver itself.

This command handles setting parameter values for all resource types. Each resource type requires a separate SET PARAMETER message.

8.21.4 Examples

CMD/A="SET_PARAM" MID/A="TRK1" MTY/A="C" TID/U4=21 TYPE/A="RECIPE" X_FOCUS/F4=0.5

CMD/A="SET_PARAM" MID/A="TRK1" MTY/A="C" TID/U4=21 TYPE/A="VFEI" DEBUG/U1=2 VERBOSE/U1=3

8.21.5 Reply

The equipment returns a COMMAND ACK message for the SET PARAMETER command. This return indicates whether the SET PARAMETER command was accepted or rejected.

COMMAND ACK returns the following error codes:

EQUIP_INVALID_TYPE
Type specified is not valid for the equipment.

EQUIP_INVALID_COMMAND
SET PARAMETER not valid for this equipment.

EQUIP_CANNOT_DO_NOW
SET PARAMETER cannot be performed at this time.

EQUIP_INVALID_PARAM
Invalid parameter specified with the SET PARAMETER.

EQUIP_INVALID_C_STATE
The equipment is in an invalid control state for accepting SET PARAMETER. The equipment must be in a control state that accepts the remote SET PARAMETER command.

EQUIP_INVALID_P_STATE
The equipment is in an invalid process state for accepting SET PARAMETER. The current process state does not accept a SET PARAMETER command.

8.22 START

Start Processing (H->E)

Data Item	Type	Value
COMMAND_ID	ASCII	START
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C
TID	UNSIGNED 4	<transaction_id>
[<user defined name>]*	<format>	<value>

Message Type: Command Reply Message: COMMAND ACK

8.22.1 Predefined Data Items

None defined.

8.22.2 Baseline Reference

This command is documented in Baseline scenario S202 Start Machine Batch.

8.22.3 Description

START commands the equipment to begin processing material. The cell application is responsible for ensuring that all setup activities have taken place before the START command is sent to the equipment. This includes doing any necessary resource download and resource activation.

The user defined data items give the START command equipment-specific capability within START.

8.22.4 Examples

CMD/A="START" MID/A="TRK1" MTY/A="C" TID/U4=2

CMD/A="START" MID/A="TRK1" MTY/A="C" TID/U4=2
INPUT_LOC/U1=2 MBC_SIZE/U1=0

8.22.5 Reply

The equipment returns a COMMAND ACK message for the START command. This return indicates whether the START command was accepted or rejected. When the equipment starts processing material, the equipment sends an EVENT REPORT with an EVENT_ID of MBC_START.

COMMAND ACK returns the following error codes:

EQUIP_INVALID_COMMAND
START not valid for this equipment.

EQUIP_CANNOT_DO_NOW
START cannot be performed at this time.

EQUIP_INVALID_PARAM
Invalid parameter specified with the START.

EQUIP_INVALID_C_STATE
The equipment is in an invalid control state for accepting START. The equipment must be in a control state that accepts the remote START command.

EQUIP_INVALID_P_STATE
The equipment is in an invalid process state for accepting START. The current process state does not accept a START command.

8.23 STATUS LIST

Status Variable List (E->H)

Data Item	Type	Value
COMMAND_ID	ASCII	STATUS_LIST
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	R
TID	UNSIGNED 4	<value>
ERROR_CODE	UNSIGNED 4	<error_code>
ERROR_TEXT	ASCII	<error_text>
[FILE]	ASCII	<value>
[<user defined name>]*	<format>	<value>

Message Type: Reply

8.23.1 Predefined Data Items

FILE Full file specification containing the variable values.

8.23.2 Baseline Reference

This command is documented in Baseline scenario S007 Data Collection Mechanism.

8.23.3 Description

STATUS LIST is the reply message to the STATUS QUERY command. STATUS QUERY requests the current status of specified equipment variables. STATUS LIST contains the current status of the valid specified variables. Variables requested by the STATUS QUERY command that are invalid will not be returned.

Variables that were requested in the STATUS QUERY command will be reported as a series of user defined data items. If any of the specified variables in the STATUS QUERY command did not exist, then STATUS LIST will report the values of all valid variables and return an error code of EQUIP_INCOMPLETE_RESPONSE.

8.23.4 Examples

CMD/A="STATUS_LIST" MID/A="STP1" MTY/A="R" TID/U4=1 ECD/U4=0 ETX/A="" PP_EXEC_NAME/A="RECIPE1"

CMD/A="STATUS_LIST" MID/A="STP1" MTY/A="R" TID/U4=1 ECD/U4=50020 ETX/A="Not all variable IDs were valid" MB_ID/A="WAFER1"

8.23.5 Error Codes

EQUIP_INCOMPLETE_RESPONSE
All of the data requested is not available.

8.24 STATUS QUERY

Query for Current Status (H->E)

Data Item	Type	Value
COMMAND_ID	ASCII	STATUS_QUERY
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C
TID	ASCII	<transaction_id>
VARIABLE_ID+	ASCII	<value>
[FILE]	ASCII	<value>
[<user defined name>]*	<format>	<value>

Message Type: Command Reply Message: STATUS LIST

8.24.1 Predefined Data Items

VARIABLE_ID Variable ID.

FILE Full file specification to put status variable values.

8.24.2 Baseline Reference

This command is documented in Baseline scenario S007, Data Collection Mechanism.

8.24.3 Description

STATUS QUERY allows cell controller applications to query for data within the equipment.

The equipment manufacturer is required to document unique IDs for the variable data that can be collected on the equipment. In addition, variables containing data about the VFEI driver is also available. Refer to Section 6 for a list of commonly provided IDs.

Note that VFEI uses the variable designation as specified in the section. The driver implementation makes any necessary equipment-specific translation.

If any of the specified variables in the STATUS QUERY command do not exist then the resulting STATUS LIST will report the values of all valid variables and return an error code of EQUIP_INCOMPLETE_RESPONSE.

STATUS QUERY can request that the list of status variables be stored in a file by giving a file specification in the FILE field. The machine driver must have the necessary permission to access this file.

User defined data items give STATUS QUERY any equipment-specific capability.

8.24.4 Examples

CMD/A="STATUS_QUERY" MID/A="TRK1" MTY/A="C" TID/U4=2
VAR_ID/A="PP_EXEC_NAME"

CMD/A="STATUS_QUERY" MID/A="TRK1" MTY/A="C" TID/U4=2
VAR_ID/A[2]=["MB_ID" "CLOCK"]

8.24.5 Reply

STATUS LIST message returns the data specified.

8.25 TRACE REPORT

Data Collection Based On Time (E->H)

Data Item	Type	Value
COMMAND_ID	ASCII	TRACE_REPORT
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	E
ERROR_CODE	UNSIGNED 4	<error_code>
ERROR_TEXT	ASCII	<error_text>
TRACE_ID	UNSIGNED 2	<value>
REPORT_SIZE	UNSIGNED 2	<value>
[FILE]	ASCII	<value>
[<user defined name>]*	<format>	<value>

Message Type: Unsolicited

8.25.1 Predefined Data Items

TRACE_ID Trace ID

REPORT_SIZE Number of collections taken

FILE Full file specification for file that contains the trace data.

8.25.2 Baseline Reference

This command is documented in Baseline scenario S007 Data Collection Mechanism.

8.25.3 Description

TRACE REPORT is a data collection report sent from the equipment as specified in a TRACE SETUP. This report is based on the periodic time interval and variables defined in the TRACE SETUP. The value of the data is in the format returned by the equipment.

Variables linked to the trace report by the TRACE SETUP command will be reported as a series of user defined data items. At least one variable will always be contained in TRACE REPORT. The TRACE SETUP command requires at least one variable. This variable must exist or the TRACE SETUP command is rejected and no TRACE REPORT is received.

If TRACE SETUP specified a directory, then the trace data will be stored in a file in the directory specified. The name of the file will be the TRACE_ID and the file extension will be a unique extension. The format of the data in the file will be in the format described in Appendix B. The machine driver must have the necessary permission to access the file system.

8.25.4 Examples

CMD/A="TRACE_REPORT" MID/A="STP1" MTY/A="E" ECD/U4=0 ETX/A="" TRACE_ID/U2=1 REPORT_SIZE/U2=1 TEMP/I2=120

CMD/A="TRACE_REPORT" MID/A="STP1" MTY/A="E" ECD/U4=0 ETX/A="" TRACE_ID/U2=2 REPORT_SIZE/U2=2 PRESSURE/U4[2]=[760 759]

8.25.5 Error Codes

EQUIP_INVALID_PARAM
Invalid trace ID received from the equipment.

EQUIP_INVALID_FORMAT
Number of variables received is not the expected number based on the number of collections taken and the number of expected variables per collection.

8.25.6 TRACE SETUP

Enable Data Collection on Time (H->E)

Data Item	Type	Value
COMMAND_ID	ASCII	TRACE_SETUP
MACHINE_ID	ASCII	<mach_id>
MSG_TYPE	ASCII	C
TID	UNSIGNED 4	<transaction_id>
ENABLE	UNSIGNED 1	<value>
TRACE_ID	UNSIGNED 2	<value>
FREQUENCY	UNSIGNED 4	<value>
SAMPLES	UNSIGNED 2	<value>
REPORT_SIZE	UNSIGNED 2	<value>
VARIABLE_ID+	ASCII	<value>
[DIR]	ASCII	<value>
[<user defined name>]*	<format>	<value>

Message Type: Command Reply Message: COMMAND ACK

8.25.7 Predefined Data Items

ENABLE	Enable flag: 0 - disable, 1 - enable
TRACE_ID	Trace ID
FREQUENCY	Sample frequency in seconds
SAMPLES	Total number of samples to take
REPORT_SIZE	Number of samples to report in one report
VARIABLE_ID	Variable ID to report
DIR	Directory specification for file containing trace data associated with the TRACE_ID

8.25.8 Baseline Reference

This command is documented in Baseline scenario S245 Enable Data Collection.

8.25.9 Description

TRACE SETUP enables/disables data collection to be performed based on a periodic time basis on equipment. For the report, the cell application designates the ID for the trace report, time interval for data sampling, total number of samples to be taken, samples per trace report, and listing of data that will be sent with the report. TRACE REPORT returns the data specified.

The equipment manufacturer is required to provide unique IDs for the various variable data that can be collected in the equipment. Refer to Section 6 for a list of IDs commonly provided by the equipment manufacturers.

The cell application may modify or re-initiate a trace function currently in progress by specifying the same trace ID in a TRACE SETUP definition. The old trace is terminated and the new trace is initiated. The cell application can instruct the equipment to terminate the trace report prior to its completion.

If any of the specified variables in the TRACE SETUP command do not exist, then no action is taken with respect to the TRACE REPORT and the COMMAND ACK message will contain the error code EQUIP_INVALID_PARAM.

TRACE SETUP can specify that the data associated with a trace be put into a file by giving a directory specification in the DIR field. The machine driver must have the necessary permission to create this file. The file name will consist of the TRACE_ID with a unique file extension. The cell controller application is responsible for any cleanup of old files in the directory specified in the DIR field.

User defined data items give TRACE SETUP any equipment-specific capability.

8.25.10 Examples

CMD/A="TRACE_SETUP" MID/A="TRK1" MTY/A="C" TID/U4=2 ENABLE/U1=1 TRACE_ID/U2=1 FREQUENCY/U4=60 SAMPLES/U2=120 REPORT_SIZE/U2=1 VAR_ID/A="TEMP"

CMD/A="TRACE_SETUP" MID/A="TRK1" MTY/A="C" TID/U4=2 ENABLE/U1=1 TRACE_ID/U2=1 FREQUENCY/U4=10 SAMPLES/U2=60 REPORT_SIZE/U2=2 VAR_ID/A="PRESSURE"

8.25.11 Reply

The equipment returns a COMMAND ACK message for the TRACE SETUP command. This return indicates whether the TRACE SETUP command was accepted or rejected.

COMMAND ACK returns the following error codes:

EQUIP_WILL_NOT_ACCEPT
Message contains invalid trace ID.

EQUIP_INVALID_PARAM
Message contains an invalid variable ID.

EQUIP_INVALID_FORMAT
Variable format is not the format with the trace ID.

EQUIP_NOT_ENOUGH_SPACE
The equipment does not contain enough space for another trace report.

EQUIP_CANNOT_DO_NOW
The equipment cannot perform the trace at this time.

APPENDIX A: GLOSSARY

Activity Flow Chart	**n.** A diagram describing the flow of scenarios in some context. For example, the flow of material in scenarios is described in an Activity Flow Chart. May be viewed as a PERT chart (dependency diagram).
Alarm	**n.** Abnormal situation that may endanger people, equipment, or material being processed [SEMI GEM 3.0].
Business Rule	**n.** A decision rule placed in the SCC1 "Cell Application" domain to describe variations in activities caused by procedural differences among sites. Business rules may change from site to site (e.g., When is preventative maintenance scheduled?)
Carrier Transfer	**v.** 1. To relocate manufactured goods from one carrier to another (e.g., transferring from a quartz boat to the wafer carrier.)
Component	**n.** A manufactured or purchased good that is added to an assembly to create a new manufactured good.
Consumable	**n.** Resources used during the processing of a machine batch collection (e.g., photoresist, bottled gases). Such resources are usually expended, but catalysts may be considered a consumable.
Dispatch Group	**n.** 1. One or more MGCs requiring the same machine setup grouped together for processing on a class of machine(s).
ECN	See Engineering Change Notice.
Effectivity	**n.** The period in which a specification is available for use. Periods may be bounded by time, usage (e.g., number of Machine Batch runs), product, or other criteria.
Engineering Change Notice	**n.** A collection of specifications bundled together for the purpose of change control. Each specification within the Engineering Change Notice will be reviewed and made available as a whole. For example, a new process flow requires many operational specifications, recipes, etc., which could be packages as one Engineering Change Notice.
Event	**n.** A significant detectable occurrence [SEMI GEM 3.0].
Input Parameter	**n.** A process variable applied to a machine run to control an operation.
InSitu Process Control	**n.** A method for controlling a manufacturing process by observing output parameters during the process, performing some analysis to those parameters to determine the amount of error contained in the output parameters, and taking corrective action on input parameters to maintain a process within specifications.
Location	n. Name of the physical position on a piece of equipment at which a transfer object may reside.

Machine	**n.** 1. (equipment automation hardware/manufacturing resource) A specific piece of machinery, apparatus, process module, or device used to execute an operation. [SEMATECH] 2. (equipment automation software) The intelligent system that communicates with a host computer. [SEMI E4-87 and E5-90] [SEMATECH]
Machine Batch (MB)	**n.** A collection of manufactured goods processed simultaneously in one machine run. A Machine Batch is a proper subset of a Dispatch Group.
Machine Batch Collection (MBC)	**n.** A collection of manufactured goods no larger than the load size of a machine. Example: In semiconductor manufacturing, a cassette is usually placed on a machine's input queue as the machine batch collection. Note: A Machine Batch Collection may contain more than one Dispatch Group. This implies an Machine Batch Collection may require multiple setups.
Machine Rule	**n.** A decision rule place in the SCC1 "Machine Interface" domain to describe variations in activities caused by differences among manufacturing machines. Machine rules usually focus on communication protocol and machine capability differences. When the machine forces the way the business would like to work, a machine rule has been identified. These are usually more subjective.
Machine Run	**n.** A single process iteration, working on a batch of zero or more products extracted from an input station and placed on an output station when complete.
Machine Setup	**n.** The physical and logical state of the machine required prior to processing a dispatch group (e.g., correct recipes/reticles are loaded, the machine is in the correct E10 state).
Machine Setup Class	**n.** A group of machines capable of running the same process using the same machine setup parameters.
Manufactured Good (MG)	**n.** (materials) 1. Smallest unit of manufactured product being counted as inventory. [example: 1) Diffusion Furnace used in a fab acts on "wafers" as the MG. 2) Bonding machine used in assembly acts on a "die" as the MG.] 2. A product manufactured with the intent of being sold.
Manufactured Goods Collection (MGC)	**n.** 1. Any collection of Manufactured Goods tracked and/or scheduled as a unit. 2. All the Manufactured Goods of identical size and characteristics contained in a single shipment. Often used synonymously with manufacturing "lot," but technically an MGC refers to any collection of Manufactured Goods. "MGC" comes from the SEMATECH object-oriented view of manufacturing, in which a group of singular objects is called a "collection."
Operation	**n.** 1. One or more steps within a process on one machine without operator intervention [SEMATECH].

Output Parameters	**n.** 1. Observable variable viewed from outside a process.
Port	**n.** 1. A point or area on the equipment at which a change in equipment ownership of material may occur. [SEMI E-32 MMM]
Process	**n.** 1. A unique, finite course of events defined by purpose or effect, and achieved under given conditions. [SEMATECH] **v.** 2. (equipment automation software) To perform operations on data in process. [SEMATECH] **n.** 3. (micropatterning) A major group of sequential operations in the manufacture of an integrated circuit. Examples are E-beam lithography, thermal deposition, and reactive ion etch. [SEMATECH]
Process Flow	**n.** The sequence of operations and processes necessary to create product from raw materials. Also known, in some companies, as a **routing**.
Queue	**n.** A waiting line; for example, materials or data to be processed. [SEMATECH]. Queues can be either physical (e.g., a material storage rack), or logical (e.g., a list of material that is ready for processing at a machine).
Recipe	**n.** A recipe is the preplanned and reusable portion of the set of instructions, settings, and parameters under control of the equipment that determine the processing environment seen by the manufactured object and that may be subject to change between runs or processing cycles. [RMTF Draft 3.1, 5/91]
Scenario	**n.** A sequence of activities.
Statistical Process Control (SPC)	**n.** The use of statistical methods to analyze a process or its outputs to take appropriate actions to achieve and maintain a state of statistical control and continuously to improve the process capability. [SEMATECH]
Step	**n.** A single action in the performance of an operation, procedure, or process. [SEMATECH]
System Rule	**n.** A decision rule describing the software system architecture used at a site. For example, “are recipes stored on the Cell Controller or the Manufacturing Machine?”
Target Value	**n.** A datum used as a means for specifying the desired value of an output parameter.
Tracking Group	**n.** Any object (e.g., MG, Tool) needing to be tracked. A form of Manufactured Goods Collection (MGC) used to simplify material tracking. This most closely corresponds to the concept of a “lot.” MGCs in a tracking group must be associated with the same process flow and are moved from operation to operation as a whole.

Transfer	**v.** To transport objects (e.g., Manufactured Goods, Tools) and changes of ownership (hand off) between a source and a destination.
Transport	**v.** To physically deliver objects, usually in a "transport carrier," by either human or mechanical mechanisms.
Transport Carrier	**n.** A container used for storing materials, tools, etc. during transport from one location to another.
Transporter	**n.** 1. A machine or human capable of completing those actions required to transport the Transport Group.
Transport Group	**n.** A form of Manufactured Goods Collection (MGC) packaged in a container for transport from one place to another. Note: a transport group may contain more than one tracking group.
Tool	**n.** A removable physical attachment to a machine. Tools may be specific to a set of operations (such as a reticle or probe card), or may be generic to all operations (such as a quartz furnace tube) at a machine.
Unit	**n.** A single piece of product goods (a Manufactured Good).
VFEI	See Virtual Factory Equipment Interface.
Virtual Factory Equipment Interface	**n.** The set of messages used for communication between the cell application system components and the machine interface system components in a logical cell controller.

APPENDIX B: VFEI FILE SYNTAX

Some VFEI messages like EVENT_SETUP and RESOURCE_QUERY offer the option of directing the resulting reply data through a file system. This section defines the syntax for this file. This file format syntax is applicable for both the ASCII and non-ASCII busses.

BNF is used to define the file syntax. Refer to Appendix C for a review of the BNF notation used and definition of some nonterminals. The data item syntax is basically the same as that for the VFEI messages with two main differences.

1. A length field indicating the length of the data item value is added to allow the more efficient parsing of the data items within the file.
2. The allowable format codes are expanded to allow for binary versions of the standard data items formats. Other formats supporting other data types can be defined and included as necessary.

Examples follow to illustrate the syntax.

VFEI File Syntax

<data_item_file> ::= <data_item>+

<data_item> ::= <data_item_name> "/" <format_code>
"/" <data_item_length> "=" <data_item_value>
| <array_item>
| <list_item>

<array_item> ::= <data_item_name> "/" <format_code>
"[" <array_size>"]" "/" <data_item_length> "="
"[" <data_item_value>+ "]"

<array_size> ::= UNSIGNED_INT

<list_item> ::= <data_item_name> "/" "L" "[" <list_size>"]"
"/" <data_item_length> "=" "[" <data_item>+ "]"

The “L” in the above example is referred to as a list. The list allows grouping

data items of related information that may have different formats into a useful structure.

<list_size> ::= UNSIGNED_INT

<data_item_name> ::= STRING

<format_code> ::= "U1" | XU1
| "U2" | XU2
| "U4" | XU4
| "U8" | XU8
| "I1" | XI1
| "I2" | XI2
| "I4" | XI4
| "I8" | XI8
| "F4" | XF4
| "F8" | XF8
| "B" | XB
| "A"
| "BL"

<data_item_length> ::= UNSIGNED_INT

<data_item_value> ::= STRING
| UNSIGNED_INT
| SIGNED_INT
| REAL_NUMBER
| BOOLEAN
| BINARIES

BINARIES ::= Binary formats for the format codes of "XB," "XUn," "XIn," and "XFn" follow the rules defined in SEMI E5-95 section 6.3.1 summarized as follows: Most significant byte first; signed integers are two’s complement; Floating point will conform to IEEE 754 with byte containing sign bit first.

Example file

RECIPE_NAME/A/11="A1D2 COAT"

ALARM_CODE /B /4=0x00

TEMPERATURE /F8[2] /15 = [123.45 678.90]

PRESSURE/F4[3]/16=[10.3 11.5 14.6]

SITE_MEASUREMENT/L[3]/52=[X_COORD/I2/1=1 Y_COORD/I2/1=2
THICKNESS/F4/5=12.24

CD_ARRAY/XB/2105={2105 bytes of unprintable binary}

APPENDIX C: ASCII COMMAND SYNTAX

ASCII command syntax is expressed in Backus Naur Form (BNF). BNF notation is used to describe the frequency and argument type on a command line. BNF was developed for describing computer language lexical rules.

BNF Notation

The notation character sets do not show in the command line but are used for syntax clarification only.

::=	"is defined as"
"ABC"	Quotes indicate a terminal (a symbol that has no rules in the grammar and therefore is grammatically indivisible) that will appear without the quotes.
CAPS	CAPS indicate a terminal that is defined with text.
|	"OR" bar is used to indicate mutually exclusive choices.
<variable>	Indicates a nonterminal (a grammatical construct that can be expressed through rules in terms of smaller constructs) that appears only one time.
<variable>+	+ indicates a nonterminal that appears one or more times.
<var1> <var2>	Nonterminals are separated from each other by "white space" as defined in the ANSI C language.

VFEI Syntax

```
<vfei_message>   ::=  <data_item>+

<data_item>      ::=  <data_item_name> "/" <format_code> "=" <data_item_value>
                      | <array_item>
                      | <list_item>

<array_item>     ::=  <data_item_name> "/" <format_code>
                      "[" <array_size> "]" "=" "["
                      <data_item_value>+ "]"
```

```
<array_size>        ::=   UNSIGNED_INT

<list_item>         ::=   <data_item_name> "/" "L" "[" <list_size> "]"
                          "=" "[" <data_item>+ "]"
```

The “L” in the above example is referred to as a list. The list allows grouping data items of related information which may have different formats into a useful structure.

```
<list_size>         ::=   UNSIGNED_INT

<data_item_name>    ::=   STRING

<format_code>       ::=   "U1"
                    |     "U2"
                    |     "U4"
                    |     "U8"
                    |     "I1"
                    |     "I2"
                    |     "I4"
                    |     "I8"
                    |     "F4"
                    |     "F8"
                    |     "A"
                    |     "B"
                    |     "BL"

<data_item_value>   ::=   STRING
                    |     UNSIGNED_INT
                    |     SIGNED_INT
                    |     REAL_NUMBER
                    |     BOOLEAN
```

STRING	::=	A string of single byte ASCII characters following the syntax conventions of the ANSI C language string literal. This type is associated with the format_code of “A”"
SIGNED_INT	::=	The ASCII representation of a signed integer. This type is associated with the format_codes of "I1", "I2", "I4" and "I8."
UNSIGNED_INT	::=	The ASCII representation of a unsigned integer. This type is associated with the format_codes of "U1", "U2", "U4", "U8" and "B." Both SIGNED_INT and UNSIGNED_INT following the syntax conventions of the ANSI C language integer constants.
REAL_NUMBER	::=	The ASCII representation of a real number following the syntax conventions of the ANSI C language float constant. This type is associated with the format_codes of "F4" and "F8."
BOOLEAN	::=	The ASCII representation of a unsigned integer following the syntax conventions of the ANSI C language integer constants. This type is associated with the format_code of "BL." Additionally a value of 0 indicates FALSE while a non 0 value indicates TRUE.

VFEI Data Item Syntax

The following table lists all of the VFEI standard and predefined data item names and their values. These data items are upper-case and are reserved strings.

Table 14 VFEI Data Item Syntax

VFEI Data Item	Reserved STRING for data_item_name	Format	Valid data_item_value	VFEI messages where used
ALARM_CATEGORY	ALARM_CATEGORY	U1		ALARM REPORT
ALARM_ID	ALARM_ID	I4		ALARM SETUP ALARM REPORT
ALARM_STATE	ALARM_STATE	U1	0 1	ALARM REPORT
ALARM_TEXT	ALARM_TEXT	A		ALARM REPORT
COMMAND_ID	CMD	A	ABORT ALARM_REPORT ALARM_SETUP CMD_ACK DISPLAY_MSG EVENT_REPORT EVENT_SETUP INITIALIZE MACH_CMD PAUSE RES_ACTIVATE RES_DEACT RES_DELETE RES_LIST RES_QUERY RES_REQUEST RES_TRANS RESTART RESUME SET_PARAM START STATUS_LIST STATUS_QUERY TRACE_REPORT TRACE_SETUP	All
COMMAND_TYPE	CMD_TYPE	A		MACHINE COMMAND
DEST_LOCATION	DEST_LOCATION	A		MOVE

VFEI Data Item	Reserved STRING for data_item_name	Format	Valid data_item_value	VFEI messages where used
DEST_PORT	DEST_PORT	A		MOVE
DESTINATION	DESTINATION	A		RESOURCE TRANSFER
DIR	DIR	A		EVENT SETUP TRACE SETUP
ENABLE	ENABLE	U1	0 1	ALARM SETUP EVENT SETUP TRACE SETUP
ERROR_CODE	ECD	U4	See section 7 VFEI Predefined Error Codes	All reply and unsolicited messages
ERROR_TEXT	ETX	A		All reply and unsolicited messages
EVENT_ID	EVENT_ID	A	See section 5 VFEI Events	EVENT REPORT EVENT SETUP
FILE	FILE	A		EVENT REPORT RESOURCE LIST RESOURCE QUERY STATUS LIST STATUS QUERY TRACE REPORT
FREQUENCY	FREQUENCY	U4		TRACE SETUP
LEVEL	LEVEL	U2		ABORT
MACHINE_ID	MID	A		All
MESSAGE	MESSAGE	A		DISPLAY MESSAGE
MSG_TYPE	MTY	A	C R E	All
NOEVENT *****OBSOLETE*****	NOEVENT	U1	0 1	INITIALIZE
REPORT_SIZE	REPORT_SIZE	U2		TRACE REPORT TRACE SETUP
RESOURCE_BODY	RES_BODY	*1		RESOURCE TRANSFER
RESOURCE_FILE	RES_FILE	A		RESOURCE REQUEST

VFEI Data Item	Reserved STRING for data_item_name	Format	Valid data_item_value	VFEI messages where used
				RESOURCE TRANSFER
RESOURCE_ID	RES_ID	A		RESOURCE ACTIVATE RESOURCE DEACTIVATE RESOURCE DELETE RESOURCE LIST RESOURCE QUERY RESOURCE REQUEST RESOURCE TRANSFER
RESOURCE_LENGTH	RES_LENGTH	U4		RESOURCE TRANSFER
SAMPLES	SAMPLES	U2		TRACE SETUP
SOURCE	SOURCE	A		RESOURCE TRANSFER
SRC_LOCATION	SRC_LOCATION	A		MOVE
SRC_PORT	SRC_PORT	A		MOVE
TERMINAL	TERMINAL	B		DISPLAY MESSAGE
TG_ID	TG_ID	A		MOVE
TG_TYPE	TG_TYPE	A		MOVE
TID	TID	U4		All command and reply messages
TRACE_ID	TRACE_ID	U2		TRACE REPORT TRACE SETUP
TYPE	TYPE	A	RECIPE TOOL COMPONENT CONSUMABLE MACHINE VFEI	RESOURCE ACTIVATE RESOURCE DEACTIVATE RESOURCE DELETE RESOURCE LIST RESOURCE QUERY RESOURCE REQUEST

VFEI Data Item	Reserved STRING for data_item_name	Format	Valid data_item_value	VFEI messages where used
				RESOURCE TRANSFER SET PARAM
VARIABLE_ID	VAR_ID	A	See section 6 VFEI Variables	EVENT SETUP STATUS QUERY TRACE SETUP

*1 : RESOURCE BODY format is defined by the equipment.

VFEI Syntax examples

CMD/A="INITIALIZE" MID/A="STP1" MTY/A="C" TID/U4=1

CMD/A="ALARM_SETUP" MID/A="STP1" MTY/A="C" TID/U4=2 ENABLE/U1=0 ALARM_ID/I4[3]=[1 2 3]

CMD/A="STATUS_LIST" MID/A="TRK1" MTY/A="R" TID/U4=10 ECD/U4=0 ETX/A="" SOMEVAR/L[3]=[FIRST/A="TEXT" SECOND/U2=0xFF THIRD/F4=0.003]

CMD/A="EVENT_SETUP" MID/A="STP1" MTY/A="C" TID/U4=11 ENABLE/U1=1 EVENT_ID/A="MBC_START" VAR_ID/A[2]=["CLOCK" "MBC_SIZE"]]

CMD/A="EVENT_REPORT" MID/A="STP1" MTY/A="E" ECD/U4=0 ETX/A="" EVENT_ID/A="MB_COMPLETE" SOMEVAR/L[4]=[CLOCK/A="1992021115010000" FIRST/A="TEXT" SECOND/U2=255 THIRD/F4=0.003]

APPENDIX D: NON-ASCII COMMAND SYNTAX

Material on this topic will be provided in a subsequent release.

SEMATECH Technology Transfer
2706 Montopolis Drive
Austin, TX 78741

http://www.sematech.org